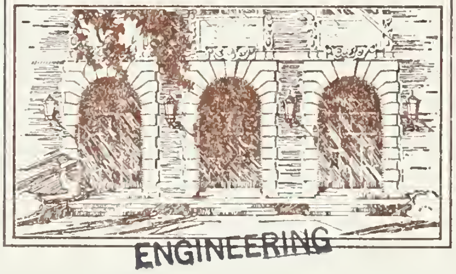


Center for Advanced Computation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

CAC Document No. 17

A FAST FOURIER TRANSFORM SUBROUTINE FOR ILLIAC IV

by

James E. Stevens, Jr.

October 15, 1971

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUC-CAC-DN-71-17	2.	3. Recipient's Accession No.

4. Title and Subtitle

A FAST FOURIER TRANSFORM SUBROUTINE FOR ILLIAC IV

5. Report Date
October 15, 1971

6.

7. Author(s)
James E. Stevens, Jr.

8. Performing Organization Rept. No. CAC-17

9. Performing Organization Name and Address

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.
DAHC04-72-C-0001

12. Sponsoring Organization Name and Address

U. S. Army Research Office-Durham
Duke Station
Durham, North Carolina

13. Type of Report & Period Covered
Research

14.

15. Supplementary Notes

16. Abstracts

This report is a description of a Fast Fourier Transform (FFT) subroutine written in assembly language for the ILLIAC IV computer. The subroutine uses the Cooley-Tukey algorithm for performing discrete Fourier transforms. The parallel nature of the Cooley-Tukey method lends itself very well to a highly parallel machine like ILLIAC IV. Timing simulation results have shown that this program will perform Fourier transforms faster than they are being done on any existing computer system.

17. Key Words and Document Analysis. 17a. Descriptors

Seismic Data Processing
ASK
Fast Fourier Transform (FFT)
Discrete Fourier Transform (DFT)
Time Series
Frequency Spectrum

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement No restriction on distribution. Available from National Technical Information Service, Springfield, Virginia 22151	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 67
	20. Security Class (This Page) UNCLASSIFIED	22. Price

CAC Document No. 17

A FAST FOURIER TRANSFORM SUBROUTINE FOR ILLIAC IV

by

James E. Stevens, Jr.

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

October 15, 1971

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the U.S. Army Research Office-Durham under Contract No. DAHC04 72-C-0001.

ABSTRACT

This report is a description of a Fast Fourier Transform (FFT) subroutine written in assembly language for the ILLIAC IV computer. The subroutine uses the Cooley-Tukey algorithm for performing discrete Fourier transforms. The parallel nature of the Cooley-Tukey method lends itself very well to a highly parallel machine like ILLIAC IV. Timing simulation results have shown that this program will perform Fourier transforms faster than they are being done on any existing computer system.

TABLE OF CONTENTS

Introduction

The Fourier transform is one of the most important mathematical aids to signal processing. The basic property of the Fourier transform is its ability to distinguish waves of different frequencies which have been additively combined. The result of a Fourier transform on a complex function of time is a frequency spectrum of that function. This resulting frequency spectrum is also a complex function. To facilitate processing on digital machines, a method for transforming discrete functions of time into discrete functions of frequency has been derived. The time function must consist of evenly spaced samples of some interval, T. A function of N samples could be represented as $f(nT)$ for $0 \leq n \leq N - 1$. The spectrum of the function would also consist of discrete values evenly spaced in frequency. If the spectrum were composed of N points, the interval of frequency, Ω, would be $2\pi/NT$. The spectrum $F(k\Omega)$, $0 \leq k \leq N - 1$ would be given by

$$F(k\Omega) = \sum_{n=o}^{N-1} f(nT)e^{-jnTk\Omega} \tag{1}$$

This equation defines the discrete Fourier transform (DFT) of a sequence of N samples. It follows directly that the DFT is linear since

$$\mathrm{DFT}\{f(nT) + g(nT)\} = \mathrm{DFT}\{f(nT)\} + \mathrm{DFT}\{g(nT)\} \tag{2}$$

and

$$\mathrm{DFT}\{c(f(nT))\} = c[\mathrm{DFT}\{f(nT)\}] \tag{3}$$

This follows quickly from the definition.

There exists an inverse DFT, a transform which maps a discrete Fourier transform back into the sequence from which it was composed. It is given by

$$f(nT) = \frac{1}{N} \sum_{k=o}^{N-1} F(k\Omega)e^{jnTk\Omega} \tag{4}$$

which differs from the DFT equation 1, only by a scale factor and by the sign of the exponential. The same procedure used for the DFT can thus be applied to a frequency spectrum to perform the inverse Fourier Transform by simply dividing the result by N.

The above method for computing DFT requires N^2 complex operations to be performed in order to transform a function with N samples. The method of Cooley and Tukey [1] eliminates redundant steps in that method resulting in a procedure requiring only $N \log_2 N$ complex operations. This method is highly parallel and therefore very well suited for use on ILLIAC IV [2].

Subroutine Specifications

The ILLIAC IV fast Fourier subroutine accepts discrete input functions composed of N complex samples where N is an even power of two between 8 and 4096, inclusive. The output of the subroutine is a discrete frequency spectrum composed of N complex samples as defined by equation (1). The output is stored over the input in ascending sequence in a contiguous block of memory. N words of storage are used, where each word contains the real (outer) and imaginary (inner) portions of one sample in the 32-bit format. Input to the subroutine must also be stored in this manner; therefore, all values have 24-bit significance and are represented in floating point. Several functions can be input to the subroutine at one time as long as they are stored contiguously and provided that they all are the same length. If the transform of a function of M samples is desired where M is not a power of two, N-M zero samples should be added to the function where N is the smallest power of two greater than M. The result can be input to the subroutine as a function of N samples. The output will be a frequency spectrum equivalent to the transform of the original function of M samples with a smaller frequency spacing.

Implementation of the Cooley-Tukey Procedure

The method of Cooley and Tukey [1] is illustrated in Figure 1. Implementation of this method on ILLIAC IV has been discussed by G.M. Ackins [3]. Ackins' methods have been refined and generalized for use in this subroutine. The following description gives the details of the final implementation.

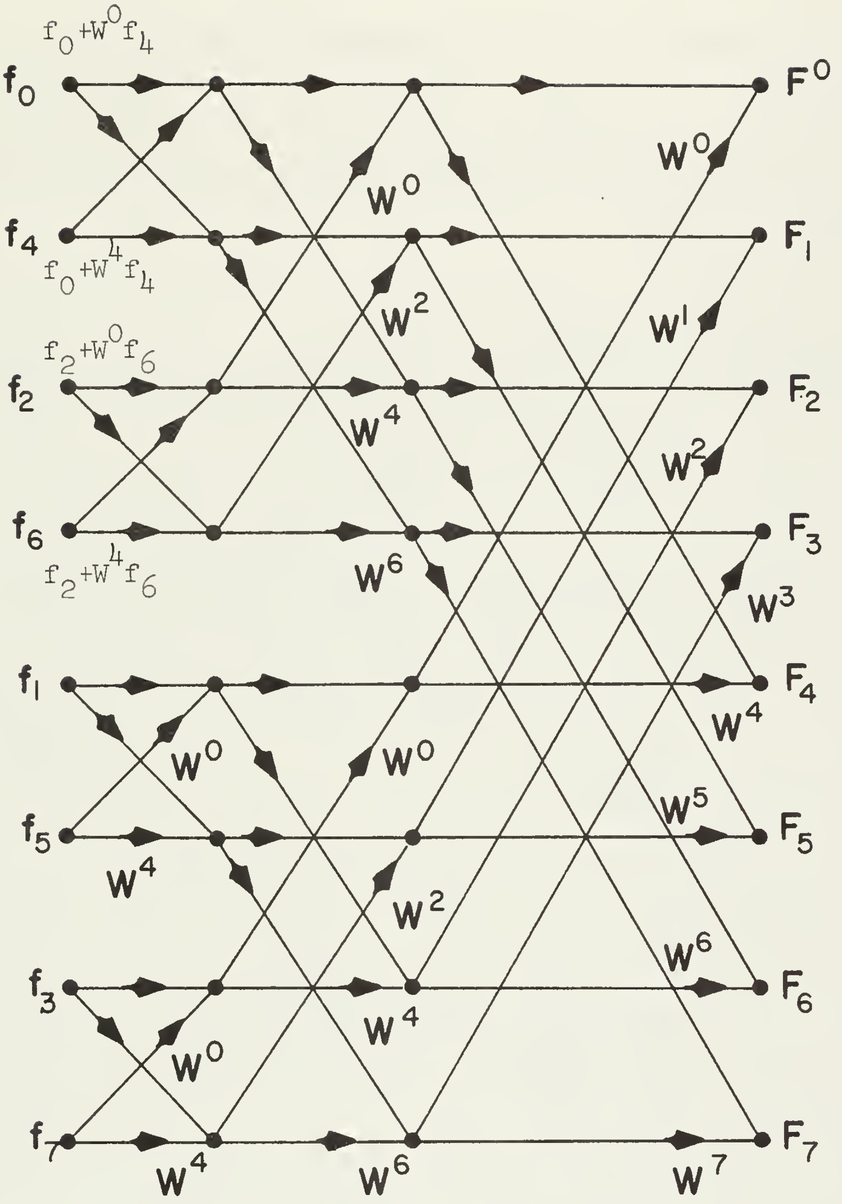


Figure 1

In Figure 1, f_n represents f(nT) and F_n represents F(nΩ). Also W is used to replace the constant $e^{-j\frac{2}{N}}$. At each node in the diagram, a complex multiply involving the lower of the two inputs to the mode and the constant, W, raised to the appropriate power is performed. The result is added to the upper of the two inputs giving an answer which is used as an input to the next iteration until the transform is completed. At each stage in the process where an iteration is complete, the partial results are stored into the initial locations so that the data is transformed in place. Figure 2 illustrates how this transform would be accomplished in parallel on an eight PE ILLIAC IV.

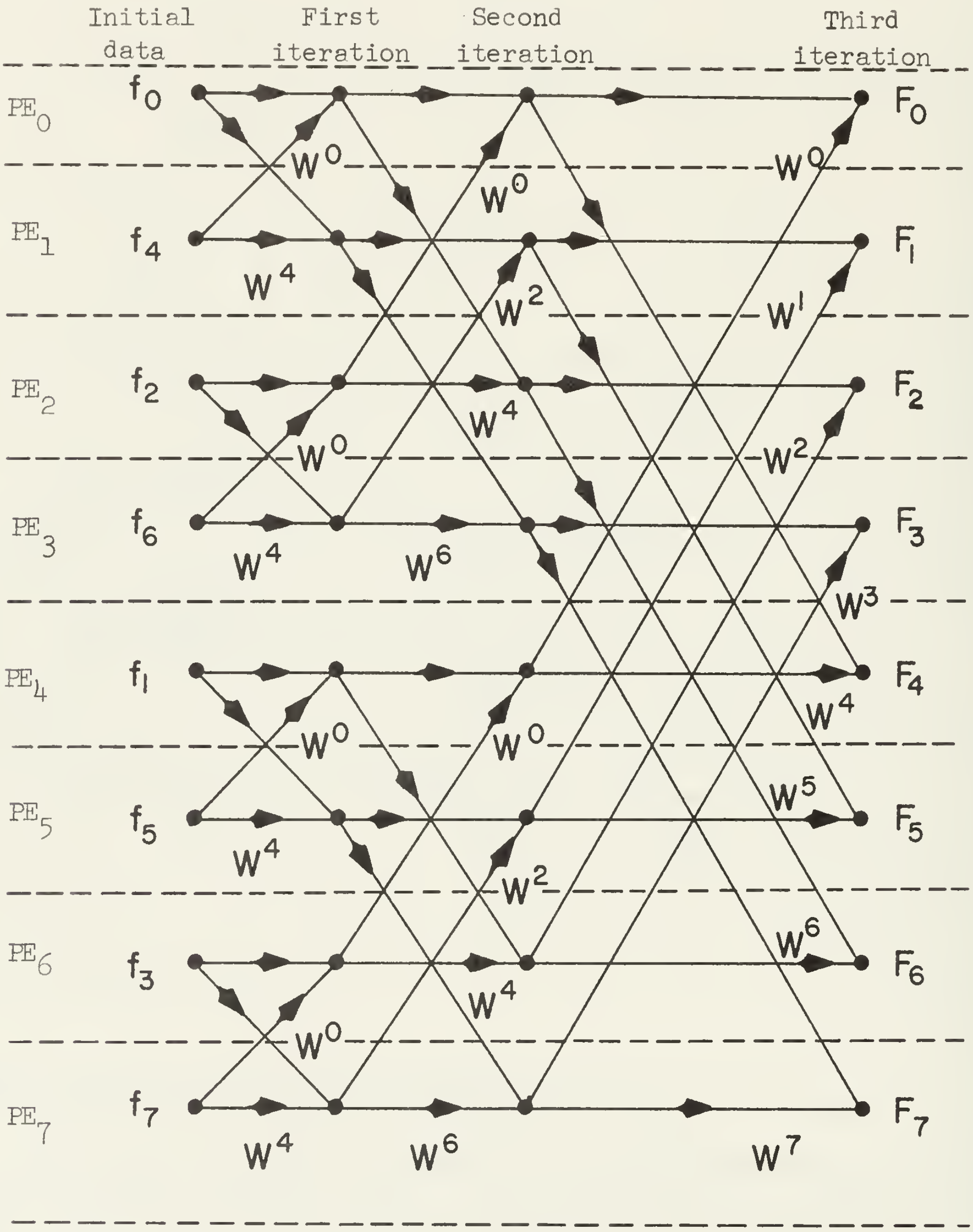


Figure 2

In Figure 2, the arrows represent routing of data between PE's while the nodes represent the operations which take place within the PE's as given for Figure 1. f(t), and all of the necessary powers of W are stored in PE memory for use in the calculation. From Figures 1 and 2, it is seen that the input data has been "shuffled" so that the output appears in ascending sequence. This shuffling also causes the routing (arrow) pattern to be very regular. The algorithm for ordering input data is explained in a later section of this report which describes the routine called Scramble which performs the shuffling in ILLIAC IV.

The method described above and illustrated in Figure 2 can be generalized to transform much larger functions. Careful examination of Figure 1 reveals that after two iterations, two separate four point transforms have been completed. A four point transform is depicted in Figure 3. It should be noted that the value of W depends on the size of the function being transformed.

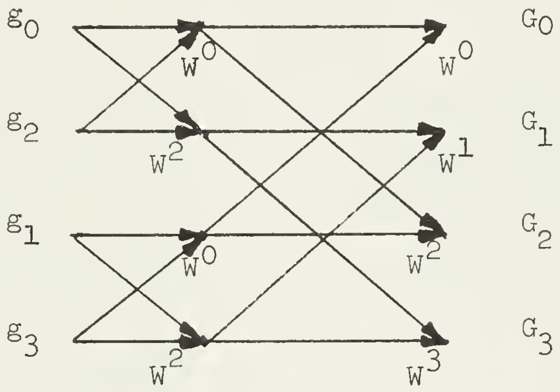
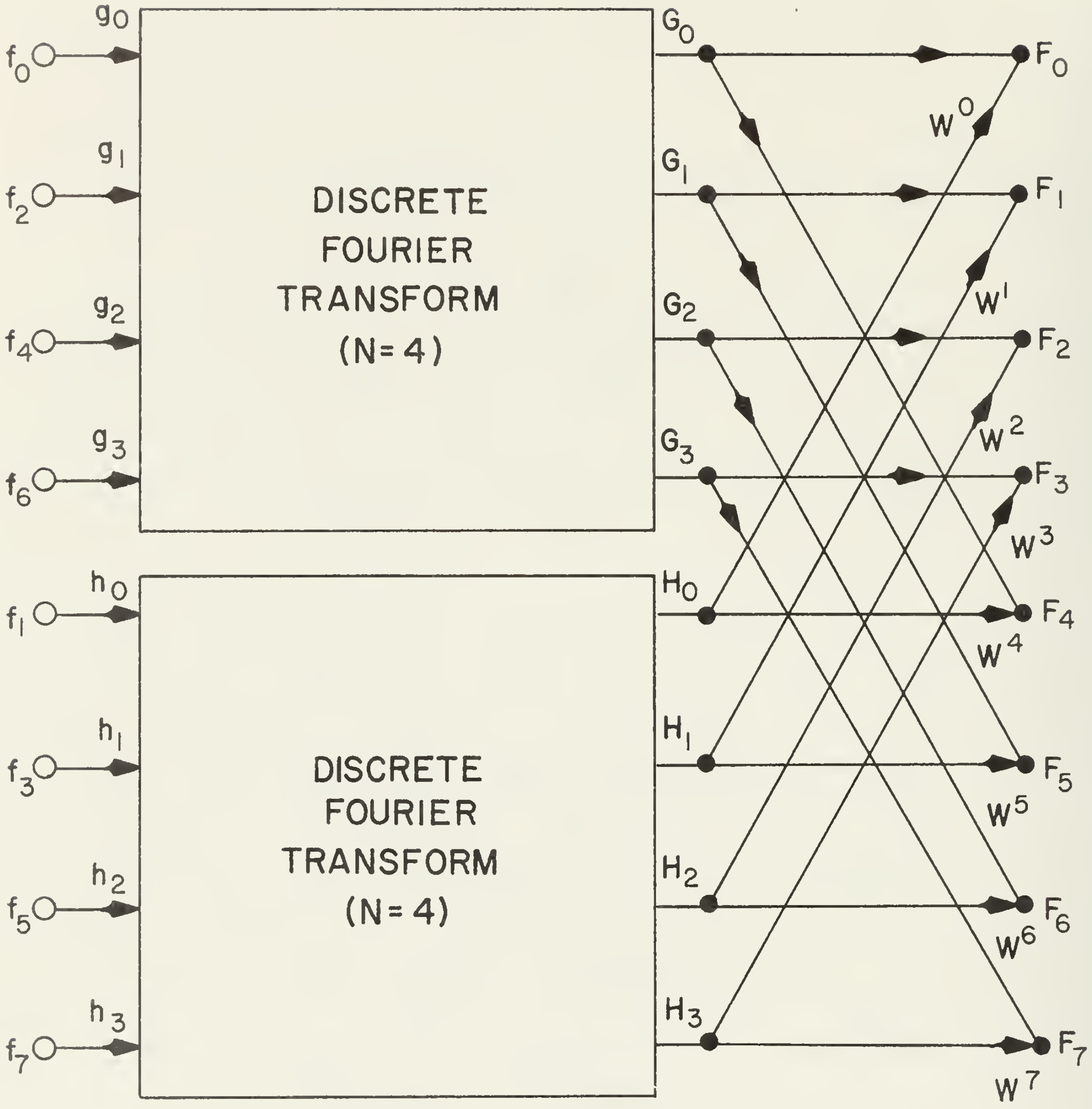


Figure 3

The eight point transform can now be represented in terms of the four point transform as depicted in Figure 4.



f_0
f_2
f_4
f_6
f_1
f_3
f_5
f_7
g_0
g_1
g_2
g_3
h_0
h_1
h_2
h_3
DISCRETE FOURIER TRANSFORM (N=4)
DISCRETE FOURIER TRANSFORM (N=4)
G_0
G_1
G_2
G_3
H_0
H_1
H_2
H_3
F_0
F_1
F_2
F_3
F_4
F_5
F_6
F_7
W^0
W^1
W^2
W^3
W^4
W^5
W^6
W^7

Figure 4

The equality of these two representations (Figure 1 and Figure 4) is important to the structure of the subroutine. In an eight point transform, $W = e^{-j\frac{2}{8}}$, so that $W^4 = e^{-j}$. In a four point transform, $W = e^{-j\frac{2}{4}}$, so that $W^2 = e^{-j}$. Thus, an eight point transform can be accomplished by performing two four point transforms and iterating once more. In the general case any N point transform, where N is a power of two, can be accomplished by performing two N/2 point transforms and combining those results with one final iteration. One consequence of this is that the first step in performing any size transform is to perform an eight point transform on each group of eight points. The entire subroutine is built up from that basis.

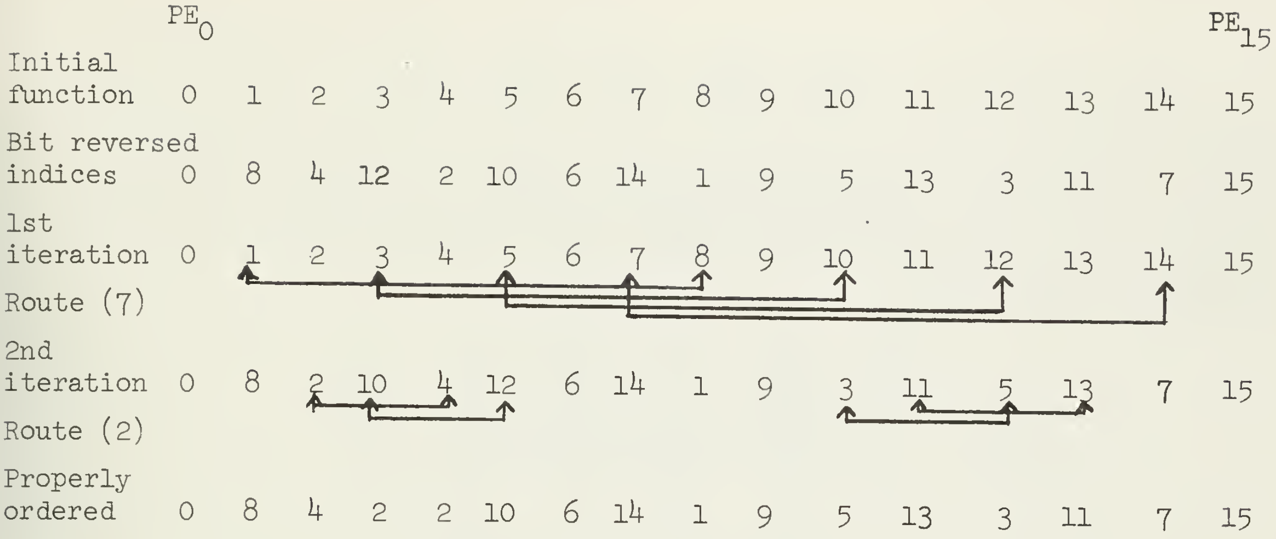
The other consequence of this partitioning of the procedure is that each stage is completely independent. When a function composed of 128 data samples is to be transformed on ILLIAC IV, it is simply stored in two rows of PE memory. One row of PE memory consists of one word in each PE. The first 64 samples are stored in one row (after proper scrambling) and the second half are stored in the other row. This is referred to as "folding" 128 words of data into two 64 word rows of memory. A 64 point transform can now easily be performed on the first half of the data since this operation is independent of the other half. This 64 point transform is a simple extension of the eight point transform of Figure 2, using 64 PE's. Another independent 64 point transform is then performed on the second half of the data.

The results of these two 64 point transforms can be combined in one simple iteration giving the final 128 point transform. Since the input data was folded in PE memory, no routing is required to align the two halves for this final iteration simplifying it even further. This can be illustrated by observing that the final iteration of each transform requires a route of N/2 positions. For a 128 point transform this would be a route of 64 which is equal to a route of 0 in a single quadrant ILLIAC IV. Furthermore, every transform larger than 64 requires a final route which is a multiple of 64 which means no routing is actually required. Since there is a smaller proportion of time spent in routing, the largest transform is actually the most efficient in its computation.

The final generalization is that each larger transform above 64 is simply accomplished by folding the function and treating the halves independently until the final iteration. The size of the function will determine how many eight point transforms are done as the first step. Also the number of functions to be transformed at a time can be taken into account through proper loop control. For example, two 2048 point transforms could quite easily be done at the same time. They would each be scrambled separately and then the transform would proceed as if the two functions composed a single 4096 point function--only, the final iteration would not be performed.

Scrambling

The shuffling of data in the processing elements before performing the transform allows the computation to be done in place, that is, by writing all intermediate results over the original data. This process will be referred to as prescrambling and when it is applied the transform generates results which are sequentially ordered across PE's. The original function is composed of discrete samples at even spaced time intervals. They are assigned indices 0 through N in order. Each index can be represented by a binary number with $\log_2 N$ digits, $K_0 K_1 \ldots K_n$, where $n = \log_2 N - 1$. The scrambling process requires that each sample having index $K_0 K_1 \ldots K_n$ be interchanged with the sample having index $K_n K_{n-1} \ldots K_0$. Thus by a simple "bit reversal" of each binary index the destination of each sample in the scrambling process is determined. Note that in Figure 1 a three digit bit reversal on the sequence 0,1,2,3,4,5,6,7 yields the proper sequence for input to the transform procedure, i.e., 0,4,2,6,1,5,3,7. Consider the following example to see how the scrambling is accomplished within ILLIAC IV. The input data is represented by indices only.



	PE_0															PE_{15}
Initial function	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bit reversed indices	0	8	4	12	2	10	6	14	1	9	5	13	3	11	7	15
1st iteration	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Route (7)																
2nd iteration	0	8	2	10	4	12	6	14	1	9	3	11	5	13	7	15
Route (2)																
Properly ordered	0	8	4	2	2	10	6	14	1	9	5	13	3	11	7	15

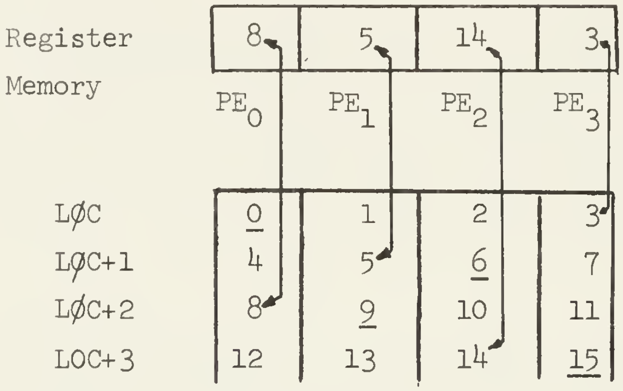
This process can be generalized for scrambling N points in the following manner. The odd numbered PE's less than N/2 send data to the corresponding even numbered PE's greater than N/2 and receive data back from them. The next iteration is the same except the first N/2 PE's and the second N/2 PE's perform it independently and handle the data by pairs. Note that only two routes are required for each iteration. By this method ILLIAC IV can scramble a 64 point data set in three iterations requiring only 6 routes. For larger data sets the data must be folded in storage and a different scheme is used. The following example shows how a sixteen point data set would be scrambled by a 4 PE ILLIAC IV.

	PE_0	PE_1	PE_2	PE_3
LOC	0	1	2	3
LOC+1	4	5	6	7
LOC+2	8	9	10	11
LOC+3	12	13	14	15

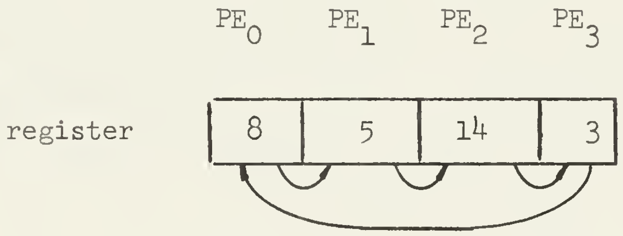
Proper reordering
determined by bit reversal

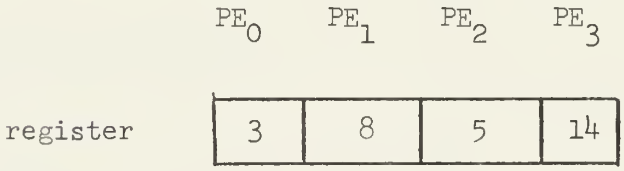
PE_0	PE_1	PE_2	PE_3
0	8	4	12
2	10	6	14
1	9	5	13
3	11	7	15

The underlined indices denote the positions that are unaffected by scrambling since data in these positions does not need to be moved. The location of these stationary positions within each PE is given by performing a bit reverse operation on the PE numbers. Each PE now loads a register with data from the location corresponding to the stationary location in the next higher numbered PE.

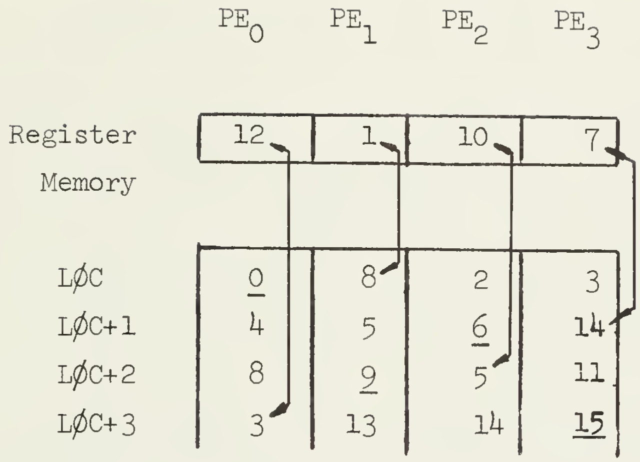


Each PE routes the data in its register to the next higher numbered PE. The rightmost PE routes back to the first PE.



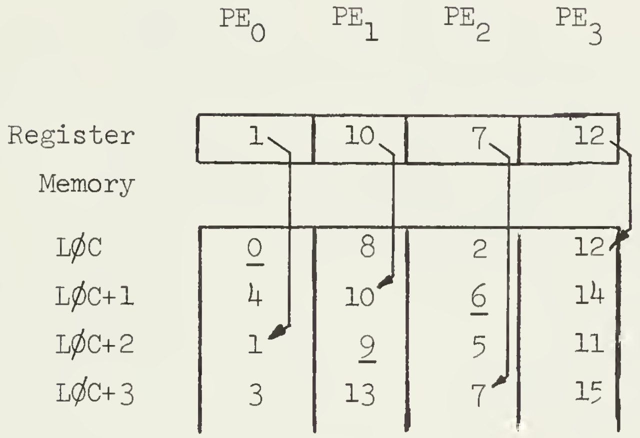


Now the data in each register is exchanged with the data from the location in the PE corresponding to the stationary location in the next lower numbered PE.



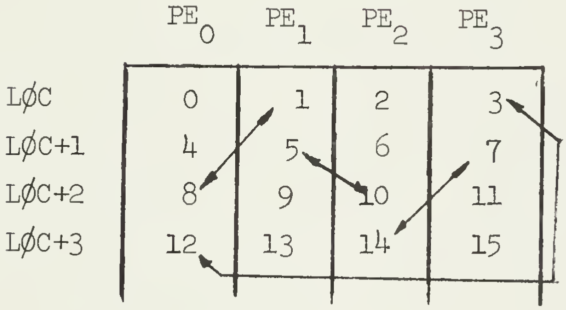
	PE$_0$	PE$_1$	PE$_2$	PE$_3$
Register	12	1	10	7
Memory				
LØC	0	8	2	3
LØC+1	4	5	6	14
LØC+2	8	9	5	11
LØC+3	3	13	14	15

The data in the registers is now routed back one PE and stored where the first data came from.



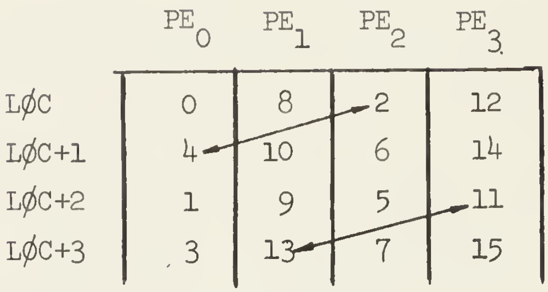
	PE$_0$	PE$_1$	PE$_2$	PE$_3$
Register	1	10	7	12
Memory				
LØC	0	8	2	12
LØC+1	4	10	6	14
LØC+2	1	9	5	11
LØC+3	3	13	7	15

The transfers accomplished can be illustrated as shown below:



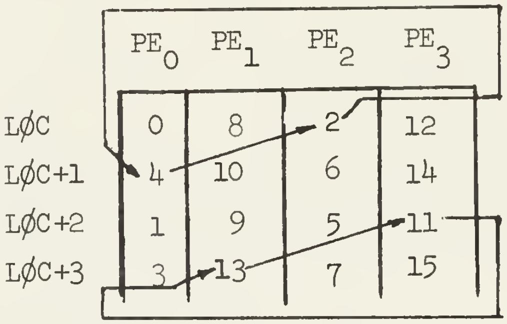
	PE$_0$	PE$_1$	PE$_2$	PE$_3$
LØC	0	1	2	3
LØC+1	4	5	6	7
LØC+2	8	9	10	11
LØC+3	12	13	14	15

The transfers to complete the scrambling would be:



	PE_0	PE_1	PE_2	PE_3
LØC	0	8	2	12
LØC+1	4	10	6	14
LØC+2	1	9	5	11
LØC+3	3	13	7	15

This final transfer is accomplished by only one route since each data point moves exactly two PE's away (end around). Once again all PE's are active. The following diagram shows how this final step is accomplished:



	PE_0	PE_1	PE_2	PE_3
LØC	0	8	2	12
LØC+1	4	10	6	14
LØC+2	1	9	5	11
LØC+3	3	13	7	15

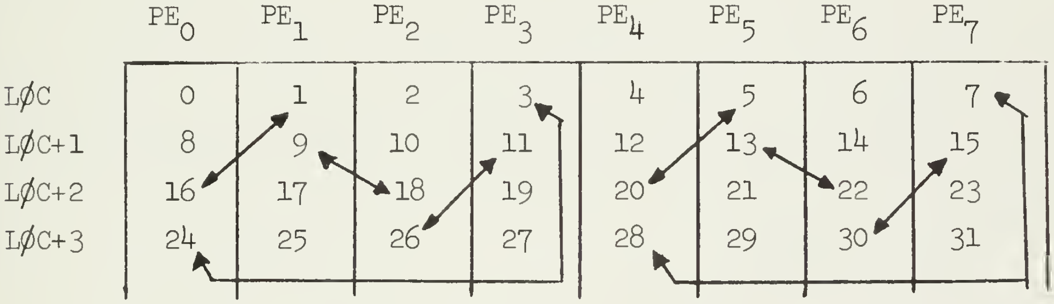
Thus, after two iterations, the data has been properly ordered for input to the transform routine.

	PE_0	PE_1	PE_2	PE_3
LØC	0	8	4	12
LØC+1	2	10	6	14
LØC+2	1	9	5	13
LØC+3	3	11	7	15

In the operations described above, each PE is always performing some tasks and the data is never moved to an intermediate memory location so that the scrambling is done in the minimum number of steps and with 100% efficiency in the use of PE's. It can be seen that neither of these statements is true about the scheme described at the beginning of this section or the scheme described by Ackins [3]. Unfortunately this algorithm works only on data sets which have N^2 points, where N is the number of PE's in the array.

Since ILLIAC IV has 64 PE's it can scramble a folded data set of 4096 points 100% efficiently. The algorithm has been extended to handle small data sets.

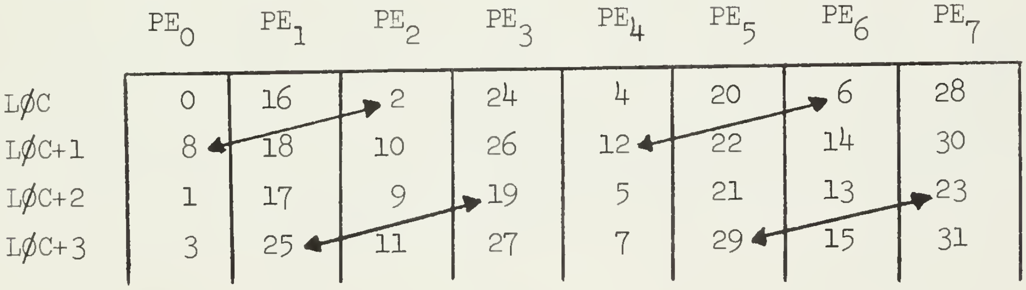
Consider the following example of scrambling a 32 point data set in an 8 PE ILLIAC. The initial state would be:



	PE_0	PE_1	PE_2	PE_3	PE_4	PE_5	PE_6	PE_7
LØC	0	1	2	3	4	5	6	7
LØC+1	8	9	10	11	12	13	14	15
LØC+2	16	17	18	19	20	21	22	23
LØC+3	24	25	26	27	28	29	30	31

Now perform the transfers indicated. It should be apparent that these operations take four rather than two routes since each group of four PE's cannot be treated separately in an end around fashion. Also these transfers cannot be 100% efficient since more than one routing distance is involved.

The result of the transfers would be:



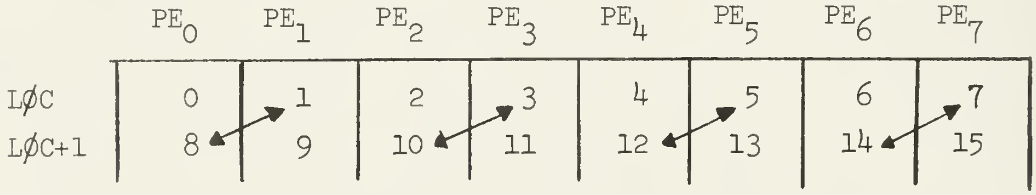
	PE_0	PE_1	PE_2	PE_3	PE_4	PE_5	PE_6	PE_7
LØC	0	16	2	24	4	20	6	28
LØC+1	8	18	10	26	12	22	14	30
LØC+2	1	17	9	19	5	21	13	23
LØC+3	3	25	11	27	7	29	15	31

Now only the above indicated transfers are required to complete the scrambling process. Note that the process is simply two applications of the previous method performed in parallel. The final result is:

	PE_0	PE_1	PE_2	PE_3	PE_4	PE_5	PE_6	PE_7
LOC	0	16	8	24	4	20	12	28
LOC+1	2	18	10	26	6	22	14	30
LOC+2	1	17	9	25	5	21	13	29
LOC+3	3	19	11	27	7	23	15	31

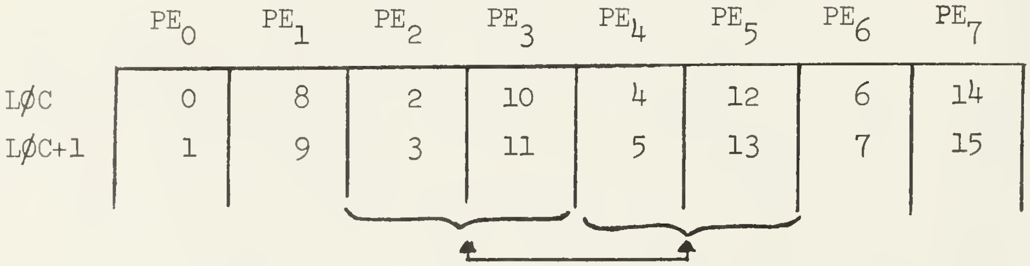
A generalized method for scrambling data which is folded in memory can now be given. Data is first scrambled in groups which are composed of as many PE's as there are rows of data by the method developed for "square" data sets (referring to N^2 data points in N PE's). Finally, consider these "squares" as units and scramble their position. One further example will make this last step more clear. Consider 16 data points in an 8 PE array.

First iteration



	PE_0	PE_1	PE_2	PE_3	PE_4	PE_5	PE_6	PE_7
LOC	0	1	2	3	4	5	6	7
LOC+1	8	9	10	11	12	13	14	15

The heavy lines below indicate the second step which completes the scrambling, giving the proper result.



	PE_0	PE_1	PE_2	PE_3	PE_4	PE_5	PE_6	PE_7
LOC	0	8	2	10	4	12	6	14
LOC+1	1	9	3	11	5	13	7	15

Result

	PE_0	PE_1	PE_2	PE_3	PE_4	PE_5	PE_6	PE_7
LØC	0	8	4	12	2	10	6	14
LØC+1	1	9	5	13	3	11	7	15

Although the present implementation allows for only up to 4096 data points, larger data sets could be handled and it is easily seen that scrambling would be 100% efficient, but the core restrictions on the size of the data set and the size of the program would eventually limit the program. A short discussion of these restrictions is included under future considerations.

Future Considerations

Various extensions to this subroutine have been explored. Some of them seem impractical while others are quite worthy of further investigation and perhaps implementation. There also has been a suggestion that a separate subroutine be written to handle several parallel transforms, each with all of its data within one PE. If all transforms could be done in groups of 64 this would appear to be a very efficient method; however, if groups of data are not even multiples of 64, some PE's will be completely idle during the whole process. It is this author's opinion that such a subroutine would be very straightforward to code and would have enough value to be worth programming at some future time.

The question of expanding the present subroutine to accept data sets larger than 4096 sample is largely decided by available core. There does not seem to be too much demand for greater capability at this time and the program itself already consumes approximately 2,500 words of memory, mostly composed of stored constants. Each increase of a power of two in the size of data sets transformable will increase the size of the subroutine by almost a factor of two. Such increases can only be justified by a large demand for that capability. If and when expansion is warranted, the process would be straightforward with the possible restrictions of a 64K limit on the size of the data set so that it could be completely core contained. Problems with formatting and accessing disk files tend to be formidable.

It has been shown by Robinson [4] that two real data sets can be handled simultaneously as if they composed one complex data set of the same size. It was suggested that capability be added for handling real data in this manner, with hopes of a factor of two increase in efficiency. The algorithm was studied and appears simple enough but applying it to a parallel structure like ILLIAC IV does not seem feasible at this time.

ILLIAC IV FAST FOURIER TRANSFORM
USERS MANUAL

This report is intended to be a description of how to use the Fast Fourier Transform subroutine and exactly what it can do. An explanation of the Fourier Transform and of how the subroutine works on ILLIAC IV is given in ILLIAC IV Report #226 entitled "A Fast Fourier Transform Subroutine for ILLIAC IV." Also, copies of the assembly language listing of the subroutine are available from the Center for Advanced Computation and the University of Illinois.

The Fast Fourier Transform is used to transform a discrete time series of data points into a discrete frequency spectrum. Both input and output data values are represented as complex numbers in the following way: each number occupies on 64-bit word using the 32-bit floating point configuration where the real part of each number occupies the inner 32 bits. When the input series consists of only real numbers the numbers should be stored only in the outer portion of each word and the inner portions should be set to zero. In any case the output will always consist of complex numbers and will be stored in the same locations as the input, thus "overwriting" the input.

Due to the structure of the Fast Fourier Transform algorithm all input sequences must contain a number of data points which is an even power of two. The sizes of data sets which can be transformed by the ILLIAC IV subroutine are: 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096. Other sizes of data sets can be transformed by "padding" them with zeros on the end until their size is in an even power of two between 8 and 4096 inclusive. The result of such a transform will be correct with only the frequency spacing of the result being effected.

This subroutine uses the standard subroutine linking conventions as described in "The ILLIAC IV Software Reference Manual." When referenced from a GLYPNIR program, a standard subroutine call statement can be used. When referenced from an assembly language program, the standard call macro can be used. This macro and an example of its use is listed in Appendix A. The Fast Fourier Transform subroutine requires three parameters named size, number, and pointer. These parameters have the following significance.

SIZE

Size gives the number of data points present in each input time series. Size must be an integer and should be an even power of two between 8 and 4096 inclusive. If it is not an even power of two, the next greatest power of two will be assumed and will determine the size of the transform and the number of results returned. Note that the data must still be "padded" with zeros for the result to be meaningful. If the size specified is zero or greater than 4096, an error condition will be indicated and no transform will be performed. An error is indicated by filling ACAR2 with all ones.

NUMBER

Number must also be an integer specifying the number of transforms to be performed during this call to the subroutine. Any number of transforms can be performed as long as they are all the same size and as long as all of the input data ia available in one continuous block of memory. If number is set to zero, no transform will be performed and ACAR2 will be set to all ones.

POINTER

Pointer is a 24-bit CU address giving the location of the first word of a continuous block of memory which contains all of the input data for this call of the transform subroutine. Pointer will also address the first word of the resulting frequency spectrum(s) when the subroutine is exited. This pointer is automatically set up by the calling macro which is exhibited in Appendix A.

The actual subroutine is contained in a file called FASTFOURIER/TRANSFORM which is available on a system tape of the subroutine library. This file must be properly linked to the calling program. The subroutine is called by using the entry identifier FFT.

INVERSE TRANSFORM

For performing inverse transforms the only change required is to call by using the entry identifier FFTINV.

APPENDIX A

Standard call macro from "The ILLIAC IV Software Reference Manual"

```
DEFINE CALL &NAME(&PARAMETERS)=
       &IF &SIGN(&MFIELD (&NAME))
           &THEN EXTERNAL &NAME; &FI

       BEGIN BLOCK
           BEGIN USE 63
           LIST:  DATA &PARAMETERS;
           END;
           CLC(2);
           SLIT(2)  LIST;

       END;
       CLC(3);
       SLIT(3)  &NAME;
       EXCHL(3)  $ICR; ##;
```

Sample calling sequence to execute two 138 point transforms.

```
CALL FFT(HOWBIG,HOWMANY,WHERE);
.
.
.
WHERE:  DATA        [128 words of data];
HOWBIG:  DATA       64;
HOWMANY:  DATA      2;
```

APPENDIX B

Demonstration Run

This appendix compares a Fourier Transform performed on a Univac 1108 and the same transform performed on the ILLIAC IV simulator. The information is presented in the form of actual computer listing including the following: 1) A listing of the 256 real input data points. 2) The results of the Univac run (real part first followed by imaginary part). 3) The main program used to call the FFT routine for the ILLIAC IV simulation run. 4) The results of the simulation run. Differences in the two results are due primarily to the difference of the word size used in each case.

Input Data

```
        0.        0.      -56.      -24.     4822.   -13007.    -9326.     5835.
    17167.    12047.        3.     -696.    -1676.     7734.    -2215.    -7602.
    10211.      165.   -17335.     9888.   -12968.     2744.    14245.    10537.
   -11413.    14762.     4049.     3069.    17389.     6574.    18416.      507.
    10625.     2364.    -3213.    -3609.    -6436.   -11208.     3291.     7014.
    -7760.    -3226.   -16879.    12173.    -6848.    -8584.     7979.    13897.
     1273.    -1518.    11723.     3709.      768.    -7109.     3905.   -17075.
    17524.    -6860.     8704.     1967.    -4606.   -11202.    16318.     3432.
     4807.   -19886.    -6603.   -11205.   -11151.      762.     7412.      951.
     3167.     3270.    11954.     3003.    -1335.   -14578.    17089.     -845.
    -6565.   -12545.     4537.     5618.    -5144.   -13695.    10170.    -3683.
    -8762.     -924.    -2686.     4718.    -4320.    -5671.   -11186.    12147.
     -651.    11049.   -12576.     3387.     1053.   -12898.     5476.     1719.
    -5692.    -9780.     6118.     4889.   -15757.     6186.    -1947.    -1902.
     9280.     4003.   -13438.    13466.    -1407.    -5415.     2584.   -13501.
     4027.    -4176.     2236.   -10937.     8573.   -13987.     3345.    -1129.
    -5302.    -9614.    -6415.   -13410.    13447.     1673.   -10575.    -3613.
      165.     -191.      499.     1514.    -5063.    19279.      832.    -5213.
   -12544.     4976.   -17782.   -11996.    10533.    -5940.    -2264.     7824.
     6165.   -16655.     9024.    17666.     3006.    -7317.    -8118.     3976.
    -5297.   -18975.    10253.    16419.     2224.    12068.      392.     1382.
     -312.     -238.    10886.     1551.     2753.     6381.     8903.     -393.
   -14823.    11119.    -9271.    -7248.     1997.    -1026.      809.     4387.
    -3143.     6325.    -4877.      195.     -496.     1732.    16910.    -3611.
    -5082.    -3918.    -5548.    -2748.     5597.     3206.    -8721.    -7531.
     1112.    15922.     2935.    -9695.    -2051.    18043.     2013.     -306.
    -5539.    -9017.   -18840.    10401.     3189.   -18287.    -3289.    -4301.
    -2925.    -3157.    10105.     3154.    -7270.   -11441.     2041.    -3746.
     3139.    -8121.   -12835.     9925.     2178.    -6024.    -8531.     4158.
    -4861.    -8923.    18110.    13781.    10607.     5129.    15787.     3428.
    -1069.    -2068.   -12438.    12253.    -7699.   -10595.     6591.    -1159.
    -1575.     2816.    18167.     1954.    -2797.    -6039.    17566.     2517.
```

Real Univac 1108 Result

```
   .85580000+04    .16432637+06    .30633514+05    .12878585+06   -.43274833+05
  -.23935650+05   -.88190126+05   -.79698125+05   -.32208377+05    .12423822+06
   .10475777+06    .12018498+06   -.89308050+05   -.31424301+05   -.18619060+04
   .40882889+05   -.92820702+05   -.11257891+05   -.14610218+05   -.12452422+06
  -.79964560+04   -.21300992+05    .11725652+05   -.11268548+06    .68437002+04
   .19196831+06    .36610626+05    .12051347+06    .46726594+05   -.14447113+05
  -.88314708+05    .60155586+05   -.29508836+05    .10651084+06    .62851124+05
   .54568208+05    .10175734+06   -.81431963+04   -.35035923+04    .59602295+02
   .50177875+05    .23466077+05   -.65641566+05   -.11844547+06   -.12877611+06
   .62304680+05    .12676695+06    .95429297+04   -.22496512+05   -.12558822+06
  -.75575159+05   -.50596355+05   -.94544403+05    .23490922+05   -.11136791+06
   .36708912+04    .11267843+06   -.17295822+06    .24522155+05    .18504945+05
   .86965552+05    .57614010+05    .17787883+06   -.13490052+06   -.10801400+06
   .54710958+05    .28340147+05   -.28900264+06   -.70621265+05    .95541438+05
  -.72483458+05    .93026729+05   -.42655679+05   -.87760739+05    .55541602+05
  -.66131058+05   -.12314756+06   -.46119934+05    .63107815+05   -.35125625+05
   .11841725+06    .21500096+06    .13246392+06    .18619414+05   -.12489943+06
   .41877700+05   -.19586924+06    .35511073+05   -.18944333+06   -.49808327+05
  -.10685043+06    .39295759+05   -.34118988+05    .87388329+05   -.56760156+05
   .10627192+06    .17858835+05   -.61932731+05    .48561633+05   -.11576040+06
   .25315634+05    .12305369+06   -.93928612+05   -.35586001+05    .75343836+05
   .28277265+05   -.11360036+06   -.38144223+05   -.20365177+05   -.24503150+05
  -.86787733+05    .63162532+05   -.84544034+05    .14108025+06   -.25588855+05
   .67912168+04    .62923921+05   -.22494207+06    .12123530+06    .59887308+05

   .19538353+06   -.19840547+04    .59866334+05   -.42466545+05    .19907523+06
  -.35014947+05   -.44793332+05    .32308614+04    .11306600+06    .32308614+04
  -.44793332+05   -.35014947+05    .19907523+06   -.42466545+05    .59866334+05
  -.19840547+04    .19538353+06    .59887308+05    .12123530+06   -.22494207+06
   .62923921+05    .67912168+04   -.25588855+05    .14108025+06   -.84544034+05
   .63162532+05   -.86787733+05   -.24503150+05   -.20365177+05   -.38144223+05
  -.11360036+06    .28277265+05    .75343836+05   -.35586001+05   -.93928612+05
   .12305369+06    .25315634+05   -.11576040+06    .48561633+05   -.61932731+05
   .17858835+05    .10627192+06   -.56760156+05    .87388329+05   -.34118988+05
   .39295759+05   -.10685043+06   -.49808327+05   -.18944333+06    .35511073+05
  -.19586924+06    .41877700+05   -.12489943+06    .18619414+05    .13246392+06
   .21500096+06    .11841725+06   -.35125625+05    .63107815+05   -.46119934+05
  -.12314756+06   -.66131058+05    .55541602+05   -.87760739+05   -.42655679+05
   .93026729+05   -.72483458+05    .95541438+05   -.70621265+05   -.28900264+06
   .28340147+05    .54710958+05   -.10801400+06   -.13490052+06    .17787883+06
   .57614010+05    .86965552+05    .18504945+05    .24522155+05   -.17295822+06
   .11267843+06    .36708912+04   -.11136791+06    .23490922+05   -.94544403+05
  -.50596355+05   -.75575159+05   -.12558822+06   -.22496512+05    .95429297+04
   .12676695+06    .62304680+05   -.12877611+06   -.11844547+06   -.65641566+05
   .23466077+05    .50177875+05    .59602295+02   -.35035923+04   -.81431963+04
   .10175734+06    .54568208+05    .62851124+05    .10651084+06   -.29508836+05
   .60155586+05   -.88314708+05   -.14447113+05    .46726594+05    .12051347+06
   .36610626+05    .19196831+06    .68437002+04   -.11268548+06    .11725652+05
  -.21300992+05   -.79964560+04   -.12452422+06   -.14610218+05   -.11257891+05
  -.92820702+05    .40882889+05   -.18619060+04   -.31424301+05   -.89308050+05
   .12018498+06    .10475777+06    .12423822+06   -.32208377+05   -.79698125+05
  -.88190126+05   -.23935650+05   -.43274833+05    .12878585+06    .30633514+05
   .16432637+06
```

Imaginary Univac 1108 Result

```
   .00000000     -.32348535+03    -.15282431+06    -.27383049+05     .79234054+05
  .12539749+05    .94287190+05     .34464606+05    -.41893783+05     .60283592+05
 -.71297144+05   -.15715059+06    -.96626729+05     .51172803+05    -.20518874+05
  .18855686+06    .29928419+06     .47544088+05     .49069532+05    -.76239434+05
  .17435297+06    .21966100+05    -.10578316+06     .11941393+05    -.32980707+05
  .35799352+05   -.12412578+05    -.48219366+05     .46967453+05     .36626935+05
 -.62671231+05    .14840741+06     .64964977+05    -.17344318+06    -.21326760+05
 -.11224059+06    .46812337+05    -.10760260+06    -.13977412+06     .45146798+05
 -.92065512+04   -.88596548+05     .44764577+05     .47270071+05     .94768170+05
 -.83311300+05   -.72885594+05     .95072141+05     .24357188+06    -.72368868+05
 -.21239087+05   -.13493994+04     .19846301+06     .31240915+05     .44790895+05
 -.78648652+04    .12952341+06     .93203335+05     .14410096+06     .10436177+06
 -.11474280+06    .10571027+06     .23333460+05    -.32214113+05     .23472000+06
  .10789059+06   -.33982734+05    -.15458620+05    -.24644316+06     .28431246+05
 -.57622153+05   -.44369019+05     .35402116+05    -.31654141+06     .17098704+05
  .18087399+05    .88054139+05     .15852791+06    -.22730289+06    -.47605405+04
 -.17384096+06    .50385053+05     .23991060+05    -.13077045+06    -.63503576+05
 -.97109678+05    .15650693+06    -.79624333+05     .10901463+06    -.19423950+05
 -.38454852+05    .20013107+06    -.25320520+05    -.47217832+05    -.48101416+04
 -.48075970+05   -.22981703+06    -.14226242+06     .32640422+05     .74105499+05
  .30471259+05   -.10316574+06     .36430140+05    -.12036483+06     .54591791+05
  .11034351+06   -.16088750+05     .11887141+06     .40587319+05    -.44664971+04
  .56991673+05   -.54843035+05    -.74552653+05    -.11836559+06    -.12846535+06
 -.73153277+05   -.73496373+05     .18890718+06     .12702667+06    -.19050124+05
 -.14147976+06    .75189292+05    -.92939891+05     .10989458+06    -.33364525+05
 -.20342460+06   -.34484995+05     .12785482+06    -.00000000       -.12785482+06
  .34484995+05    .20342460+06     .33364525+05    -.10989458+06     .92939891+05
 -.75189292+05    .14147976+06     .19050124+05    -.12702667+06    -.18890718+06

  .73496373+05    .73153277+05     .12846535+06     .11836559+06     .74552653+05
  .54843035+05   -.56991673+05     .44664971+04    -.40587319+05    -.11887141+06
  .16088750+05   -.11034351+06    -.54591791+05     .12036483+06    -.36430140+05
  .10316574+06   -.30471259+05    -.74105499+05    -.32640422+05     .14226242+06
  .22981703+06    .48075970+05     .48101416+04     .47217832+05     .25320520+05
 -.20013107+06    .38454852+05     .19423950+05    -.10901463+06     .79624333+05
 -.15650693+06    .97109678+05     .63503576+05     .13077045+06    -.23991060+05
 -.50385053+05    .17384096+06     .47605405+04     .22730289+06    -.15852791+06
 -.88054139+05   -.18087399+05    -.17098704+05     .31654141+06    -.35402116+05
  .44369019+05    .57622153+05    -.28431246+05     .24644316+06     .15458620+05
  .33982734+05   -.10789059+06    -.23472000+06     .32214113+05    -.23333460+05
 -.10571027+06    .11474280+06    -.10436177+06    -.14410096+06    -.93203335+05
 -.12952341+06    .78648652+04    -.44790895+05    -.31240915+05    -.19846301+06
  .13493994+04    .21239087+05     .72368868+05    -.24357188+06    -.95072141+05
  .72885594+05    .83311300+05    -.94768170+05    -.47270071+05    -.44764577+05
  .88596548+05    .92065512+04    -.45146798+05     .13977412+06     .10760260+06
 -.46812337+05    .11224059+06     .21326760+05     .17344318+06    -.64964977+05
 -.14840741+06    .62671231+05    -.36626935+05    -.46967453+05     .48219366+05
  .12412578+05   -.35799352+05     .32980707+05    -.11941393+05     .10578316+06
 -.21966100+05   -.17435297+06     .76239434+05    -.49069532+05    -.47544088+05
 -.29928419+06   -.18855686+06     .20518874+05    -.51172803+05     .96626729+05
  .15715059+06    .71297144+05    -.60283592+05     .41893783+05    -.34464606+05
 -.94287190+05   -.12539749+05    -.79234054+05     .27383049+05     .15282431+06
  .32348535+03
```

Main Program of ILLIAC IV Simulation

(Data stored in external file referenced by ABC)

```
BEGIN
             EXTERNAL      FFT,ABC;
SIZE:        DATA          256;
NU:          DATA          1;
MAIN::       SETE          E.OR.-E;
             SETE1         E.OR.-E;
             SLIT(2)       ARGLIST;
             CLC(3)
             SLIT(3)       FFT;
             EXCHL(3)      $ICR;
             LIT(0)        1.ABC+600 .ABC;
             DISPLAYR      $C0.20:8;
             HALT;
ARGLIST:     DATA          SIZE,NU,ABC;
```

Illiac IV Simulation Results

(Outer = Real, Inner = Imaginary)

```
STEVENS/TEST
***SSK/<SK=2505  START 19:06:51:21 VERSION= 1.25    STEVENS/TEST $$$


FIRST LOCATION EXECUTED:   002002 0:16

%%%%% DISPLAYS #          1      ICR=  002007 0:16  TIME=19:07:32:07      ELAPSED PROCESSOR TIME=00:00:25:34%%%%%%%%%%%%%%%%%%%%%%%%%%%

MEMORY:

------

LOCATION:8  C(LOCATION)                     C(LOCATION +0000 01:8)            C(LOCATION +0000 02:8)            C(LOCATION +0000 03:8)
0201 00(OUTER)=+.85580000@+04          (OUTER)=+.16432633@+06        (OUTER)=+.30633516@+05        (OUTER)=+.12078587@+06
       (INNER)=0                       (INNER)=+.32347556@+03        (INNER)=+.15282428@+06        (INNER)=+.27383047@+05
0201 04(OUTER)=-.43274852@+05          (OUTER)=-.23935660@+05        (OUTER)=-.88190102@+05        (OUTER)=-.79698117@+05
       (INNER)=-.79234047@+05          (INNER)=-.12539762@+05        (INNER)=-.94287211@+05        (INNER)=-.34464617@+05
0201 10(OUTER)=-.32208367@+05          (OUTER)=+.12423820@+06        (OUTER)=+.10475777@+06        (OUTER)=+.12018495@+06
       (INNER)=+.41803739@+05          (INNER)=-.60283586@+05        (INNER)=+.71297133@+05        (INNER)=+.15715061@+06
0201 14(OUTER)=-.33303039@+05          (OUTER)=-.31424100@+05        (OUTER)=-.13619121@+04        (OUTER)=+.40382891@+05
       (INNER)=+.96626727@+05          (INNER)=-.51172672@+05        (INNER)=+.20519867@+05        (INNER)=-.14955693@+06
0201 20(OUTER)=-.92820703@+05          (OUTER)=-.11257891@+05        (OUTER)=-.14610219@+05        (OUTER)=-.12452422@+05
       (INNER)=-.29923416@+06          (INNER)=-.47044074@+05        (INNER)=-.49069516@+05        (INNER)=-.76239427@+05
0201 24(OUTER)=-.79964070@+04          (OUTER)=-.21300980@+05        (OUTER)=+.11725664@+05        (OUTER)=-.11268548@+06
       (INNER)=-.17435293@+06          (INNER)=-.21966094@+05        (INNER)=+.10578316@+06        (INNER)=-.11941395@+05
0201 30(OUTER)=+.68437070@+04          (OUTER)=+.19196828@+06        (OUTER)=+.36610641@+05        (OUTER)=+.12051346@+06
       (INNER)=+.32930695@+05          (INNER)=-.35799352@+05        (INNER)=+.12412582@+05        (INNER)=+.43219375@+05
0201 34(OUTER)=+.45726005@+05          (OUTER)=-.14446836@+05        (OUTER)=-.88314703@+05        (OUTER)=-.60155586@+05
       (INNER)=-.46967457@+05          (INNER)=-.36626918@+05        (INNER)=+.62671211@+05        (INNER)=-.14840738@+06
0201 40(OUTER)=-.22598040@+05          (OUTER)=+.10651084@+06        (OUTER)=+.62851129@+05        (OUTER)=-.54968195@+05
       (INNER)=-.64964942@+05          (INNER)=+.17344316@+06        (INNER)=+.21326750@+05        (INNER)=+.11224059@+06
0201 44(OUTER)=+.10175734@+05          (OUTER)=-.81432129@+04        (OUTER)=-.35035898@+04        (OUTER)=+.59011328@+02
       (INNER)=-.46912340@+05          (INNER)=+.10760261@+06        (INNER)=+.13977409@+06        (INNER)=-.45146770@+05
0201 50(OUTER)=+.50177875@+05          (OUTER)=+.23466082@+05        (OUTER)=-.65641578@+05        (OUTER)=-.11044545@+06
       (INNER)=+.92165352@+04          (INNER)=+.88596563@+05        (INNER)=-.44764578@+05        (INNER)=-.47270055@+05
0201 54(OUTER)=-.12877607@+06          (OUTER)=+.62304375@+05        (OUTER)=+.12676694@+06        (OUTER)=+.95429219@+04
       (INNER)=-.94768134@+05          (INNER)=+.83311210@+05        (INNER)=+.72885602@+05        (INNER)=-.95072125@+05
0201 60(OUTER)=-.22426513@+05          (OUTER)=-.12558622@+06        (OUTER)=-.75575148@+05        (OUTER)=-.50596328@+05
       (INNER)=-.24357180@+06          (INNER)=+.72368875@+05        (INNER)=+.21239086@+05        (INNER)=+.13493984@+04
0201 64(OUTER)=-.94544394@+05          (OUTER)=+.23490941@+05        (OUTER)=-.11136790@+06        (OUTER)=+.36708789@+04
       (INNER)=-.19346290@+05          (INNER)=-.31240908@+05        (INNER)=-.44790891@+05        (INNER)=+.73648594@+04
0201 70(OUTER)=+.11267843@+06          (OUTER)=-.17295819@+06        (OUTER)=+.24522129@+05        (OUTER)=+.13504961@+05
       (INNER)=-.12952341@+06          (INNER)=-.93203320@+05        (INNER)=-.14410094@+06        (INNER)=-.10436176@+06
0201 74(OUTER)=+.56935555@+05          (OUTER)=+.57614030@+05        (OUTER)=+.17797330@+06        (OUTER)=-.13490048@+06
       (INNER)=+.11474275@+06          (INNER)=-.10571008@+06        (INNER)=-.23333457@+05        (INNER)=+.32214109@+05
0202 00(OUTER)=-.19401400@+06          (OUTER)=+.54710945@+05        (OUTER)=+.28340133@+05        (OUTER)=-.29900259@+06
       (INNER)=-.23472000@+06          (INNER)=-.10789058@+06        (INNER)=+.33982730@+05        (INNER)=+.15454605@+05
0202 04(OUTER)=-.70621253@+05          (OUTER)=+.95541445@+05        (OUTER)=-.72483461@+05        (OUTER)=+.93026719@+05
       (INNER)=+.24644313@+06          (INNER)=-.28431242@+05        (INNER)=+.57622164@+05        (INNER)=+.43368992@+05
0202 10(OUTER)=-.42655680@+05          (OUTER)=-.87760719@+05        (OUTER)=+.55541586@+05        (OUTER)=-.66131039@+05
       (INNER)=-.35402125@+05          (INNER)=+.31654134@+06        (INNER)=-.17098703@+05        (INNER)=-.13087408@+05
0202 14(OUTER)=-.12314755@+06          (OUTER)=-.46119465@+05        (OUTER)=+.63107789@+05        (OUTER)=-.35125641@+05
       (INNER)=-.48054141@+05          (INNER)=-.15652756@+06        (INNER)=+.22730288@+06        (INNER)=+.47005469@+04
0202 20(OUTER)=+.11041726@+06          (OUTER)=+.21500095@+06        (OUTER)=+.13246389@+06        (OUTER)=+.13019410@+05
       (INNER)=+.17384095@+06          (INNER)=-.50385047@+05        (INNER)=-.23991055@+05        (INNER)=+.13077045@+06
0202 24(OUTER)=-.12499741@+06          (OUTER)=+.41877680@+05        (OUTER)=-.19586923@+06        (OUTER)=+.35511066@+05
       (INNER)=+.63503570@+05          (INNER)=+.97109680@+05        (INNER)=-.15650691@+06        (INNER)=+.79024344@+05
0202 30(OUTER)=-.16944331@+06          (OUTER)=-.49083313@+05        (OUTER)=-.10685041@+06        (OUTER)=+.39295758@+05
       (INNER)=-.10901463@+06          (INNER)=+.19423955@+05        (INNER)=+.38454852@+05        (INNER)=-.20013108@+06
0202 34(OUTER)=-.34118934@+05          (OUTER)=+.87388188@+05        (OUTER)=-.56760152@+05        (OUTER)=+.10627191@+06
       (INNER)=+.25320504@+05          (INNER)=+.47217914@+05        (INNER)=+.48101328@+04        (INNER)=+.48075992@+05
```

```
0202 40(OUTER)=+.17858840@+05      (OUTER)=-.61932742@+05     (OUTER)=+.48561641@+05     (OUTER)=-.11576038@+06
          (INNER)=+.22981702@+06   (INNER)=+.14226241@+06     (INNER)=-.32640430@+05     (INNER)=-.74105492@+05
0202 44(OUTER)=+.25315621@+05      (OUTER)=+.12305368@+06     (OUTER)=-.93928602@+05     (OUTER)=-.35585992@+05
          (INNER)=-.30471270@+05   (INNER)=+.10316570@+06     (INNER)=-.36430145@+05     (INNER)=+.12036480@+06
0202 50(OUTER)=+.75343820@+05      (OUTER)=+.28277258@+05     (OUTER)=-.11360035@+06     (OUTER)=-.38144254@+05
          (INNER)=-.54521797@+05   (INNER)=-.11034349@+06     (INNER)=+.16088754@+05     (INNER)=-.11887141@+06
0202 54(OUTER)=-.20365199@+05      (OUTER)=-.24503148@+05     (OUTER)=-.86787734@+05     (OUTER)=+.63162516@+05
          (INNER)=-.40587305@+05   (INNER)=+.44671406@+04     (INNER)=-.56991695@+05     (INNER)=+.54843012@+05
0202 60(OUTER)=-.84544047@+05      (OUTER)=+.14108025@+06     (OUTER)=-.25588826@+05     (OUTER)=+.67911953@+04
          (INNER)=+.74552656@+05   (INNER)=+.11834558@+06     (INNER)=+.12846535@+06     (INNER)=+.73153281@+05
0202 64(OUTER)=+.62923914@+05      (OUTER)=-.22494208@+06     (OUTER)=+.12123530@+06     (OUTER)=+.59887305@+05
          (INNER)=+.73496383@+05   (INNER)=-.18890714@+06     (INNER)=-.12702666@+06     (INNER)=+.19050133@+05
0202 70(OUTER)=+.19538352@+06      (OUTER)=-.17840391@+04     (OUTER)=+.59866336@+05     (OUTER)=-.42466512@+05
          (INNER)=+.14147973@+06   (INNER)=-.75189297@+05     (INNER)=+.92939898@+05     (INNER)=-.10989459@+06
0202 74(OUTER)=+.19907522@+06      (OUTER)=-.35014648@+05     (OUTER)=-.44793336@+05     (OUTER)=+.32308672@+04
          (INNER)=+.33364516@+05   (INNER)=+.20342389@+06     (INNER)=+.34484992@+05     (INNER)=-.12785480@+06
0203 00(OUTER)=+.11306600@+06      (OUTER)=+.32308672@+04     (OUTER)=-.44793336@+05     (OUTER)=-.35014730@+05
          (INNER)=0                (INNER)=+.12795440@+06     (INNER)=-.34484992@+05     (INNER)=-.20342458@+06
0203 04(OUTER)=+.19907522@+06      (OUTER)=-.42466512@+05     (OUTER)=+.59866336@+05     (OUTER)=-.17840391@+04
          (INNER)=-.33364516@+05   (INNER)=+.10989459@+06     (INNER)=-.92939898@+05     (INNER)=+.75189297@+05
0203 10(OUTER)=+.19538352@+06      (OUTER)=+.59887305@+05     (OUTER)=+.12123530@+06     (OUTER)=-.22494208@+06
          (INNER)=-.14147973@+06   (INNER)=-.19050133@+05     (INNER)=+.12702666@+06     (INNER)=+.18890714@+06
0203 14(OUTER)=+.62923914@+05      (OUTER)=+.67914004@+04     (OUTER)=-.25588826@+05     (OUTER)=+.14108025@+06
          (INNER)=-.73496383@+05   (INNER)=-.73153203@+05     (INNER)=-.12846535@+06     (INNER)=-.11836558@+06
0203 20(OUTER)=-.84544047@+05      (OUTER)=+.63162516@+05     (OUTER)=-.86787734@+05     (OUTER)=-.24503141@+05
          (INNER)=-.74552656@+05   (INNER)=-.54843012@+05     (INNER)=+.56991695@+05     (INNER)=-.44665078@+04
0203 24(OUTER)=-.20365199@+05      (OUTER)=-.38144254@+05     (OUTER)=-.11360035@+06     (OUTER)=+.28277258@+05
          (INNER)=+.40587305@+05   (INNER)=+.11887141@+06     (INNER)=-.16088754@+05     (INNER)=+.11034349@+06
0203 30(OUTER)=+.75343820@+05      (OUTER)=-.35585992@+05     (OUTER)=-.93928602@+05     (OUTER)=+.12305368@+06
          (INNER)=+.54521797@+05   (INNER)=-.12036480@+06     (INNER)=+.36430145@+05     (INNER)=-.10316570@+06
0203 34(OUTER)=+.25315621@+05      (OUTER)=-.11576016@+06     (OUTER)=+.48561641@+05     (OUTER)=-.61932742@+05
          (INNER)=+.30471270@+05   (INNER)=+.74105148@+05     (INNER)=+.32640430@+05     (INNER)=-.14226241@+06
0203 40(OUTER)=+.17858840@+05      (OUTER)=+.10627191@+06     (OUTER)=-.56760152@+05     (OUTER)=+.87388320@+05
          (INNER)=-.22981702@+06   (INNER)=-.48075992@+05     (INNER)=-.48101328@+04     (INNER)=-.47217844@+05
0203 44(OUTER)=-.34118904@+05      (OUTER)=+.39295758@+05     (OUTER)=-.10685041@+06     (OUTER)=-.49608313@+05
          (INNER)=-.25329504@+05   (INNER)=+.20013108@+06     (INNER)=-.38454852@+05     (INNER)=-.19423955@+05
0203 50(OUTER)=-.18904331@+06      (OUTER)=+.35511066@+05     (OUTER)=-.19586923@+06     (OUTER)=+.41077680@+05
          (INNER)=+.10901463@+06   (INNER)=-.79624344@+05     (INNER)=+.15650691@+06     (INNER)=-.97109680@+05
0203 54(OUTER)=-.12489941@+06      (OUTER)=+.18619297@+05     (OUTER)=+.13246389@+06     (OUTER)=+.21500095@+06
          (INNER)=-.63503570@+05   (INNER)=-.13076994@+06     (INNER)=+.23991055@+05     (INNER)=+.50385047@+05
0203 60(OUTER)=+.11831726@+06      (OUTER)=-.35125541@+05     (OUTER)=+.63107789@+05     (OUTER)=-.46119945@+05
          (INNER)=-.17343095@+06   (INNER)=-.47605469@+04     (INNER)=-.22730284@+06     (INNER)=+.15552780@+06
0203 64(OUTER)=-.12314755@+06      (OUTER)=-.66131039@+05     (OUTER)=+.55541586@+05     (OUTER)=-.87760719@+05
          (INNER)=+.88054141@+05   (INNER)=+.18087408@+05     (INNER)=+.17098703@+05     (INNER)=-.31654134@+06
0203 70(OUTER)=-.42655680@+05      (OUTER)=+.93026719@+05     (OUTER)=-.72483461@+05     (OUTER)=+.95541445@+05
          (INNER)=+.35402125@+05   (INNER)=-.44368992@+05     (INNER)=-.57622164@+05     (INNER)=+.24431242@+05
0203 74(OUTER)=-.70621258@+05      (OUTER)=-.28900209@+06     (OUTER)=+.28340133@+05     (OUTER)=+.54710945@+05
          (INNER)=-.24644313@+06   (INNER)=-.15458855@+05     (INNER)=-.33982730@+05     (INNER)=+.10789058@+06
0204 00(OUTER)=-.10401400@+06      (OUTER)=-.13490048@+06     (OUTER)=+.17787880@+06     (OUTER)=+.57613988@+05
          (INNER)=+.23472000@+06   (INNER)=-.32214109@+05     (INNER)=+.23333457@+05     (INNER)=+.10571028@+06
0204 04(OUTER)=+.80965555@+05      (OUTER)=+.18504961@+05     (OUTER)=+.24522129@+05     (OUTER)=-.17295810@+06
          (INNER)=-.11474278@+06   (INNER)=+.10436176@+06     (INNER)=+.14410094@+06     (INNER)=+.93203320@+05
0204 10(OUTER)=+.11267843@+06      (OUTER)=+.36709789@+04     (OUTER)=-.11136790@+06     (OUTER)=+.23490941@+05
          (INNER)=+.12952341@+06   (INNER)=-.78648594@+04     (INNER)=+.44790891@+05     (INNER)=+.31240908@+05
0204 14(OUTER)=-.94544398@+05      (OUTER)=-.50596379@+05     (OUTER)=-.75575143@+05     (OUTER)=-.12558822@+06
          (INNER)=+.19846298@+06   (INNER)=-.13492500@+04     (INNER)=-.21239086@+05     (INNER)=-.72368875@+05
0204 20(OUTER)=-.22496504@+05      (OUTER)=+.95429219@+04     (OUTER)=+.12676694@+06     (OUTER)=+.62304676@+05
          (INNER)=+.24357186@+06   (INNER)=+.95072125@+05     (INNER)=-.72085602@+05     (INNER)=-.83311305@+05
0204 24(OUTER)=-.12877609@+06      (OUTER)=-.11844545@+06     (OUTER)=-.65541578@+05     (OUTER)=+.23466087@+05
          (INNER)=+.94764148@+05   (INNER)=+.47270055@+05     (INNER)=+.44764578@+05     (INNER)=-.83396563@+05
0204 30(OUTER)=+.50177875@+05      (OUTER)=+.59611328@+02     (OUTER)=-.35035898@+04     (OUTER)=-.81432123@+04
```

```
        (INNER)=-.92065352@+04     (INNER)=+.45146770@+05     (INNER)=-.13977409@+06     (INNER)=-.10760261@+06
0204 34(OUTER)=+.10175734@+06     (OUTER)=+.54567813@+05     (OUTER)=+.62851129@+05     (OUTER)=+.10651084@+06
        (INNER)=+.45812340@+05     (INNER)=-.11224054@+06     (INNER)=-.21326750@+05     (INNER)=-.17344316@+06
0204 40(OUTER)=-.22508840@+05     (OUTER)=+.60155586@+05     (OUTER)=-.88314703@+05     (OUTER)=-.14447102@+05
        (INNER)=+.64964944@+05     (INNER)=+.14840738@+06     (INNER)=-.62671211@+05     (INNER)=+.36626934@+05
0204 44(OUTER)=+.46726605@+05     (OUTER)=+.12051346@+06     (OUTER)=+.36610641@+05     (OUTER)=+.19196828@+06
        (INNER)=+.46367457@+05     (INNER)=-.48219375@+05     (INNER)=-.12412582@+05     (INNER)=+.35799352@+05
0204 50(OUTER)=+.68437070@+04     (OUTER)=-.11268548@+06     (OUTER)=+.11725664@+05     (OUTER)=-.21300980@+05
        (INNER)=-.32440595@+05     (INNER)=+.11941395@+05     (INNER)=-.10578316@+06     (INNER)=+.21966094@+05
0204 54(OUTER)=-.79364570@+04     (OUTER)=-.12452413@+06     (OUTER)=-.14610219@+05     (OUTER)=-.11257891@+05
        (INNER)=+.17435295@+06     (INNER)=-.76239305@+05     (INNER)=+.49069516@+05     (INNER)=+.47544074@+05
0204 60(OUTER)=-.22820703@+05     (OUTER)=+.40882891@+05     (OUTER)=-.18619121@+04     (OUTER)=-.31424301@+05
        (INNER)=+.29992416@+06     (INNER)=+.18855683@+06     (INNER)=-.20518867@+05     (INNER)=+.51172793@+05
0204 64(OUTER)=-.39308039@+05     (OUTER)=+.12018495@+06     (OUTER)=+.10475777@+06     (OUTER)=+.12423820@+06
        (INNER)=-.96626727@+05     (INNER)=-.15715061@+06     (INNER)=-.71207133@+05     (INNER)=+.60283586@+05
0204 70(OUTER)=-.32208367@+05     (OUTER)=-.79698117@+05     (OUTER)=-.88190102@+05     (OUTER)=-.23935660@+05
        (INNER)=-.41893789@+05     (INNER)=+.34464617@+05     (INNER)=+.94287211@+05     (INNER)=+.12539762@+05
0204 74(OUTER)=-.43274852@+05     (OUTER)=+.12878590@+06     (OUTER)=+.30633516@+05     (OUTER)=+.16432633@+06
        (INNER)=+.79234047@+05     (INNER)=-.27382734@+05     (INNER)=-.15282428@+06     (INNER)=-.32347656@+03
```

APPENDIX C

Timing Summary †

SIZE OF DATA SET	TIME FOR SCRAMBLING	TIME FOR FFT	TOTAL	$\frac{N \log N}{64} \times 3\mu$ *
4096	75.0 μ sec	2233.6 μ sec	2308.6 μ sec	2304 μ sec
2048	48.4 μ sec	1038.4 μ sec	1096.8 μ sec	1056 μ sec
1024	53.1 μ sec	480.0 μ sec	533.1 μ sec	480 μ sec
512	23.2 μ sec	220.4 μ sec	243.4 μ sec	216 μ sec
256	17.4 μ sec	100.4 μ sec	117.8 μ sec	96 μ sec
128	10.5 μ sec	45.35 μ sec	55.85 μ sec	42 μ sec
64	4.65 μ sec	20.6 μ sec	25.25 μ sec	18 μ sec
2 x 32	2.80 μ sec	15.65 μ sec	18.45 μ sec	15 μ sec
4 x 16	2.40 μ sec	11.10 μ sec	13.50 μ sec	12 μ sec
8 x 8	1.75 μ sec	6.75 μ sec	8.50 μ sec	9 μ sec
Overhead			13 μ sec	

---

* This formula gives a rough estimate of the time required to do the FFT portion where each complex multiply operation takes 3 μ sec.

† All times base on 50 n sec clock.

APPENDIX D

SUBROUTINE LISTING

```
LISTED ON: 08/13/71  AT: 09:14

     BEGIN COMMENT
          THE FAST FOURIER TRANSFORM SUBROUTINE IS COMPOSED OF FOUR
     SEPARATE LOAD MODULES.  THE FIRST IS THE MAIN PROGRAM WHICH HANDLES
     THE SUBROUTINE LINKAGE AND SETS UP THE CU REGISTERS.  NEXT COMES
     THE SCRAMBLING ROUTINE WHICH REORDERS THE INPUT DATA SO THAT IT
     CAN BE TRANSFORMED.  THEN COMES THE TRANSFORM ROUTINE WHICH
     COMPLETES THE TRANSFORM. PERFORMS AN INVERSE IF DESIRED, AND
     RETURNS CONTROL TO THE CALLING PROGRAM.  FINALLY ALL OF THE
     CONSTANTS USED FOR THE TRANSFORM ARE COMPILED IN A SEPARATE DATA
     FILE.

          THREE PARAMETERS ARE REQUIRED BY THIS PROGRAM.  THEY ARE
     LISTED BELOW IN THE ORDER IN WHICH THEY MUST BE SUPPLIED.

     1) SIZE - THE NUMBER OF DATA POINTS PER DATA SET.
     2) NUMBER - THE NUMBER OF DATA SETS TO BE TRANSFORMED.
     3) POINTER - THE CU(24 BIT) ADDRESS OF THE FIRST LOCATION OF DATA.

          A USERS MANUAL FOR THIS SUBROUTINE HAS BEEN PUBLISHED AND IS
     TITLED ?THE ILLIAC IV FAST FOURIER TRANSFORM SUBROUTINE USERS
     MANUAL" BY JAMES E STEVENS JR.

          THE MAIN PORTION OF THE PROGRAM FOLLOWS AND SETS UP CU
     REGISTERS WITH CONTROL INFORMATION AS SHOWN BELOW.
     LOCAL DATA BUFFER (LDB OR ADB)

     D0(PATTERN)
          IN THE SCRAMBLE SECTION OF THE PROGRAM THIS REGISTER IS
     USED FOR SAVING MODE PATTERNS

     D1(FLAG)
          THIS REGISTER IS DIVIDED INTO THREE FIELDS.  THE FIRST. BIT
     NUMBER 15. IS USED TO INDICATE WHETHER AN INVERSE TRANSFORM HAS
     BEEN REQUESTED.  IF BIT 15 IS SET TO ONE AN INVERSE TRANSFORM WILL
     BE PERFORMED.  THE SECOND FIELD USES THE NEXT 24 BITS AND IS SIMPLY
     A COPY OF THE FIRST PARAMETER. SIZE. WHICH GIVES THE NUMBER OF DATA
     POINTS IN EACH INPUT DATA SET.  THE VARIABLE. SIZE. MUST BE AN EVEN
     POWER OF TWO.  IF THE PARAMETER. SIZE. PASSED TO THE SUBROUTINE IS
     NOT A POWER OF TWO IT WILL BE ASSUMED THAT THE NEXT LARGEST POWER
     OF TWO IS DESIRED AND THE PROPER VALUE IS THEN GENERATED.  THE
     FINAL 24 BIT FIELD OF THE FLAG REGISTER IS LOADED WITH THE PE
     ADDRESS OF THE FIRST LOCATION OF THE DATA SET TO BE TRANSFORMED.
     THIS ADDRESS IS EQUAL TO THE THIRD PARAMETER. POINTER. DIVIDED BY
     64 SO IT IS A PE ADDRESS.
          LATER. IN THE TRANSFORM SECTION OF THIS PROGRAM. THE
     INFORMATION IN REGISTER D1(FLAG) IS LOADED INTO ACAR0 AND REFERED
     TO BY THE NAME SIZE.

     D2(RETURN)
          THIS REGISTER IS USED TO SAVE THE RETURN ADDRESS FOUND IN
     ACAR3 WHEN THE SUBROUTINE IS ENTERED.  THE ACTUAL RETURN ALSO USES
     ACAR3.  AN ERROR RETURN IS FLAGGED BY FILLING ACAR2 WITH ALL ONES.

     D3 AND D4
          SAVE AREA FOR ACAR0 AND ACAR1.

     ACAR0(SIZE.INDEX)
          IN THE MAIN SECTION OF THE SUBROUTINE THIS REGISTER IS LOADED
     WITH THE VALUE SIZE (THE FIRST PARAMETER).  AFTER THE SIZE IS USED
```

TO MAKE A CONDITIONAL BRANCH THIS REGISTER IS USED TO HOLD INDEXING INFORMATION DURING THE SCRAMBLING PROCESS. AS NOTED ABOVE, FOR THE TRANSFORM SECTION OF THIS PROGRAM THIS REGISTER IS LOADED FROM REGISTER D1(FLAG) AND REFERRED TO AGAIN BY THE NAME SIZE.

ACAR1(MODE)
THIS REGISTER IS NOT FILLED WITH ANY INITIAL VALUE AND IS USED THROUGHOUT THE SUBROUTINE FOR SAVING MODE PATTERNS FOR ENABLING PES

ACAR2(LOOP)
THIS REGISTER IS NOT FILLED WITH ANY INITIAL VALUE AND IS USE THROUGHOUT THE SUBROUTINE FOR LOOP CONTROL INFORMATION.

ACAR3(COUNT)
THIS REGISTER CONTAINS A POINTER TO THE ROW OF DATA BEING CONSIDERED, AS WELL AS A POINTER TO THE LAST ROW OF DATA (A LIMIT). IT IS USED TO INDEX PE MEMORY AND TO DETERMINE WHEN THE TRANSFORM IS COMPLETE. THIS REGISTER IS SET UP BY THE MAIN SECTION OF THE SUBROUTINE AS FOLLOWS: (1) THE INCREMENT FIELD IS SET TO ONE. (2) THE LIMIT FIELD IS SET TO THE ADDRESS OF THE FIRST ROW OF DATA PLUS THE TOTAL NUMBER OF ROWS SUPPLIED (=THE SIZE OF THE DATA SETS TIMES THE NUMBER OF DATA SETS DIVIDED BY 64). (3) THE ADDRESS FIELD (LAST 24 BITS) IS SET TO THE PE ADDRESS OF THE FIRST ROW OF DATA:

```
        EXTERNAL        ERR:
        FILL            16.   *ALIGNMENT FOR NEXT THREE STATEMENTS.
FFTENTRY1:    SLIT(0)    0;                  *SET FLAG FOR NO INVERSE.
        SKIP            .1;                  *SKIP NEXT STATEMENT.
FFTINVENTRY1: SLIT(0)    1;                  *SET FLAG FOR INVERSE.
        STL(0)          $D3;                 *SAVE ACAR0.
        STL(1)          $D4;                 *SAVE ACAR1.
        STL(3)          $D2;                 *SAVE RETURN ADDRESS.
        LOAD(2)         $C3;                 *FETCH FIRST ARGUMENT ADDRESS.
        LOAD(3)         $C3;                 *FETCH FIRST ARGUMENT *SIZE*.
        ALIT(3)         -1;                  *FIND THE SMALLEST POWER OF TWO
        CSHL(3)         1;                   *GREATER THAN OR EQUAL TO SIZE.
        LEADO(3)          :                  *WHAT POWER IS IT.
        CTSBF(3)        55,ERR;              *IF ZERO, RETURN AND FLAG ERROR.
        CRB(3)          55;                  *ERASE LEAD ONE FLAG.
        SLIT(1)         63;                  *ELSE, SUBTRACT BIT NO FROM 63
        CSUB(1)         $C3;                 *GIVING THE PROPER POWR.
        SLIT(3)         =1;                  *RECONSTRUCT SIZE
        CSHL(3)         0(1);                *BY SHIFTING.
        CSHL(0)         24;                  *SHIFT FLAG.
        CADD(0)         $C3;                 *ADD SIZE TO FLAG REGISTER.
        ALIT(2)         1;                   *INCREMENT ARG POINTER.
        LOAD(2)         $C3;                 *FETCH SECOND ARGUMENT ADDRESS.
        LOAD(3)         $C3;                 *FETCH SECOND ARGUMENT *NUMBER*.
        ZERXT(3)        .ERR;                *IF ZERO RETURN AND FLAG ERROR.
        CSHL(3)         -6(1);               *MULT NO BY SIZE AND DIV BY 64.
        ALIT(2)         1;                   *INCREMENT ARG POINTER.
        LOAD(2)         $C2;                 *FETCH THIRD ARGUMENT *POINTER*.
        CSHR(2)         6;                   *DIV BY 64.   (MAKE PE ADDRESS).
        CSHL(0)         24;                  *SHIFT FLAG REGISTER.
        CADD(0)         $C2;                 *ADD POINTER TO FLAG REGISTER.
        STL(0)          $D1;                 *SAVE FLAG REGISTER.
        CSHR(0)         24;                  *LEAVE SIZE IN ADDR FLD OF $C0.
        CADD(3)         $C2;                 *SET COUNT LIMIT.
        ALIT(3)         =-1;                 *ONE LESS FOR LOOP CONTROL.
        CROTR(3)        24;                  *POSITION INCREMENT FIELD.
        SLIT(3)         =1;                  *SET INCREMENT TO ONE.
```

```
          CROTR(3)        16;             $POSITION ADDRESS FIELD.
          SLIT(3)         =0;             $ZERO OUT ADDRESS FIELD.
          CADD(3)         $C2;            $SET ADRESS FIELD TO POINTER.
          SETE            E.OR.-E;        $TURN ON
          SETE1           E.OR.-E;        $        ALL PES.
          CHWS            64;             $WORD SIZE TO 64.
          JUMP            HERE;           $GO TO SCRAMBLING ROUTINE
ERR:      CLC(2)           ;              $ERROR (NO SIZE GIVE).
          COMPC(2)         ;              $SET ACAR 2 TO ALL ONES.
          LDL(0)          $D3;            $RESTORE ACAR0.
          LDL(1)          $D4;            $RESTORE ACAR1.
          LDL(3)          $D2;            $FETCH RETURN ADDRESS.
          EXCHL(3)        $ICR;           $RETURN TO CALLING PROGRAM.
          END             FFT.
```

```
LISTED ON: 08/13/71   AT: 09:14

          BEGIN
%  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %
%  %                SCRAMBLING  SECTION                %  %
%  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %
          EXTERNAL         FFT1,PEINDEX;
          REGC             %SIZE=%C0;
SIZE:     EQU              0;
          REGC             %INDEX=%C0;
INDEX:    EQU              0;
          REGC             %MODE=%C1;
MODE:     EQU              1;
          REGC             %LOOP=%C2;
LOOP:     EQU              2;
          REGC            %COUNT=%C3;
COUNT:    EQU              3;
          REGC             %PATTERN=%D0;
%
%
%         THE PE MEMORY LOCATION PEINDEX CONTAINS A SET OF SIX INDECES
%    FOR EACH PE PACKED INTO ONE WORD(SIX BITS PER INDEX, 36 TOTAL).
%    THE DEFINE NAMED FETCHINDEX WHICH FOLLOWS MASKS OFF     BITS FROM
%    PEINDEX AND STORES THE RESULT IN REGISTER X AND REGISTER S.  THE
%    PARAMETER TO THE DEFINE DETERMINES WHICH BITS OF PEINDEX ARE TO
%    BE FETCHED.
%
  DEFINE  FETCHINDEX       %START;%HOWMANY;=
          LDA              PEINDEX;         %FETCH PACKED INDECES.
          SHABR            63-%START;       %ALIGN TO STARTING BIT POSITION.
          CLRA              ;               %ZERO REGISTER A.
          SHABL            %HOWMANY;        %RECOVER BITS OF INTEREST.
          LDS              %A;              %INDEX TO REG S AND REG X.
          LDX              %S;%%;
%
%
%
%         EACH OPERATION OF THE SCRAMBLING ROUTINE INVOLVES ROUTING DATA
%    FROM A STARTING LOCATION IN ONE PE TO A DESTINATION IN ANOTHER PE,
%    AND RETURNING THE PREVIOUS CONTENTS OF THE DESTINAITON TO THE
%    STARTING LOCATION.  INTECHANGE IS A DEFINE WHICH ACCOMPLISHES THIS
%    IN ALL PES SIMULTANEOUSLY WHEN SUPPLIED WITH THE FOLLOWING INFORMA-
%    TION.  REGISTERS X AND S CONTAIN INDECES FOR STARTING AND DESTINA-
%    TION LOCATIONS RESPECTIVELY.  THE TRANSFERS ARE PATTERNED IN SUCH
%    A  WAY THAT THEY CAN BE ITERATED BY SIMPLY ROUTING THE INDECES.
%    THE CU MODE REG IS ASSUMED TO CONTAIN THE PROPER ENABLING PATTERN.
%    THE ONE INPUT PARAMETER REQUIRED IS THE NUMBER OF ROWS OF STORAGE
%    COMPOSING THE DATA SET.  THIS DEFINE WILL ONLY BE USED IN SCRAMBL-
%    ING DATA SETS LARGER THAN 128.  AND IT COMPLETES THE PORTION OF
%    SCRAMBLING DEALING WITH SQUARE PARTITIONS AS DESCRIBED IN THE FFT
%    DOCUMENT OF SEPTEMBER 1970. ALL PES ARE ASSUMED ENABLED.
%
  DEFINE  INTERCHANGE      %NO ;=
          BEGIN BLOCK                        THE SYMBOL DLOOP MUST BE LOCAL.
          STL(MODE)        %PATTERN;        %SAVE MODE PATTERN.
          LIT(LOOP)        1,%NO/2-1,1;     %LOOP COUNTER.
          LIT(INDEX)       %3;              %ROUTING INDEX(=LOOP COUNT).
DLOOP:    RTL              %S,1;            %DESTINATION INDEX.
          LDS              %R;              %COMPLETE S TO S TRANSFER.
          RTL              %X,-1;           %STARTING INDEX.
          LDX              %R;              %COMPLETE X TO X TRANSFER.
```

```
          LDR            *0(COUNT);        %FETCH STARTING DATA.
          RTL            0(LOOP);          %ROUTE.
          LDA            $R;               %SAVE STARTING DATA.
          RTL            -&NUM;            %FAKE END AROUND ROUTE (RIGHT).
          LDEE1          $MODE;            %END AROUND BY &NUM PES.
          LDA            $R;               %FILL IN STARTING DATA.
          SETE           E.OR.-E;          %ENABLE ALL.
          SETE1          E.OR.-E;          %
          LDR            *0(COUNT);        %FETCH DESTINATION DATA.
          STA            *0(COUNT);        %STORE STARTING DATA IN DEST.
          RTL            0(INDEX);         %ROUTE DESTINATION DATA BACK.
          LDA            $R;               %SAVE DESTINATION DATA.
          RTL            &NUM;             %FAKE END AROUND ROUTE (LEFT).
          CROTR(MODE)    &NUM(INDEX);      %END AROUND BY &NUM PES.
          LDEE1          $MODE;            %
          LDA            $R;               %FILL IN DESTINATION DATA.
          SETE           E.OR.-E;          %ENABLE ALL.
          SETE1          E.OR.-E;          %
          STA            *0(COUNT);        %STORE DEST. DATA IN START LOC.
          ALIT(INDEX)    #-1;              %INCREMENT INDEX (#-LOOP COUNT).
          CROTL(MODE)    &NUM(INDEX);      %RESTORE MODE PATTERN (ALMOST).
          COR(MODE)      $PATTERN;         %MODIFY MODE PATTERN.
          TXLTM(LOOP)    .OLOOP;           %RETURN TO LOCAL OLOOP(INC LOOP)
          RTL            $S,1;             %DESTINATION INDEX.
          LDS            $R;               %COMPLETE S TO S TRANSFER.
          LDR            *0(COUNT);        %FETCH START AND DEST. DATA.
          RTL            &NUM/2;           %EXCHANGE START AND DEST. DATA.
          LDA            $R;               %SAVE.
          RTL            -&NUM;            %FAKE END AROUND ROUTE OF &NUM/2
          LDEE1          $MODE;            %END AROUND BY &NUM PES.
          LDA            $R;               %FILL IN DESTINATION DATA.
          SETE           E.OR.-E;          %ENABLE ALL.
          SETE1          E.OR.-E;          %
          STA            *0(COUNT);        %STORE EXCHANGED START AND DEST.
          END             ;##;
%
%
%         FOR THE SMALLER DATA SETS (64 AND SMALLER) SCRAMBLING CAN BE
%    ACCOMPLISHED WITH ONLY ONE ACCESS TO ONE ROW OF STORAGE.  THE
%    INTERCHANGE OPERATION CAN BE DONE USING ONLY REGISTER TRANSFERS
%    AND ROUTING.  EXCHANGE IS A DEFINE WHICH PERFORMS ONE INTERCHANGE
%    ON DATA FOUND IN REGISTER A.  THE DISTANCE SEPARATING THE TWO WORDS
%    TO BE INTERCHANGED IS THE ONLY PARAMETER.  A PROPER MODE PATTERN
%    MUST BE SUPPLIED IN THE CU MODE REGISTER.
%
   DEFINE EXCHANGE       &NUM#=
          LDR            $A;               %DATA TO REGISTER R.
          RTL            &NUM;             %ROUTE STARTING DATA.
          LDEE1          $MODE;            %SELECT ELEMENTS TO BE EXCHANGED
          LDS            $A;               %PERFORM THE
          LDA            $R;               %EXCHANGE.
          RTL            $S,-&NUM;         %ROUTE BACK TO THE START.
          CSHL(MODE)     &NUM;             %POSITION MODE PATTERN TO
          LDEE1          $MODE;            %FILL IN DESTINATION DATA.
          LDA            $R;##;
%
HERE[ENTRY]::
          CTSBT(SIZE)    60,X8;
          CTSBT(SIZE)    59,X16;
          CTSBT(SIZE)    58,X32;
          CTSBT(SIZE)    57,X64;
```

```
            CTSBT(SIZE)     56,X128;
            CTSBT(SIZE)     55,X256;
            CTSBT(SIZE)     54,X512;
            CTSBT(SIZE)     53,X1024;
            CTSBT(SIZE)     52,X2048;
            CTSBT(SIZE)     51,A4096;
            CLC(2)          ;                   %ERROR. DATA SIZE TOO LARGE.
            COMPC(2)        ;                   %SET ACAR2 TO ALL ONES.
            LDL(0)          $03;                %RESTORE ACAR0.
            LDL(1)          $04;                %RESTORE ACAR1.
            LDL(3)          $02;                %FETCH RETURN ADDRESS.
            EXCHL(3)        $ICR;               %RETURN.
X8:         JUMP            A8;
X16:        JUMP            A16;
X32:        JUMP            A32;
X64:        JUMP            A64;
X128:       JUMP            A128;
X256:       JUMP            A256;
X512:       JUMP            A512;
X1024:      JUMP            A1024;
X2048:      JUMP            A2048;
%
%
%
%           THIS PORTION OF THE PROGRAM REORDERS (SCRAMBLES) THE DATA IN
%      SUCH A WAY THAT THE RESULTS OF THE TRANSFORM ROUTINE WILL COME OUT
%      SEQUENTIALLY STORED IN PE MEMORY.  ONLY ONE OF THE FOLLOWING TEN
%      SECTIONS OF CODE WILL BE EXECUTED DEPENDING ON THE SIZE OF THE DATA
%      SETS.  THE SCRAMBLING PROCESS IS PURELY A DATA TRANSFER PROBLEM
%      AND IS ACCOMPLISHED QUICKLY(A FRACTION OF THE TRANSFORM TIME).  THE
%      PRIMARY OPERATIONS ARE MEMORY ACCESS AND ROUTING.  THE OTHER IN-
%      STRUCTION SET UP INDEXING AND LOOP CONTROL.
%
A4096::     FETCHINDEX      57:6;               %DEFINE (FETCH FROM PEINDEX).
            LIT(LOOP)       1,31,1;             %LOOP COUNTER.
            LIT(INDEX)      =63;                %ROUTING INDEX (-LOOP COUNT).
LOOP1:      RTL             $S,1;               %DESTINATION INDEX.
            LDS             $R;                 %COMPLETE S TO S TRANSFER.
            RTL             $X,-1;              %STARTING INDEX.
            LDX             $R;                 %COMPLETE X TO X TRANSFER.
            LDR             *0(COUNT);          %FETCH STARTING DATA.
            RTL             0(LOOP);            %ROUTE.  END AROUND BY 64
            LDA             *0(COUNT);          %FETCH DESTINATION DATA.
            STR             *0(COUNT);          %STORE STARTING DATA IN DEST.
            RTL             $A,0(INDEX);        %ROUTE DESTINATION DATA BACK.
            STR             *0(COUNT);          %STORE DEST. DATA IN START LOC.
            ALIT(INDEX)     =-1;                %INCREMENT INDEX (=-LOOP COUNT).
            TXLTM(LOOP)     ,LOOP1;             %RETURN TO LOOP1(INC LOOP COUNT)
            RTL             $S,1;               %DESTINATION INDEX.
            LDS             $R;                 %COMPLETE S TO S TRANSFER.
            LDR             *0(COUNT);          %FETCH START AND DEST. DATA.
            RTL             32;                 %ROUTE END AROUND BY 64.
            STR             *0(COUNT);          %STORE EXCHANGED START AND DEST.
            ALIT(COUNT)     =63;                %INCREMENT COUNT.
            TXLTM(COUNT)    ,A4096;             %GO BACK FOR NEXT DATA SET.
            JUMP            FFT1;               %JUMP TO TRANSFORM ROUTINE.
A2048::     FETCHINDEX      51:6;               %DEFINE (FETCH FROM PEINDEX).
            LIT(MODE)       =10000000000200000000000:8;
            INTERCHANGE     32;                 %DEFINE (LARGE).
            ALIT(COUNT)     =31;                %INCREMENT COUNT.
            TXLTM(COUNT)    ,A2048;             %GO BACK FOR NEXT DATA SET.
```

```
            JUMP           FFT1;               %JUMP TO TRANSFORM ROUTINE.
A1024::     FETCHINDEX     45:6;               %DEFINE (FETCH FROM PEINDEX).
            LIT(MODE)      =1000004000020000100000:8;
            INTERCHANGE    16;                 %DEFINE (LARGE).
            LIT(LOOP)      1,15,0;             %LOOP COUNTER.
LOOP2:      LDA            0(COUNT);           %FETCH DATA.
            LIT(MODE)      =37777600000:8;
            EXCHANGE       16;                 %DEFINE.
            SETE           E.OR.-E;            %ENABLE ALL.
            SETE1          E.OR.-E;            %
            STA            0(COUNT);           %STORE RESULT.
            ALIT(COUNT)    =1;                 %INCREMENT COUNT.
            TXLTM(LOOP)    ,LOOP2;             %GO BACK FOR NEXT ROW.
            TXLT(COUNT)    $COUNT,A1024;       %GO BACK FOR NEXT DATA SET.
            JUMP           FFT1;               %JUMP TO TRANSFORM ROUTINE.
A512::      FETCHINDEX     39:6;               %DEFINE (FETCH FROM PEINDEX).
            LIT(MODE)      =1002004010020040100200:8;
            INTERCHANGE    8;                  %DEFINE (LARGE).
            LIT(LOOP)      1,7,0;              %LOOP COUNTER.
LOOP3:      LDA            0(COUNT);           %FETCH DATA.
            LIT(MODE)      =37700177400:8;
            EXCHANGE       24;                 %DEFINE.
            SETE           E.OR.-E;            %ENABLE ALL.
            SETE1          E.OR.-E;            %
            STA            0(COUNT);           %STORE RESULT.
            ALIT(COUNT)    =1;                 %INCREMENT COUNT.
            TXLTM(LOOP)    ,LOOP3;             %GO BACK FOR NEXT ROW.
            TXLT(COUNT)    $COUNT,A512;        %GO BACK FOR NEXT DATA SET.
            JUMP           FFT1;               %JUMP TO TRANSFORM ROUTINE.
A256::      FETCHINDEX     33:6;               %DEFINE (FETCH FROM PEINDEX).
            LIT(MODE)      =1042104210421042104210:8;
            INTERCHANGE    4;                  %DEFINE (LARGE).
            LIT(LOOP)      1,3,0;              %LOOP COUNTER.
LOOP4:      LDA            0(COUNT);           %FETCH DATA.
            LIT(MODE)      =36074170360:8;
            EXCHANGE       28;                 %DEFINE.
            LIT(MODE)      =7760000000177400:8;
            EXCHANGE       8;                  %DEFINE.
            SETE           E.OR.-E;            %ENABLE ALL.
            SETE1          E.OR.-E;            %
            STA            0(COUNT);           %STORE RESULT.
            ALIT(COUNT)    =1;                 %INCREMENT COUNT.
            TXLTM(LOOP)    ,LOOP4;             %GO BACK FOR NEXT ROW.
            TXLT(COUNT)    $COUNT,A256;        %GO BACK FOR NEXT DATA SET.
            JUMP           FFT1;               %JUMP TO TRANSFORM ROUTINE.
A128::      FETCHINDEX     27:6;               %DEFINE (FETCH FROM PEINDEX).
            RTL            $S,1;               %ROUTE INDEX.
            LDS            $R;                 %COMPLETE S TO S TRANSFER.
            LDA            $0(COUNT);          %FETCH DATA.
            LIT(MODE)      =5252525252525252525252:8;
            EXCHANGE       1;                  %DEFINE.
            SETE           E.OR.-E;            %ENABLE ALL.
            SETE1          E.OR.-E;            %
            STA            $0(COUNT);          %STORE RESULT.
            LIT(LOOP)      1,1,0;              %LOOP COUNTER.
LOOP5:      LDA            0(COUNT);           %FETCH DATA.
            LIT(MODE)      =31463146314:8;
            EXCHANGE       30;                 %DEFINE.
            LIT(MODE)      =74177417001703770360:8;
            EXCHANGE       12;                 %DEFINE.
            SETE           E.OR.-E;            %ENABLE ALL.
```

```
            SETE1               E.OR.-E;            %
            STA                 0(COUNT);           %STORE RESULT.
            ALIT(COUNT)         =1;                 %INCREMENT COUNT.
            TXLTM(LOOP)         .LOOP5;             %GO BACK FOR OTHER ROW.
            TXLT(COUNT)         $COUNT,A128;        %GO BACK FOR NEXT DATA SET.
            JUMP                FFT1;               %JUMP TO TRANSFORM ROUTINE.
A64::       LDA                 0(COUNT);           %FETCH DATA.
            LIT(MODE)           =25252525252:8;
            EXCHANGE            31;                 %DEFINE.
            LIT(MODE)           =63146000001463141:8;
            EXCHANGE            14;                 %DEFINE.
            LIT(MODE)           =36000170000740003601:8;
            EXCHANGE            4;                  %DEFINE.
            SETE                E.OR.-E;            %ENABLE ALL.
            SETE1               E.OR.-E;            %
            STA                 0(COUNT);           %STORE RESULT.
            TXLTM(COUNT)        .A64;               %GO BACK FOR NEXT DATA SET.
            JUMP                FFT1;               %JUMP TO TRANSFORM ROUTINE.
A32::       LDA                 0(COUNT);           %FETCH DATA.
            LIT(MODE)           =5252500000125252:8;
            EXCHANGE            15;                 %DEFINE.
            LIT(MODE)           =31400146000630003141:8;
            EXCHANGE            6;                  %DEFINE.
            SETE                E.OR.-E;            %ENABLE ALL.
            SETE1               E.OR.-E;            %
            STA                 0(COUNT);           %STORE RESULT.
            TXLTM(COUNT)        .A32;               %GO BACK FOR NEXT 2 DATA SETS.
            JUMP                FFT1;               %JUMP TO TRANSFORM ROUTINE.
A16::       LDA                 0(COUNT);           %FETCH DATA.
            LIT(MODE)           =25200125000524002521:8;
            EXCHANGE            7;                  %DEFINE.
            LIT(MODE)           =6014030060140300601:8;
            EXCHANGE            2;                  %DEFINE.
            SETE                E.OR.-E;            %ENABLE ALL.
            SETE1               E.OR.-E;            %
            STA                 0(COUNT);           %STORE RESULT.
            TXLTM(COUNT)        .A16;               %GO BACK FOR NEXT 4 DATA SETS.
            JUMP                FFT1;               %JUMP TO TRANSFORM ROUTINE.
A8::        LDA                 0(COUNT);           %FETCH DATA.
            LIT(MODE)           =50120240501202405012:8;
            EXCHANGE            3;                  %DEFINE.
            SETE                E.OR.-E;            %ENABLE ALL.
            SETE1               E.OR.-E;            %
            STA                 0(COUNT);           %STORE RESULT.
            TXLTM(COUNT)        .A8;                %GO BACK FOR NEXT 8 DATA SETS.
%
%
%
%
%
%
%
%
%
%
%
            END                 HERE.
```

```
LISTED ON: 06/13/71  AT: 09:13

         BEGIN
%  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %
% %                     TRANSFORM  SECTION                  %  %
%  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %  %
         EXTERNAL          SIN;
SIZE:    EQU               0;
         REGC              $SIZE=$C0;
MODE:    EQU               1;
         REGC              $MODE=$C1;
LOOP:    EQU               2;
         REGC              $LOOP=$C2;
COUNT:   EQU               3;
         REGC              $COUNT=$C3;
%        THE FOLLOWING DEFINE IS A STANDARD TRANSFER OPERATION USED IN
%    THE FOURIER TRANSFORM ROUTINE.  THE PARTIAL RESULTS OF EACH ITERA-
%    TION ARE USED IN THE PE WHERE THEY WERE COMPUTED AS WELL AS IN A
%    PE SOME DISTANCE AWAY FOR THE NEXT ITERATION.  THE FUNCTION OF THIS
%    DEFINE IS TO MOVE THESE PARTIAL RESULTS(FOUND IN REGISTER A) TO THE
%    PROPER PES (&DIST IS A NUMBER TELLING HOW FAR).  THE PROPER MODE
%    PATTERN MUST BE SUPPLIED IN THE CU REGISTER,MODE(ACAR1).  NOTE THAT
%    REGISTER MODE (ACAR1) IS NUT CHANGED BY THIS DEFINE.
%
%
  DEFINE   TRANSFER          &DIST;=
         RTL               $A,&DIST;          %ROUTE REGISTER A RIGHT.
         COMPC(MODE)        ;                 %COMPLIMENT MODE PATTERN.
         LDEE1             $MODE;             %ENABLE PES.
         LDS               $R;                %LOAD REGISTER S IN DESIRED PES.
         RTL               $A,-&DIST;         %ROUTE REGISTER A LEFT.
         COMPC(MODE)        ;                 %COMPLIMENT MODE PATTERN.
         LDEE1             $MODE;             %ENABLE PES.
         LDS               $A;                %LOAD REGISTER S IN DESIRED PES.
         LDA               $R;                %LOAD REGISTER A IN SAME PES.
         SETE              E.OR.-E;           %ENABLE ALL PES.
         SETE1             E.OR.-E;##;
$ PAGE    ;
%        THE FOLLOWING DEFINE PERFOMS THE COMPLEX MULTIPLICATION
%    OPERATION ON TWO OPERANDS.  EACH OPERAND IS ON WORD COMPOSED OF
%    TWO 32-BIT FLOATING POINT NUMBERS, THE OUTER OF WHICH IS THE REAL
%    PART AND THE INNER IS THE IMAGINARY PART.  THE TWO OPERANDS MUST
%    BE IN THE A AND S REGISTERS IN ALL PES.  THE RESULT IS LEFT IN
%    REGISTER A.  THE RESULTS IN TERMS OR R1(REAL PART OF FIRST OPERAND)
%    R2,I1,AND I2 ARE AS FOLLOWS.
%
%        R=R1*R2-I1*I2
%        I=R1*I2+R2*I1
%
  DEFINE   COMPLEXMULT       &WHO;=
         MLRN              $S;                %MULTIPLY REALS AND IMAGINARIES.
         LDR               &WHO;              %RELOAD SECOND OPERAND.
         SETE              E.AND.-E;          %
         CHSA               ;                 %CHANGE SIGN OF INNER (I**2=-1).
         SETE              E.OR.-E;           %
         LDB               $S;                %ONE OPERAND TO B REGISTER.
         LDS               $A;                %SAVE PARTIAL RESULT.
         LDA               $B;                %FIRST OPERAND TO A REGISTER.
         SWAPA              ;                 %IMAG TO OUTER, REAL TO INNER.
         MLRN              $R;                %MULTIPLY R1*I2 AND R2*I1.
         LDB               $S;                %POSITION PARTIAL RESULTS
```

```
        SWAPX              ;                    %FOR FINAL ADDITION.
        ADN                $B;**;
%       THE FOLLOWING DEFINE REQUIRES PROPER CONSTANTS TO BE FOUND
%   IN REGISTER S AND PRODUCES TWO RESULTS AND STORES THEM IN APPROPRI-
%   ATE MEMORY LOCATIONS.  THE OFFSET USED TO FIND THE FIRST OPERAND
%   SHOULD BE EQUAL TO HALF THE NUMBER OF ROWS BEING OPERATED ON IN A
%   GIVEN ITERATION.  THE SIGNIFICANCE OF THIS DEFINE FOLLOW IN BRIEF.
%       THE BASIC OPERATION OF THE FAST FOURIER TRANSFORM IS A COMPLEX
%   MULTIPLICATION BETWEEN THE RESULTS OF ONE ITERATION AND A SET OF
%   CONSTANTS (W S).  THIS NEW QUANTITY IS THEN ADDED TO THE RESULTS OF
%   THE PREVIOUS ITERATION TO PRODUCE RESULTS FOR THE PRESENT ITERA-
%   TION.  THE SYMETRY OF THE CALCULATIONS AS WELL AS THAT OF THE W S
%   ALLOWS HALF THE OPERATIONS TO BE ELIMINATED BY ADDING ONE SUBTRAC-
%   TION TO THE OTHER HALF.  THUS A STANDARD DEFINE WHICH ACCOMPLISHES
%   ON COMPLEX MULTIPLY ONE ADD AND ONE SUBTRACT AND PRODUCES TWO RES.
%
DEFINE  MULTADDSUB         &OFFSET;=
        LDA                &OFFSET(COUNT);%FETCH FIRST OPERAND.
        COMPLEXMULT        &OFFSET(COUNT);%MULTIPLY BY CONSTANT(W) IN RGS.
        LDR                0(COUNT);            %FETCH SECOND OPERAND.
        LDS                $A;                  %SAVE PARTIAL RESULT.
        ADN                $R;                  %ADD 2ND OPERAND TO PARTIAL RES.
        STA                0(COUNT);            %STORE ONE RESULT.
        LDA                $R;                  %2ND OPERAND TO REGISTER A.
        SBN                $S;                  %SUBTRACT PARTIAL RESULT.
        STA                &OFFSET(COUNT);**;  %STORE THE OTHER RESULT.
%       THE CONSTANTS (W S) FOR THIS PROGRAM ARE REPRESENTED AS SINES
%   AND COSINES RATHER THAN EXPONENTIALS.  TO DECREASE THE NUMBER OF
%   CONSTANTS REQUIRED TO BE STORED AS PART OF THE PROGRAM ADVANTAGE
%   WAS TAKEN OF THEIR REGULARITY AND SYMETRY. THE SINES OR COSINES
%   OF ANGLES IN ANY OF THE FOUR QUADRANTS ARE SIMPLE TRANSFORMATION
%   OF THOSE OF ANGLES IN THE FIRST QUADRANT.  FOR THIS REASON ONLY
%   FIRST QUADRANT SINES AND COSINES ARE STORED.
%       THE FOLLOWING DEFINE LOADS A SET OF CONSTANTS AND TRANSFORMS
%   THEM INTO THE PROPER SECOND QUADRANT VALUES.  THE RESULTS OF THIS
%   DEFINE ARE LEFT IN REGISTER S AND THE MEMORY LOCATIONS OF THE
%   STORED CONSTANTS ARE NOT DISTURBED.  THE INPUT PARAMETER IS ASSUMED
%   TO BE THE ADDRESS OF THE MEMORY LOCATION CONTAINING THE CONSTANTS
%   REQUIRED IN THE CURRENT ITERATION.  THE TRANSFORMATION IS DEFINED:
%
%       REAL(QUAD 2)=-IMAGINARY(QUAD 1)
%       IMAGINARY(QUAD 2)=REAL(QUAD 1)
%
DEFINE  CHANGEQUAD         &COMPLEXNUM;=
        LDA                &COMPLEXNUM;         %FETCH CONSTANTS.
        SWAPA               ;                   %SWAP REAL AND IMAGINARY PARTS.
        SETE1              E.AND.-E;            %
        CHSA                ;                   %CHANGE SIGN OF REAL PART.
        SETE1              E.OR.-E;             %
        LDS                $A;**;               %LEAVE RESULT IN REGISTER S.
%
%       THE FOLLOWING DEFINE RESETS THE TRUE-FALSE FLIP-FLOP (TFFF)
%   WHICH IS USED TO CONTROL A SMALL LOOP.
%
DEFINE  RESETFF             =
        CACRB              0 **;                %RESET ACR REGISTER BIT 0 (TFFF)
%
%       THE FOLLOWING DEFINE SETS THE TRUE-FALSE FLIP-FLOP WHICH IS
%   USED TO CONTROL A SMALL LOOP.
%
DEFINE  SETFF               =
```

```
        CACRB          200:8 **;       %SET ACR REGISTER BIT 0 (TFFF).
%       AT THIS POINT SCRAMBLING HAS BEEN COMPLETED AND THE REAL WORK
%     MUST BEGIN.  THE TRANSFORM ALGORITHM BEGINS BY WORKING WITH GROUPS
%     OF EIGHT DATA POINTS CONTAINED IN EIGHT CONSECUTIVE PES.  IN EACH
%     SUCCEEDING STEP TWICE THE PREVIOUS NUMBER OF DATA POINTS ARE USED
%     IN EACH ITERATION.  THIS PROCESS CONTINUES UNTIL THE ENTIRE DATA
%     SET IS USED FOR THE FINAL STEP.  THE BEGINING OF EACH SECTION OF
%     CODE IS LABELED BY THE NUMBER OF DATA POINTS IN A GROUP USED IN
%     THAT SECTION (ONE STEP OF THE TRANSFORM).  WHEN THE STEP HAS BEEN
%     COMPLETED WHICH CORRESPONDS TO THE SIZE OF THE DATA SET THE TRANS-
%     FORM IS DONE AND THE REMAIN CODE IS SKIPPED.  IT IS IMPORTANT THAT
%     EACH STEP IS COMPLETELY INDEPENT OF THE SIZE OR THE NUMBER OF THE
%     DATA SET(S) BEING TRANSFORMED.
%
        FILL           16;             %ALIGN TO 8 PE BOUNDRY.
FFT1[ENTRY]::
        SETE           E.OR.-E;        %ENABLE ALL PES.
        SETE1          E.OR.-E;        %
        CHWS           32;             %CHANGE TO 32-BIT MODE.
        CACRB          211:8;          %TURN OFF UNDERFLOW.
        LDL(SIZE)      $D1;            %FETCH SIZE INFO.
        SLIT(COUNT)    =0;             %RESET COUNT.
        CADD(COUNT)    $SIZE;          %SET TO STARTING POINTER.
F8:     LDA            0(COUNT);       %LOAD DATA.
        LIT(MODE)      =1252525252525252525252:8;
        RTL            $A,1;           %ROUTE REGISTER A RIGHT.
        COMPC(MODE)     ;              %COMPLIMENT MODE PATTERN.
        LDEE1          $MODE;          %ENABLE PES.
        LDS            $R;             %LOAD REGISTER S IN DESIRED PES.
        RTL            $A,-1;          %ROUTE REGISTER A LEFT.
        CHSA            ;              %CHANGE SIGN OF A IN DESIRED PES
        COMPC(MODE)     ;              %COMPLIMENT MODE PATTERN.
        LDEE1          $MODE;          %ENABLE PES.
        LDS            $A;             %LOAD REGISTER S IN DESIRED PES.
        LDA            $R;             %LOAD REGISTER A IN SAME PES.
        SETE           E.OR.-E;        %ENABLE ALL PES.
        SETE1          E.OR.-E;        %
        ADN            $S;             %PARTIAL RESULT.
        LIT(MODE)      =1463146314631463146314:8;
        RTL            $A,2;           %ROUTE REGISTER A RIGHT.
        COMPC(MODE)     ;              %COMPLIMENT MODE PATTERN.
        LDEE1          $MODE;          %ENABLE PES.
        LDS            $R;             %LOAD REGISTER S IN DESIRED PES.
        RTL            $A,-2;          %ROUTE REGISTER A LEFT.
        CHSA            ;              %CHANGE SIGN OF A IN DESIRED PES
        COMPC(MODE)     ;              %COMPLIMENT MODE PATTERN.
        LDEE1          $MODE;          %ENABLE PES.
        LDS            $A;             %LOAD REGISTER S IN DESIRED PES.
        LDA            $R;             %LOAD REGISTER A IN SAME PES.
        LIT(MODE)      =525252525252525252525:8;
        LDEE1          $MODE;          %
        SWAPA           ;              %     FAKE
        SETE1          E.AND.-E;       %     SINES
        CHSA            ;              %     AND
        SETE           E.OR.-E;        %     COSINES.
        SETE1          E.OR.-E;        %
        ADN            $S;             %PARTIAL RESULT.
        LIT(MODE)      =1703607417036074170360:8;
        TRANSFER       4;              %ROUTING DEFINE.
        MLRN           SIN;            %MULTIPLY BY CONSTANT.
        LIT(MODE)      =525252525252525252525:8;
```

```
          LDEE1             $MODE;            %
          LDR               $A;               %     FAKE
          SWAPA              ;                %     COMPLEX
          SETE1             E.AND.-E;         %     MULTIPLY
          CHSA               ;                %     BY
          SETE1             E.OR. E;          %     TAKING
          ADN               $R;               %     ADVANTAGE
          LIT(MODE)         =210421042104210421042:8;
          LDEE1             $MODE;            %     OF
          SWAPA              ;                %     THE
          CSHR(MODE)        1;                %     SYMETRY
          LDE1              $MODE;            %     OF
          SETE              E.AND.-E;         %     SINES
          CHSA               ;                %     AND
          SETE              E.OR.-E;          %     COSINES.
          SETE1             E.OR.-E;          %
          ADN               $S;               %PARTIAL RESULT.
          STA               0(COUNT);         %END OF ITERATION.
          TXLTM(COUNT)      .F8;              %ITERATE ONCE FOR EACH ROW.
          CTSBF(SIZE)       36,NE8;           %GO ON IF NOT DONE.
          LIT(2)            PAIR(4.0,4.0);    %SET UP EXPONENT FOR INVERSE.
          JUMP              OUT;              %QUIT IF SIZE=8.
NE8:      SLIT(COUNT)       =0;               %ZERO COUNT.
          CADD(COUNT)       $SIZE;            %START WITH FIRST ROW OF DATA.
          LIT(MODE)         =1774007760037700177400:8;
F16:      LDA               0(COUNT);         %RESULTS FROM PREVIOUS STEP.
          TRANSFER          8;                %ROUTING DEFINE.
          STS               0(COUNT);         %SAVE UTHER OPERAND.
          LDS               $A;               %SAVE FIRSTOAPERAND.
          LDA               SIN+1;            %FETCH CONSTANTS (W S).
          COMPLEXMULT       SIN+1;            %DEFINE (RGS*SIN).
          ADN               0(COUNT);         %ADD OTHER OPERAND.
          STA               0(COUNT);         %END OF ITERATION.
          TXLTM(COUNT)      .F16;             %ITERATE ONCE PER ROW.
          CTSBF(SIZE)       35,NE16;          %GO ON IF NOT DONE.
          LIT(2)            PAIR(8.0,8.0);    %SET UP EXPONENT FOR INVERSE.
          JUMP              OUT;              %QUIT IF SIZE=16.
NE16:     SLIT(COUNT)       =0;               %ZERO COUNT.
          CADD(COUNT)       $SIZE;            %START WITH FIRST ROW OF DATA.
          LIT(MODE)         =177777000037777600000:8;
F32:      LDA               0(COUNT);         %RESULTS FROM PREVIOUS STEP.
          TRANSFER          16;               %ROUTING DEFINE
          STS               0(COUNT);         %SAVE UTHER OPERAND.
          LDS               $A;               %SAVE FIRSTOAPERAND.
          LDA               SIN+2;            %FETCH CONSTANTS (W S).
          COMPLEXMULT       SIN+2;            %DEFINE (RGS*SIN).
          ADN               0(COUNT);         %ADD OTHER OPERAND.
          STA               0(COUNT);         %END OF ITERATION.
          TXLTM(COUNT)      .F32;             %ITERATE ONCE PER ROW.
          CTSBF(SIZE)       34,NE32;          %GO ON IF NOT DONE.
          LIT(2)            PAIR(16.0,16.0);%SET UP EXPONENT FOR INVERSE.
          JUMP              OUT;              %QUIT IF SIZE=32.
NE32:     SLIT(COUNT)       =0;               %ZERO COUNT.
          CADD(COUNT)       $SIZE;            %START WITH FIRST ROW OF DATA.
          LIT(MODE)         =1777777777740000000000:8;
F64:      LDA               0(COUNT);         %RESULTS FROM PREVIOUS STEP.
          RTL               $A,32;            %ROUTE END AROUND 32.
          LDS               $R;               %SAVE IN REGISTER S.
          LDEE1             $MODE;            %ENABLE PES 0 THRU 31.
          LDS               $A;               %INTERCHANGE REGISTER S
          LDA               $R;               %AND REGISTER A.
```

```
          SETE                E.OR.-E;            %ENABLE ALL PES.
          SETE1               E.OR.-E;            %
          STS                 O(COUNT);           %SAVE UTHER OPERAND.
          LDS                 $A;                 %SAVE FIRSTUAPERAND.
          LDA                 SIN+3;              %FETCH CONSTANTS (W S).
          COMPLEXMULT         SIN+3;              %DEFINE (RGS*SIN).
          ADN                 O(COUNT);           %ADD OTHER OPERAND.
          STA                 O(COUNT);           %END OF ITERATION.
          TXLTM(COUNT)        ,F64;               %ITERATE ONCE PER ROW.
          CTSBF(SIZE)         33,NE64;            %GO ON IF NOT DONE.
          LIT(2)              PAIR(32.0,32.0);%SET UP EXPONENT FOR INVERSE.
          JUMP                OUT;                %QUIT IF SIZE=64.
NE64:     SLIT(COUNT)         =0;                 %ZERO COUNT.
          CADD(COUNT)         $SIZE;              %START WITH FIRST ROW OF DATA.
F128:     LDS                 SIN+4;              %FETCH CONSTANTS (W S).
          MULTADDSUB          1;                  %DEFINE  (DATA+1)*SIN +AND- DATA
          ALIT(COUNT)         =1;                 %GO BY PAIRS OF ROWS.
          TXLTM(COUNT)        ,F128;              %ITERATE ONCE PEP PAIR UF ROWS.
          CTSBF(SIZE)         32,NE128;           %GO ON IF NOT DONE.
          LIT(2)              PAIR(64.0,64.0);%SET UP EXPONENT FOR INVERSE.
          JUMP                OUT;                %QUIT IF SIZE=128.
NE128:    SLIT(COUNT)         =0;                 %ZERO COUNT.
          CADD(COUNT)         $SIZE;              %START WITH FIRST ROW OF DATA.
P256:     RESETFF              ;                  %DEFINE (TFFF=0) LOOP CONTROL.
          LDS                 SIN+5;              %FETCH CONSTANTS (W S).
F256:     MULTADDSUB          2;                  %DEFINE (DATA+2)*SIN +AND- DATA.
          ALIT(COUNT)         =1;                 %INCREMENT COUNT.
          SKIPT               ,SKIP1;             %CHECK INNER LOOP.
          CHANGEQUAD          SIN+5;              %DEFINE-MODIFY CONSTANTS.
          SETFF                ;                  %DEFINE (TFFF=1) LEAVE NEXT TIME
          SKIP                ,F256;              %INNER LOOP.
SKIP1:    ALIT(COUNT)         =1;                 %INCREMENT COUNT.
          TXLTM(COUNT)        ,P256;              %ITERATE ONCE FOR EACH 4 ROWS.
          CTSBF(SIZE)         31,NE256;           %GO ON IF NOT DONE.
          LIT(2)              PAIR(128.0,128.0);%SET UP EXPONENT FOR INVERSE.
          JUMP                OUT;                %QUIT IF SIZE=256.
NE256:    SLIT(COUNT)         =0;                 %ZERO COUNT.
          CADD(COUNT)         $SIZE;              %START WITH FIRST ROW OF DATA.
          LIT(LOOP)           1,1,0;              %LOOP COUNTER.
P512:     RESETFF              ;                  %DEFINE (TFFF=0) LOOP CONTROL.
          LDS                 6+SIN(LOOP);        %FETCH CONSTANTS (W S).
F512:     MULTADDSUB          4;                  %DEFINE (DATA+4)*SIN +AND- DATA.
          ALIT(COUNT)         =2;                 %INCREMENT COUNT.
          SKIPT               ,SKIP2;             %CHECK INNER LOOP.
          CHANGEQUAD          6+SIN(LOOP);        %DEFINE-MODIFY CONSTANTS.
          SETFF                ;                  %DEFINE (TFFF=1) LEAVE NEXT TIME
          SKIP                ,F512;              %INNER LOOP.
SKIP2:    ALIT(COUNT)         =-3;                %INCREMENT COUNT.
          TXLTM(LOOP)         ,P512;              %GO BACK FOR NEXT PAIR OF ROWS.
          ALIT(LOOP)          -2;                 %RESET LOOP COUNTER.
          ALIT(COUNT)         =5;                 %SET COUNT FOR NEXT DATA SET.
          TXLTM(COUNT)        ,P512;              %GO BACK FOR NEXT DATA SET.
          CTSBF(SIZE)         30,NE512:           %GO ON IF NOT DONE.
          LIT(2)              PAIR(256.0,256.0);%SET UP EXPONENT FOR INVERSE.
          JUMP                OUT;                %QUIT IF SIZE=512.
NE512:    SLIT(COUNT)         =0;                 %ZERO COUNT.
          CADD(COUNT)         $SIZE;              %START WITH FIRST ROW OF DATA.
          LIT(LOOP)           1,3,0;              %LOOP COUNTER.
P1024:    RESETFF              ;                  %DEFINE (TFFF=0) LOOP CONTROL.
          LDS                 8+SIN(LOOP);        %FETCH CONSTANTS (W S).
F1024:    MULTADDSUB          8;                  %DEFINE (DATA+8)*SIN +AND- DATA.
```

```
           ALIT(COUNT)      =4;               %INCREMENT COUNT.
           SKIPT            ,SKIP3;           %CHECK INNER LOOP.
           CHANGEQUAD       8+SIN(LOOP);      %DEFINE-MODIFY CONSTANTS.
           SETFF             ;                %DEFINE (TFFF=1) LEAVE NEXT TIME
           SKIP             ,F1024;           %INNER LOOP.
SKIP3:     ALIT(COUNT)      =-7;              %INCREMENT COUNT.
           TXLTM(LOOP)      ,P1024;           %GO BACK FOR NEXT PAIR OF ROWS.
           ALIT(LOOP)       =-4;              %RESET LOOP COUNTER.
           ALIT(COUNT)      =11;              %SET COUNT FOR NEXT DATA SET.
           TXLTM(COUNT)     ,P1024;           %GO BACK FOR NEXT DATA SET.
           CTSBF(SIZE)      29,NE1024;        %GO ON IF NOT DONE.
           LIT(2)           PAIR(512.0,512.0);%SET UP EXPONENT FOR INVERSE.
           JUMP             OUT;              %QUIT IF DONE (SIZE=1024).
NE1024:    SLIT(COUNT)      =0;               %ZERO COUNT.
           CADD(COUNT)      $SIZE;            %START WITH FIRST ROW OF DATA.
           LIT(LOOP)        1,7,0;            %LOOP COUNTER.
P2048:     RESETFF           ;                %INNER LOOP CONTROL. (TFFF=0).
           LDS             12+SIN(LOOP);      %FETCH CONSTANTS (W S).
F2048:     MULTADDSUB       16;               %DEFINE (DATA+16)*SIN +AND- DATA
           ALIT(COUNT)      =8;               %INCREMENT COUNT.
           SKIPT            ,SKIP4;           %CHECK INNER LOOP.
           CHANGEQUAD       12+SIN(LOOP);     %DEFINE-MODIFY CONSTANTS.
           SETFF             ;                %DEFINE (TFFF=1) LEAVE NEXT TIME
           SKIP             ,F2048;           %INNER LOOP.
SKIP4:     ALIT(COUNT)      =-15;             %INCREMENT COUNT.
           TXLTM(LOOP)      ,P2048;           %GO BACK FOR NEXT PAIR OF ROWS.
           ALIT(LOOP)       =-8;              %RESET LOOP COUNTER.
           ALIT(COUNT)      =23;              %SET COUNT FOR NEXT DATA SET.
           TXLTM(COUNT)     ,P2048;           %GO BACK FOR NEXT DATA SET.
           CTSBF(SIZE)      28,NE2048;        %GO ON IF NOT DONE.
           LIT(2)           PAIR(1024.0,1024.0);     %SET UP EXPONENT.
           JUMP             OUT;              %QUIT IF DONE (SIZE=2048).
NE2048:    SLIT(COUNT)      =0;               %ZERO COUNT.
           CADD(COUNT)      $SIZE;            %START WITH FIRST ROW OF DATA.
           LIT(LOOP)        1,15,0;           %LOOP COUNTER.
P4096:     RESETFF           ;                %DEFINE (TFFF=0) LOOP CONTROL.
           LDS             20+SIN(LOOP);      %FETCH CONSTANTS (W S).
F4096:     MULTADDSUB       32;               %DEFINE (DATA+32)*SIN +AND- DATA
           ALIT(3)          =16;              %INCREMENT COUNT.
           SKIPT            ,SKIP5;           %CHECK INNER LOOP.
           CHANGEQUAD       20+SIN(LOOP);     %DEFINE-MODIFY CONSTANTS.
           SETFF             ;                %DEFINE (TFFF=1) LEAVE NEXT TIME
           SKIP             ,F4096;           %INNER LOOP.
SKIP5:     ALIT(COUNT)      =-31;             %INCREMENT COUNT.
           TXLTM(LOOP)      ,P4096;           %GO BACK FOR NEXT PAIR OF ROWS.
           ALIT(LOOP)       =-16;             %RESET LOOP COUNTER.
           ALIT(COUNT)      =47;              %SET COUNT FOR NEXT DATA SET.
           TXLTM(COUNT)     ,P4096;           %GO BACK FOR NEXT DATA SET.
           LIT(2)           PAIR(2048.0,2048.0);     %SET UP EXPONENT.
OUT::      CTSBF(SIZE)      15,RETURN;        %LEAVE IF NOT INVERSE FFT.
%
%          THE FOLLOWING CODE IS EXECUTED IF AN INVERSE TRANSFORM IS
%     REQUESTED.  THE ONLY OPERATION REQUIRED IS THAT EACH ELEMENT OF
%     THE SOLUTION MUST BE DIVIDED BY N(SIZE). THE NUMBER OF DATA POINTS
%     IN THE FUNCTION.  SINCE N IS A POWER OF TWO THIS CAN BE DONE BY
%     SUBTRACTING LOG(BASE 2)N FROM EACH EXPONENT.  THE PROPER EXPONENTS
%     ARE CREATED AT A TIME WHEN N(SIZE) IS KNOW EXACTLY AND THEY ARE
%     PUT IN ACAR2(LOOP) JUST BEFORE EXECUTION IS TRANSFERED TO OUT.
%
           SLIT(COUNT)      =0;               %ZERO COUNT.
           CADD(COUNT)      SIZE;             %START WITH FIRST ROW OF DATA.
```

```
INVERSE:  LDA              O(COUNT);      %FETCH DATA.
          SBEX             $C2;           %DIVIDE BY SIZE (NO UNDERFLOW).
          TXLTM(COUNT)     ,INVERSE;      %DO FOR ALL ROWS OF DATA.
RETURN:   LDL(3)           $D2;           %FETCH RETURN ADDRESS.
          LDL(0)           $D3;           %RESTORE ACAR0.
          LDL(1)           $D4;           %RESTORE ACAR1.
          EXCHL(3)         $ICR;          %RETURN TO CALLING PROGRAM.
          END              FFT1.
```

```
$ RESET LIST;
BEGIN
   FILL      128;
SIN[ENTRY];;DATA (PAIR(1,0,1,0),PAIR(.7071067812,.7071067812),
     PAIR(1.0,-1.0),PAIR(-.7071067812,.7071067812),
      PAIR(-1,0,-1,0),PAIR(-.7071067812,-.7071067812),
      PAIR(-1,0,1,0),PAIR(.7071067812,-.7071067812))8,
    (PAIR(1,0,0,0),PAIR(.9238795325,.3826834324),
    PAIR( 0.7071067812, 0.7071067812),PAIR( 0.3826834324, 0.9238795325),
    PAIR(0,0,1.0),PAIR(-.3826834324,.9238795325),
    PAIR(-0.7071067812, 0.7071067812),PAIR(-0.9238795325, 0.3826834324),
    PAIR(-1,0,0,0),PAIR(-.9238795325,-.3826834324),
 PAIR(-.7071067812,-.7071067812),PAIR(-.3826834324,-.9238795325),
    PAIR(0,0,-1,0),PAIR(.3826834324,-.923875325),
 PAIR( 0.7071067812,-0.7071067812),PAIR( 0.9238795325,-0.3826834324))4,
    (PAIR(1,0,0,0),PAIR(.9807852804,.195090322),
    PAIR( 0.9238795325, 0.3826834324),PAIR( 0.8314696123, 0.5555702330),
    PAIR( 0.7071067812, 0.7071067812),PAIR( 0.5555702330, 0.8314696123),
    PAIR( 0.3826834324, 0.9238795325),PAIR( 0.1950903220, 0.9807852804),
     PAIR(0.0,1.0),PAIR(-.195090322,.9807852804),
    PAIR(-0.3826834324, 0.9238795325),PAIR(-0.5555702330, 0.8314696123),
    PAIR(-0.7071067812, 0.7071067812),PAIR(-0.8314696123, 0.5555702330),
    PAIR(-0.9238795325, 0.3826834324),PAIR(-0.9807852804, 0.1950903220),
     PAIR(-1,0,0.0),PAIR(-.9807852804,-.195090322),
    PAIR(-0.9238795325,-0.3826834324),PAIR(-0.8314696123,-0.5555702330),
    PAIR(-0.7071067812,-0.7071067812),PAIR(-0.5555702330,-0.8314696123),
    PAIR(-0.3826834324,-0.9238795325),PAIR(-0.1950903220,-0.9807852804),
     PAIR(0.0,-1.0),PAIR(.195090322,-.9807852804),
    PAIR( 0.3826834324,-0.9238795325),PAIR( 0.5555702330,-0.8314696123),
    PAIR( 0.7071067812,-0.7071067812),PAIR( 0.8314696123,-0.5555702330),
 PAIR( 0.9238795325,-0.3826834324),PAIR( 0.9807852804,-0.1950903220))2,
     PAIR(1.0,0.0),PAIR(.9951847267,.0980171403),
    PAIR( 0.9807852804, 0.1950903220),PAIR( 0.9569403357, 0.2902846773),
    PAIR( 0.9238795325, 0.3826834324),PAIR( 0.8819212644, 0.4713967368),
    PAIR( 0.8314696123, 0.5555702330),PAIR( 0.7730104534, 0.6343932842),
    PAIR( 0.7071067812, 0.7071067812),PAIR( 0.6343932842, 0.7730104534),
    PAIR( 0.5555702330, 0.8314696123),PAIR( 0.4713967368, 0.8819212644),
    PAIR( 0.3826834324, 0.9238795325),PAIR( 0.2902846773, 0.9569403357),
    PAIR( 0.1950903220, 0.9807852804),PAIR( 0.0980171403, 0.9951847267),
     PAIR(0,0,1.0),PAIR(-.0980171403,.9951847267),
    PAIR(-0.1950903220, 0.9807852804),PAIR(-0.2902846773, 0.9569403357),
    PAIR(-0.3826834324, 0.9238795325),PAIR(-0.4713967368, 0.8819212644),
    PAIR(-0.5555702330, 0.8314696123),PAIR(-0.6343932842, 0.7730104534),
    PAIR(-0.7071067812, 0.7071067812),PAIR(-0.7730104534, 0.6343932842),
    PAIR(-0.8314696123, 0.5555702330),PAIR(-0.8819212644, 0.4713967368),
    PAIR(-0.9238795325, 0.3826834324),PAIR(-0.9569403357, 0.2902846773),
    PAIR(-0.9807852804, 0.1950903220),PAIR(-0.9951847267, 0.0980171403),
     PAIR(-1.0,0.0),PAIR(-.9951847267,-.0980171404),
    PAIR(-0.9807852804,-0.1950903220),PAIR(-0.9569403357,-0.2902846773),
    PAIR(-0.9238795325,-0.3826834324),PAIR(-0.8819212644,-0.4713967368),
    PAIR(-0.8314696123,-0.5555702330),PAIR(-0.7730104534,-0.6343932842),
    PAIR(-0.7071067812,-0.7071067812),PAIR(-0.6343932842,-0.7730104534),
    PAIR(-0.5555702330,-0.8314696123),PAIR(-0.4713967368,-0.8819212644),
    PAIR(-0.3826834324,-0.9238795325),PAIR(-0.2902846773,-0.9569403357),
 PAIR(-.1950903220,-.9807852804),PAIR(-.0980171403,-.9951847267),
     PAIR(0.0,-1.0),PAIR(.0980171403,-.9951847267),
    PAIR( 0.1950903220,-0.9807852804),PAIR( 0.2902846773,-0.9569403357),
```

```
PAIR( 0.3826834324,-0.9238795325),PAIR( 0.4713967368,-0.8819212644),00005600
PAIR( 0.5555702330,-0.8314696123),PAIR( 0.6343932842,-0.7730104534),00005700
PAIR( 0.7071067812,-0.7071067812),PAIR( 0.7730104534,-0.6343932842),00005800
PAIR( 0.8314696123,-0.5555702330),PAIR( 0.8819212644,-0.4713967368),00005900
PAIR( 0.9238795325,-0.3826834324),PAIR( 0.9569403357,-0.2902846773),00006000
PAIR( 0.9807852804,-0.1950903220),PAIR( 0.9951847267,-0.0980171403),00006100
 PAIR(1.0,0.0),PAIR(.9987954562,.0490676743),                        00006200
PAIR( 0.9951847267, 0.0980171403),PAIR( 0.9891765100, 0.1467304745),00006300
PAIR( 0.9807852804, 0.1950903220),PAIR( 0.9700312532, 0.2429801799),00006400
PAIR( 0.9569403357, 0.2902846773),PAIR( 0.9415440652, 0.3368898534),00006500
PAIR( 0.9238795325, 0.3826834324),PAIR( 0.9039892931, 0.4275550934),00006600
PAIR( 0.8819212644, 0.4713967368),PAIR( 0.8577286100, 0.5141027442),00006700
PAIR( 0.8314696123, 0.5555702330),PAIR( 0.8032075315, 0.5956993045),00006800
PAIR( 0.7730104534, 0.6343932842),PAIR( 0.7409511254, 0.6715589548),00006900
PAIR( 0.7071067812, 0.7071067812),PAIR( 0.6715589548, 0.7409511254),00007000
PAIR( 0.6343932842, 0.7730104534),PAIR( 0.5956993045, 0.8032075315),00007100
PAIR( 0.5555702330, 0.8314696123),PAIR( 0.5141027442, 0.8577286100),00007200
PAIR( 0.4713967368, 0.8819212644),PAIR( 0.4275550934, 0.9039892931),00007300
PAIR( 0.3826834324, 0.9238795325),PAIR( 0.3368898534, 0.9415440652),00007400
PAIR( 0.2902846773, 0.9569403357),PAIR( 0.2429801799, 0.9700312532),00007500
PAIR( 0.1950903220, 0.9807852804),PAIR( 0.1467304745, 0.9891765100),00007600
PAIR( 0.0980171403, 0.9951847267),PAIR( 0.0490676743, 0.9987954562),00007700
 PAIR(0.0,1.0),PAIR(-.0490676743,.9987954562),                       00007800
PAIR(-0.0980171403, 0.9951847267),PAIR(-0.1467304745, 0.9891765100),00007900
PAIR(-0.1950903220, 0.9807852804),PAIR(-0.2429801799, 0.9700312532),00008000
PAIR(-0.2902846773, 0.9569403357),PAIR(-0.3368898534, 0.9415440652),00008100
PAIR(-0.3826834324, 0.9238795325),PAIR(-0.4275550934, 0.9039892931),00008200
PAIR(-0.4713967368, 0.8819212644),PAIR(-0.5141027442, 0.8577286100),00008300
PAIR(-0.5555702330, 0.8314696123),PAIR(-0.5956993045, 0.8032075315),00008400
PAIR(-0.6343932842, 0.7730104534),PAIR(-0.6715589548, 0.7409511254),00008500
PAIR(-0.7071067812, 0.7071067812),PAIR(-0.7409511254, 0.6715589548),00008600
PAIR(-0.7730104534, 0.6343932842),PAIR(-0.8032075315, 0.5956993045),00008700
PAIR(-0.8314696123, 0.5555702330),PAIR(-0.8577286100, 0.5141027442),00008800
PAIR(-0.8819212644, 0.4713967368),PAIR(-0.9039892931, 0.4275550934),00008900
PAIR(-0.9238795325, 0.3826834324),PAIR(-0.9415440652, 0.3368898534),00009000
PAIR(-0.9569403357, 0.2902846773),PAIR(-0.9700312532, 0.2429801799),00009100
PAIR(-0.9807852804, 0.1950903220),PAIR(-0.9891765100, 0.1467304745),00009200
PAIR(-0.9951847267, 0.0980171403),PAIR(-0.9987954562, 0.0490676743),00009300
 PAIR(1.0,0.0),PAIR(.9996988187,.0245412285),                        00009400
PAIR( 0.9987954562, 0.0490676743),PAIR( 0.9972904567, 0.0735645636),00009500
PAIR( 0.9951847267, 0.0980171403),PAIR( 0.9924795346, 0.1224106752),00009600
PAIR( 0.9891765100, 0.1467304745),PAIR( 0.9852776424, 0.1709618888),00009700
PAIR( 0.9807852804, 0.1950903220),PAIR( 0.9757021300, 0.2191012402),00009800
PAIR( 0.9700312532, 0.2429801799),PAIR( 0.9637760658, 0.2667127575),00009900
PAIR( 0.9569403357, 0.2902846773),PAIR( 0.9495281806, 0.3136817404),00010000
PAIR( 0.9415440652, 0.3368898534),PAIR( 0.9329927988, 0.3598950365),00010100
PAIR( 0.9238795325, 0.3826834324),PAIR( 0.9142097557, 0.4052413140),00010200
PAIR( 0.9039892931, 0.4275550934),PAIR( 0.8932243012, 0.4496113297),00010300
PAIR( 0.8819212644, 0.4713967368),PAIR( 0.8700869911, 0.4928981922),00010400
PAIR( 0.8577286100, 0.5141027442),PAIR( 0.8448535653, 0.5349976199),00010500
PAIR( 0.8314696123, 0.5555702330),PAIR( 0.8175848132, 0.5758081914),00010600
PAIR( 0.8032075315, 0.5956993045),PAIR( 0.7883464276, 0.6152315906),00010700
PAIR( 0.7730104534, 0.6343932842),PAIR( 0.7572088465, 0.6531728430),00010800
PAIR( 0.7409511254, 0.6715589548),PAIR( 0.7242470830, 0.6895405447),00010900
PAIR( 0.7071067812, 0.7071067812),PAIR( 0.6895405447, 0.7242470830),00011000
PAIR( 0.6715589548, 0.7409511254),PAIR( 0.6531728430, 0.7572088465),00011100
PAIR( 0.6343932842, 0.7730104534),PAIR( 0.6152315906, 0.7883464276),00011200
```

```
PAIR( 0.5956993045, 0.8032075315),PAIR( 0.5758081914, 0.8175848132),00011300
PAIR( 0.5555702330, 0.8314696123),PAIR( 0.5349976199, 0.8448535653),00011400
PAIR( 0.5141027442, 0.8577286100),PAIR( 0.4928981922, 0.8700869911),00011500
PAIR( 0.4713967368, 0.8819212644),PAIR( 0.4496113297, 0.8932243012),00011600
PAIR( 0.4275550934, 0.9039892931),PAIR( 0.4052413140, 0.9142097557),00011700
PAIR( 0.3826834324, 0.9238795325),PAIR( 0.3598950365, 0.9329927988),00011800
PAIR( 0.3368898534, 0.9415440652),PAIR( 0.3136817404, 0.9495281806),00011900
PAIR( 0.2902846773, 0.9569403357),PAIR( 0.2667127575, 0.9637760658),00012000
PAIR( 0.2429801799, 0.9700312532),PAIR( 0.2191012402, 0.9757021300),00012100
PAIR( 0.1950903220, 0.9807852804),PAIR( 0.1709618888, 0.9852776424),00012200
PAIR( 0.1467304745, 0.9891765100),PAIR( 0.1224106752, 0.9924795346),00012300
PAIR( 0.0980171403, 0.9951847267),PAIR( 0.0735645636, 0.9972904567),00012400
PAIR( 0.0490676743, 0.9987954562),PAIR( 0.0245412285, 0.9996988187),00012500
 PAIR(1.0,0.0),PAIR(.9999247018,.0122715383),                        00012600
PAIR( 0.9996988187, 0.0245412285),PAIR( 0.9993223846, 0.0368072229),00012700
PAIR( 0.9987954562, 0.0490676743),PAIR( 0.9981181129, 0.0613207363),00012800
PAIR( 0.9972904567, 0.0735645636),PAIR( 0.9963126122, 0.0857973123),00012900
PAIR( 0.9951847267, 0.0980171403),PAIR( 0.9939069700, 0.1102222073),00013000
PAIR( 0.9924795346, 0.1224106752),PAIR( 0.9909026354, 0.1345807085),00013100
PAIR( 0.9891765100, 0.1467304745),PAIR( 0.9873014182, 0.1588581433),00013200
PAIR( 0.9852776424, 0.1709618888),PAIR( 0.9831054874, 0.1830398880),00013300
PAIR( 0.9807852804, 0.1950903220),PAIR( 0.9783173707, 0.2071113762),00013400
PAIR( 0.9757021300, 0.2191012402),PAIR( 0.9729399522, 0.2310581083),00013500
PAIR( 0.9700312532, 0.2429801799),PAIR( 0.9669764710, 0.2548656596),00013600
PAIR( 0.9637760658, 0.2667127575),PAIR( 0.9604305194, 0.2785196894),00013700
PAIR( 0.9569403357, 0.2902846773),PAIR( 0.9533060404, 0.3020059493),00013800
PAIR( 0.9495281806, 0.3136817404),PAIR( 0.9456073254, 0.3253102922),00013900
PAIR( 0.9415440652, 0.3368898534),PAIR( 0.9373390119, 0.3484186802),00014000
PAIR( 0.9329927988, 0.3598950365),PAIR( 0.9285060805, 0.3713171940),00014100
PAIR( 0.9238795325, 0.3826834324),PAIR( 0.9191138517, 0.3939920401),00014200
PAIR( 0.9142097557, 0.4052413140),PAIR( 0.9091679831, 0.4164295601),00014300
PAIR( 0.9039892931, 0.4275550934),PAIR( 0.8986744657, 0.4386162385),00014400
PAIR( 0.8932243012, 0.4496113297),PAIR( 0.8876396204, 0.4605387110),00014500
PAIR( 0.8819212644, 0.4713967368),PAIR( 0.8760700942, 0.4821837721),00014600
PAIR( 0.8700869911, 0.4928981922),PAIR( 0.8639728561, 0.5035383837),00014700
PAIR( 0.8577286100, 0.5141027442),PAIR( 0.8513551931, 0.5245896827),00014800
PAIR( 0.8448535653, 0.5349976199),PAIR( 0.8382247056, 0.5453249884),00014900
PAIR( 0.8314696123, 0.5555702330),PAIR( 0.8245893028, 0.5657318108),00015000
PAIR( 0.8175848132, 0.5758081914),PAIR( 0.8104571983, 0.5857978575),00015100
PAIR( 0.8032075315, 0.5956993045),PAIR( 0.7958369046, 0.6055110414),00015200
PAIR( 0.7883464276, 0.6152315906),PAIR( 0.7807372286, 0.6248594881),00015300
PAIR( 0.7730104534, 0.6343932842),PAIR( 0.7651672656, 0.6438315429),00015400
PAIR( 0.7572088465, 0.6531728430),PAIR( 0.7491363945, 0.6624157776),00015500
PAIR( 0.7409511254, 0.6715589548),PAIR( 0.7326542717, 0.6806009978),00015600
PAIR( 0.7242470830, 0.6895405447),PAIR( 0.7157308253, 0.6983762494),00015700
PAIR( 0.7071067812, 0.7071067812),PAIR( 0.6983762494, 0.7157308253),00015800
PAIR( 0.6895405447, 0.7242470830),PAIR( 0.6806009978, 0.7326542717),00015900
PAIR( 0.6715589548, 0.7409511254),PAIR( 0.6624157776, 0.7491363945),00016000
PAIR( 0.6531728430, 0.7572088465),PAIR( 0.6438315429, 0.7651672656),00016100
PAIR( 0.6343932842, 0.7730104534),PAIR( 0.6248594881, 0.7807372286),00016200
PAIR( 0.6152315906, 0.7883464276),PAIR( 0.6055110414, 0.7958369046),00016300
PAIR( 0.5956993045, 0.8032075315),PAIR( 0.5857978575, 0.8104571983),00016400
PAIR( 0.5758081914, 0.8175848132),PAIR( 0.5657318108, 0.8245893028),00016500
PAIR( 0.5555702330, 0.8314696123),PAIR( 0.5453249884, 0.8382247056),00016600
PAIR( 0.5349976199, 0.8448535653),PAIR( 0.5245896827, 0.8513551931),00016700
PAIR( 0.5141027442, 0.8577286100),PAIR( 0.5035383837, 0.8639728561),00016800
PAIR( 0.4928981922, 0.8700869911),PAIR( 0.4821837721, 0.8760700942),00016900
```

```
PAIR( 0.4713967368, 0.8819212644),PAIR( 0.4605387110, 0.8876396204),00017000
PAIR( 0.4496113297, 0.8932243012),PAIR( 0.4386162385, 0.8986744657),00017100
PAIR( 0.4275550934, 0.9039892931),PAIR( 0.4164295601, 0.9091679831),00017200
PAIR( 0.4052413140, 0.9142097557),PAIR( 0.3939920401, 0.9191138517),00017300
PAIR( 0.3826834324, 0.9238795325),PAIR( 0.3713171940, 0.9285060805),00017400
PAIR( 0.3598950365, 0.9329927988),PAIR( 0.3484186802, 0.9373390119),00017500
PAIR( 0.3368898534, 0.9415440652),PAIR( 0.3253102922, 0.9456073254),00017600
PAIR( 0.3136817404, 0.9495281806),PAIR( 0.3020059493, 0.9533060404),00017700
PAIR( 0.2902846773, 0.9569403357),PAIR( 0.2785196894, 0.9604305194),00017800
PAIR( 0.2667127575, 0.9637760658),PAIR( 0.2548656596, 0.9669764710),00017900
PAIR( 0.2429801799, 0.9700312532),PAIR( 0.2310581083, 0.9729399522),00018000
PAIR( 0.2191012402, 0.9757021300),PAIR( 0.2071113762, 0.9783173707),00018100
PAIR( 0.1950903220, 0.9807852804),PAIR( 0.1830398880, 0.9831054874),00018200
PAIR( 0.1709618888, 0.9852776424),PAIR( 0.1588581433, 0.9873014182),00018300
PAIR( 0.1467304745, 0.9891765100),PAIR( 0.1345807085, 0.9909026354),00018400
PAIR( 0.1224106752, 0.9924795346),PAIR( 0.1102222073, 0.9939069700),00018500
PAIR( 0.0980171403, 0.9951847267),PAIR( 0.0857973123, 0.9963126122),00018600
PAIR( 0.0735645636, 0.9972904567),PAIR( 0.0613207363, 0.9981181129),00018700
PAIR( 0.0490676743, 0.9987954562),PAIR( 0.0368072229, 0.9993223846),00018800
PAIR( 0.0245412285, 0.9996988187),PAIR( 0.0122715383, 0.9999247018),00018900
 PAIR(1.0,0.0),PAIR(.9999811753,.0061358846),                          00019000
PAIR( 0.9999247018, 0.0122715383),PAIR( 0.9998305818, 0.0184067299),00019100
PAIR( 0.9996988187, 0.0245412285),PAIR( 0.9995294175, 0.0306748032),00019200
PAIR( 0.9993223846, 0.0368072229),PAIR( 0.9990777278, 0.0429382569),00019300
PAIR( 0.9987954562, 0.0490676743),PAIR( 0.9984755806, 0.0551952443),00019400
PAIR( 0.9981181129, 0.0613207363),PAIR( 0.9977230666, 0.0674439196),00019500
PAIR( 0.9972904567, 0.0735645636),PAIR( 0.9968202993, 0.0796824380),00019600
PAIR( 0.9963126122, 0.0857973123),PAIR( 0.9957674145, 0.0919089565),00019700
PAIR( 0.9951847267, 0.0980171403),PAIR( 0.9945645707, 0.1041216339),00019800
PAIR( 0.9939069700, 0.1102222073),PAIR( 0.9932119492, 0.1163186309),00019900
PAIR( 0.9924795346, 0.1224106752),PAIR( 0.9917097537, 0.1284981108),00020000
PAIR( 0.9909026354, 0.1345807085),PAIR( 0.9900582103, 0.1406582393),00020100
PAIR( 0.9891765100, 0.1467304745),PAIR( 0.9882575677, 0.1527971853),00020200
PAIR( 0.9873014182, 0.1588581433),PAIR( 0.9863080972, 0.1649131205),00020300
PAIR( 0.9852776424, 0.1709618888),PAIR( 0.9842100924, 0.1770042204),00020400
PAIR( 0.9831054874, 0.1830398880),PAIR( 0.9819638691, 0.1890686641),00020500
PAIR( 0.9807852804, 0.1950903220),PAIR( 0.9795697657, 0.2011046348),00020600
PAIR( 0.9783173707, 0.2071113762),PAIR( 0.9770281427, 0.2131103199),00020700
PAIR( 0.9757021300, 0.2191012402),PAIR( 0.9743393828, 0.2250839114),00020800
PAIR( 0.9729399522, 0.2310581083),PAIR( 0.9715038910, 0.2370236060),00020900
PAIR( 0.9700312532, 0.2429801799),PAIR( 0.9685220943, 0.2489276057),00021000
PAIR( 0.9669764710, 0.2548656596),PAIR( 0.9653944417, 0.2607941179),00021100
PAIR( 0.9637760658, 0.2667127575),PAIR( 0.9621214043, 0.2726213555),00021200
PAIR( 0.9604305194, 0.2785196894),PAIR( 0.9587034749, 0.2844075372),00021300
PAIR( 0.9569403357, 0.2902846773),PAIR( 0.9551411683, 0.2961508882),00021400
PAIR( 0.9533060404, 0.3020059493),PAIR( 0.9514350210, 0.3078496400),00021500
PAIR( 0.9495281806, 0.3136817404),PAIR( 0.9475855910, 0.3195020308),00021600
PAIR( 0.9456073254, 0.3253102922),PAIR( 0.9435934582, 0.3311063058),00021700
PAIR( 0.9415440652, 0.3368898534),PAIR( 0.9394592236, 0.3426607173),00021800
PAIR( 0.9373390119, 0.3484186802),PAIR( 0.9351835099, 0.3541635254),00021900
PAIR( 0.9329927988, 0.3598950365),PAIR( 0.9307669611, 0.3656129978),00022000
PAIR( 0.9285060805, 0.3713171940),PAIR( 0.9262102421, 0.3770074102),00022100
PAIR( 0.9238795325, 0.3826834324),PAIR( 0.9215140393, 0.3883450467),00022200
PAIR( 0.9191138517, 0.3939920401),PAIR( 0.9166790599, 0.3996241998),00022300
PAIR( 0.9142097557, 0.4052413140),PAIR( 0.9117060320, 0.4108431711),00022400
PAIR( 0.9091679831, 0.4164295601),PAIR( 0.9065957045, 0.4220002708),00022500
PAIR( 0.9039892931, 0.4275550934),PAIR( 0.9013488470, 0.4330938189),00022600
```

```
PAIR( 0.8986744657, 0.4386162385),PAIR( 0.8959662498, 0.4441221446),00022700
PAIR( 0.8932243012, 0.4496113297),PAIR( 0.8904487232, 0.4550835871),00022800
PAIR( 0.8876396204, 0.4605387110),PAIR( 0.8847970984, 0.4659764958),00022900
PAIR( 0.8819212644, 0.4713967368),PAIR( 0.8790122264, 0.4767992301),00023000
PAIR( 0.8760700942, 0.4821837721),PAIR( 0.8730949784, 0.4875501601),00023100
PAIR( 0.8700869911, 0.4928981922),PAIR( 0.8670462455, 0.4982276670),00023200
PAIR( 0.8639728561, 0.5035383837),PAIR( 0.8608669386, 0.5088301425),00023300
PAIR( 0.8577286100, 0.5141027442),PAIR( 0.8545579884, 0.5193559902),00023400
PAIR( 0.8513551931, 0.5245896827),PAIR( 0.8481203448, 0.5298036247),00023500
PAIR( 0.8448535653, 0.5349976199),PAIR( 0.8415549774, 0.5401714727),00023600
PAIR( 0.8382247056, 0.5453249884),PAIR( 0.8348628750, 0.5504579729),00023700
PAIR( 0.8314696123, 0.5555702330),PAIR( 0.8280450453, 0.5606615762),00023800
PAIR( 0.8245893028, 0.5657318108),PAIR( 0.8211025150, 0.5707807459),00023900
PAIR( 0.8175848132, 0.5758081914),PAIR( 0.8140363297, 0.5808139581),00024000
PAIR( 0.8104571983, 0.5857978575),PAIR( 0.8068475535, 0.5907597019),00024100
PAIR( 0.8032075315, 0.5956993045),PAIR( 0.7995372691, 0.6006164794),00024200
PAIR( 0.7958369046, 0.6055110414),PAIR( 0.7921065773, 0.6103828063),00024300
PAIR( 0.7883464276, 0.6152315906),PAIR( 0.7845565972, 0.6200572118),00024400
PAIR( 0.7807372286, 0.6248594881),PAIR( 0.7768884657, 0.6296382389),00024500
PAIR( 0.7730104534, 0.6343932842),PAIR( 0.7691033376, 0.6391244449),00024600
PAIR( 0.7651672656, 0.6438315429),PAIR( 0.7612023855, 0.6485144010),00024700
PAIR( 0.7572088465, 0.6531728430),PAIR( 0.7531867990, 0.6578066933),00024800
PAIR( 0.7491363945, 0.6624157776),PAIR( 0.7450577854, 0.6669999223),00024900
PAIR( 0.7409511254, 0.6715589548),PAIR( 0.7368165689, 0.6760927036),00025000
PAIR( 0.7326542717, 0.6806009978),PAIR( 0.7284643904, 0.6850836678),00025100
PAIR( 0.7242470830, 0.6895405447),PAIR( 0.7200025080, 0.6939714609),00025200
PAIR( 0.7157308253, 0.6983762494),PAIR( 0.7114321957, 0.7027547445),00025300
PAIR( 0.7071067812, 0.7071067812),PAIR( 0.7027547445, 0.7114321957),00025400
PAIR( 0.6983762494, 0.7157308253),PAIR( 0.6939714609, 0.7200025080),00025500
PAIR( 0.6895405447, 0.7242470830),PAIR( 0.6850836678, 0.7284643904),00025600
PAIR( 0.6806009978, 0.7326542717),PAIR( 0.6760927036, 0.7368165689),00025700
PAIR( 0.6715589548, 0.7409511254),PAIR( 0.6669999223, 0.7450577854),00025800
PAIR( 0.6624157776, 0.7491363945),PAIR( 0.6578066933, 0.7531867990),00025900
PAIR( 0.6531728430, 0.7572088465),PAIR( 0.6485144010, 0.7612023855),00026000
PAIR( 0.6438315429, 0.7651672656),PAIR( 0.6391244449, 0.7691033376),00026100
PAIR( 0.6343932842, 0.7730104534),PAIR( 0.6296382389, 0.7768884657),00026200
PAIR( 0.6248594881, 0.7807372286),PAIR( 0.6200572118, 0.7845565972),00026300
PAIR( 0.6152315906, 0.7883464276),PAIR( 0.6103828063, 0.7921065773),00026400
PAIR( 0.6055110414, 0.7958369046),PAIR( 0.6006164794, 0.7995372691),00026500
PAIR( 0.5956993045, 0.8032075315),PAIR( 0.5907597019, 0.8068475535),00026600
PAIR( 0.5857978575, 0.8104571983),PAIR( 0.5808139581, 0.8140363297),00026700
PAIR( 0.5758081914, 0.8175848132),PAIR( 0.5707807459, 0.8211025150),00026800
PAIR( 0.5657318108, 0.8245893028),PAIR( 0.5606615762, 0.8280450453),00026900
PAIR( 0.5555702330, 0.8314696123),PAIR( 0.5504579729, 0.8348628750),00027000
PAIR( 0.5453249884, 0.8382247056),PAIR( 0.5401714727, 0.8415549774),00027100
PAIR( 0.5349976199, 0.8448535653),PAIR( 0.5298036247, 0.8481203448),00027200
PAIR( 0.5245896827, 0.8513551931),PAIR( 0.5193559902, 0.8545579884),00027300
PAIR( 0.5141027442, 0.8577286100),PAIR( 0.5088301425, 0.8608669386),00027400
PAIR( 0.5035383837, 0.8639728561),PAIR( 0.4982276670, 0.8670462455),00027500
PAIR( 0.4928981922, 0.8700869911),PAIR( 0.4875501601, 0.8730949784),00027600
PAIR( 0.4821837721, 0.8760700942),PAIR( 0.4767992301, 0.8790122264),00027700
PAIR( 0.4713967368, 0.8819212644),PAIR( 0.4659764958, 0.8847970984),00027800
PAIR( 0.4605387110, 0.8876396204),PAIR( 0.4550835871, 0.8904487232),00027900
PAIR( 0.4496113297, 0.8932243012),PAIR( 0.4441221446, 0.8959662498),00028000
PAIR( 0.4386162385, 0.8986744657),PAIR( 0.4330938189, 0.9013488470),00028100
PAIR( 0.4275550934, 0.9039892931),PAIR( 0.4220002708, 0.9065957045),00028200
PAIR( 0.4164295601, 0.9091679831),PAIR( 0.4108431711, 0.9117060320),00028300
```

```
PAIR( 0.4052413140, 0.9142097557),PAIR( 0.3996241998, 0.9166790599),00028400
PAIR( 0.3939920401, 0.9191138517),PAIR( 0.3883450467, 0.9215140393),00028500
PAIR( 0.3826834324, 0.9238795325),PAIR( 0.3770074102, 0.9262102421),00028600
PAIR( 0.3713171940, 0.9285060805),PAIR( 0.3656129978, 0.9307669611),00028700
PAIR( 0.3598950365, 0.9329927988),PAIR( 0.3541635254, 0.9351835099),00028800
PAIR( 0.3484186802, 0.9373390119),PAIR( 0.3426607173, 0.9394592236),00028900
PAIR( 0.3368898534, 0.9415440652),PAIR( 0.3311063058, 0.9435934582),00029000
PAIR( 0.3253102922, 0.9456073254),PAIR( 0.3195020308, 0.9475855910),00029100
PAIR( 0.3136817404, 0.9495281806),PAIR( 0.3078496400, 0.9514350210),00029200
PAIR( 0.3020059493, 0.9533060404),PAIR( 0.2961508882, 0.9551411683),00029300
PAIR( 0.2902846773, 0.9569403357),PAIR( 0.2844075372, 0.9587034749),00029400
PAIR( 0.2785196894, 0.9604305194),PAIR( 0.2726213555, 0.9621214043),00029500
PAIR( 0.2667127575, 0.9637760658),PAIR( 0.2607941179, 0.9653944417),00029600
PAIR( 0.2548656596, 0.9669764710),PAIR( 0.2489276057, 0.9685220943),00029700
PAIR( 0.2429801799, 0.9700312532),PAIR( 0.2370236060, 0.9715038910),00029800
PAIR( 0.2310581083, 0.9729399522),PAIR( 0.2250839114, 0.9743393828),00029900
PAIR( 0.2191012402, 0.9757021300),PAIR( 0.2131103199, 0.9770281427),00030000
PAIR( 0.2071113762, 0.9783173707),PAIR( 0.2011046348, 0.9795697657),00030100
PAIR( 0.1950903220, 0.9807852804),PAIR( 0.1890686641, 0.9819638691),00030200
PAIR( 0.1830398880, 0.9831054874),PAIR( 0.1770042204, 0.9842100924),00030300
PAIR( 0.1709618888, 0.9852776424),PAIR( 0.1649131205, 0.9863080972),00030400
PAIR( 0.1588581433, 0.9873014182),PAIR( 0.1527971853, 0.9882575677),00030500
PAIR( 0.1467304745, 0.9891765100),PAIR( 0.1406582393, 0.9900582103),00030600
PAIR( 0.1345807085, 0.9909026354),PAIR( 0.1284981108, 0.9917097537),00030700
PAIR( 0.1224106752, 0.9924795346),PAIR( 0.1163186309, 0.9932119492),00030800
PAIR( 0.1102222073, 0.9939069700),PAIR( 0.1041216339, 0.9945645707),00030900
PAIR( 0.0980171403, 0.9951847267),PAIR( 0.0919089565, 0.9957674145),00031000
PAIR( 0.0857973123, 0.9963126122),PAIR( 0.0796824380, 0.9968202993),00031100
PAIR( 0.0735645636, 0.9972904567),PAIR( 0.0674439196, 0.9977230666),00031200
PAIR( 0.0613207363, 0.9981181129),PAIR( 0.0551952443, 0.9984755806),00031300
PAIR( 0.0490676743, 0.9987954562),PAIR( 0.0429382569, 0.9990777278),00031400
PAIR( 0.0368072229, 0.9993223846),PAIR( 0.0306748032, 0.9995294175),00031500
PAIR( 0.0245412285, 0.9996988187),PAIR( 0.0184067299, 0.9998305818),00031600
PAIR( 0.0122715383, 0.9999247018),PAIR( 0.0061358846, 0.9999811753),00031700
 PAIR(1.0,0.0),PAIR(.9999952938,.0030679568),                        00031800
PAIR( 0.9999811753, 0.0061358846),PAIR( 0.9999576446, 0.0092037548),00031900
PAIR( 0.9999247018, 0.0122715383),PAIR( 0.9998823475, 0.0153392063),00032000
PAIR( 0.9998305818, 0.0184067299),PAIR( 0.9997694053, 0.0214740803),00032100
PAIR( 0.9996988187, 0.0245412285),PAIR( 0.9996188225, 0.0276081458),00032200
PAIR( 0.9995294175, 0.0306748032),PAIR( 0.9994306046, 0.0337411719),00032300
PAIR( 0.9993223846, 0.0368072229),PAIR( 0.9992047586, 0.0398729276),00032400
PAIR( 0.9990777278, 0.0429382569),PAIR( 0.9989412932, 0.0460031821),00032500
PAIR( 0.9987954562, 0.0490676743),PAIR( 0.9986402182, 0.0521317047),00032600
PAIR( 0.9984755806, 0.0551952443),PAIR( 0.9983015449, 0.0582582645),00032700
PAIR( 0.9981181129, 0.0613207363),PAIR( 0.9979252862, 0.0643826309),00032800
PAIR( 0.9977230666, 0.0674439196),PAIR( 0.9975114561, 0.0705045734),00032900
PAIR( 0.9972904567, 0.0735645636),PAIR( 0.9970600703, 0.0766238614),00033000
PAIR( 0.9968202993, 0.0796824380),PAIR( 0.9965711458, 0.0827402645),00033100
PAIR( 0.9963126122, 0.0857973123),PAIR( 0.9960447009, 0.0888535526),00033200
PAIR( 0.9957674145, 0.0919089565),PAIR( 0.9954807555, 0.0949634953),00033300
PAIR( 0.9951847267, 0.0980171403),PAIR( 0.9948793308, 0.1010698628),00033400
PAIR( 0.9945645707, 0.1041216339),PAIR( 0.9942404495, 0.1071724250),00033500
PAIR( 0.9939069700, 0.1102222073),PAIR( 0.9935641355, 0.1132709522),00033600
PAIR( 0.9932119492, 0.1163186309),PAIR( 0.9928504145, 0.1193652148),00033700
PAIR( 0.9924795346, 0.1224106752),PAIR( 0.9920993131, 0.1254549834),00033800
PAIR( 0.9917097537, 0.1284981108),PAIR( 0.9913108598, 0.1315400287),00033900
PAIR( 0.9909026354, 0.1345807085),PAIR( 0.9904850843, 0.1376201216),00034000
```

```
PAIR( 0.9900582103, 0.1406582393),PAIR( 0.9896220175, 0.1436950332),00034100
PAIR( 0.9891765100, 0.1467304745),PAIR( 0.9887216920, 0.1497645347),00034200
PAIR( 0.9882575677, 0.1527971853),PAIR( 0.9877841416, 0.1558283977),00034300
PAIR( 0.9873014182, 0.1588581433),PAIR( 0.9868094018, 0.1618863938),00034400
PAIR( 0.9863080972, 0.1649131205),PAIR( 0.9857975092, 0.1679382950),00034500
PAIR( 0.9852776424, 0.1709618888),PAIR( 0.9847485018, 0.1739838734),00034600
PAIR( 0.9842100924, 0.1770042204),PAIR( 0.9836624192, 0.1800229014),00034700
PAIR( 0.9831054874, 0.1830398880),PAIR( 0.9825393023, 0.1860551517),00034800
PAIR( 0.9819638691, 0.1890686641),PAIR( 0.9813791933, 0.1920803970),00034900
PAIR( 0.9807852804, 0.1950903220),PAIR( 0.9801821360, 0.1980984107),00035000
PAIR( 0.9795697657, 0.2011046348),PAIR( 0.9789481753, 0.2041089661),00035100
PAIR( 0.9783173707, 0.2071113762),PAIR( 0.9776773578, 0.2101118369),00035200
PAIR( 0.9770281427, 0.2131103199),PAIR( 0.9763697313, 0.2161067971),00035300
PAIR( 0.9757021300, 0.2191012402),PAIR( 0.9750253451, 0.2220936210),00035400
PAIR( 0.9743393828, 0.2250839114),PAIR( 0.9736442497, 0.2280720832),00035500
PAIR( 0.9729399522, 0.2310581083),PAIR( 0.9722264971, 0.2340419586),00035600
PAIR( 0.9715038910, 0.2370236060),PAIR( 0.9707721407, 0.2400030224),00035700
PAIR( 0.9700312532, 0.2429801799),PAIR( 0.9692812354, 0.2459550503),00035800
PAIR( 0.9685220943, 0.2489276057),PAIR( 0.9677538371, 0.2518978182),00035900
PAIR( 0.9669764710, 0.2548656596),PAIR( 0.9661900034, 0.2578311022),00036000
PAIR( 0.9653944417, 0.2607941179),PAIR( 0.9645897933, 0.2637546790),00036100
PAIR( 0.9637760658, 0.2667127575),PAIR( 0.9629532669, 0.2696683256),00036200
PAIR( 0.9621214043, 0.2726213555),PAIR( 0.9612804858, 0.2755718193),00036300
PAIR( 0.9604305194, 0.2785196894),PAIR( 0.9595715131, 0.2814649379),00036400
PAIR( 0.9587034749, 0.2844075372),PAIR( 0.9578264130, 0.2873474595),00036500
PAIR( 0.9569403357, 0.2902846773),PAIR( 0.9560452513, 0.2932191627),00036600
PAIR( 0.9551411683, 0.2961508882),PAIR( 0.9542280951, 0.2990798263),00036700
PAIR( 0.9533060404, 0.3020059493),PAIR( 0.9523750127, 0.3049292297),00036800
PAIR( 0.9514350210, 0.3078496400),PAIR( 0.9504860739, 0.3107671527),00036900
PAIR( 0.9495281806, 0.3136817404),PAIR( 0.9485613499, 0.3165933756),00037000
PAIR( 0.9475855910, 0.3195020308),PAIR( 0.9466009131, 0.3224076788),00037100
PAIR( 0.9456073254, 0.3253102922),PAIR( 0.9446048373, 0.3282098436),00037200
PAIR( 0.9435934582, 0.3311063058),PAIR( 0.9425731976, 0.3339996514),00037300
PAIR( 0.9415440652, 0.3368898534),PAIR( 0.9405060706, 0.3397768844),00037400
PAIR( 0.9394592236, 0.3426607173),PAIR( 0.9384035341, 0.3455413250),00037500
PAIR( 0.9373390119, 0.3484186802),PAIR( 0.9362656672, 0.3512927561),00037600
PAIR( 0.9351835099, 0.3541635254),PAIR( 0.9340925504, 0.3570309612),00037700
PAIR( 0.9329927988, 0.3598950365),PAIR( 0.9318842656, 0.3627557244),00037800
PAIR( 0.9307669611, 0.3656129978),PAIR( 0.9296408958, 0.3684668300),00037900
PAIR( 0.9285060805, 0.3713171940),PAIR( 0.9273625256, 0.3741640630),00038000
PAIR( 0.9262102421, 0.3770074102),PAIR( 0.9250492408, 0.3798472089),00038100
PAIR( 0.9238795325, 0.3826834324),PAIR( 0.9227011283, 0.3855160538),00038200
PAIR( 0.9215140393, 0.3883450467),PAIR( 0.9203182767, 0.3911703843),00038300
PAIR( 0.9191138517, 0.3939920401),PAIR( 0.9179007756, 0.3968099874),00038400
PAIR( 0.9166790599, 0.3996241998),PAIR( 0.9154487161, 0.4024346509),00038500
PAIR( 0.9142097557, 0.4052413140),PAIR( 0.9129621904, 0.4080441629),00038600
PAIR( 0.9117060320, 0.4108431711),PAIR( 0.9104412923, 0.4136383122),00038700
PAIR( 0.9091679831, 0.4164295601),PAIR( 0.9078861165, 0.4192168884),00038800
PAIR( 0.9065957045, 0.4220002708),PAIR( 0.9052967593, 0.4247796812),00038900
PAIR( 0.9039892931, 0.4275550934),PAIR( 0.9026733182, 0.4303264813),00039000
PAIR( 0.9013488470, 0.4330938189),PAIR( 0.9000158920, 0.4358570799),00039100
PAIR( 0.8986744657, 0.4386162385),PAIR( 0.8973245807, 0.4413712687),00039200
PAIR( 0.8959662498, 0.4441221446),PAIR( 0.8945994856, 0.4468688402),00039300
PAIR( 0.8932243012, 0.4496113297),PAIR( 0.8918407094, 0.4523495872),00039400
PAIR( 0.8904487232, 0.4550835871),PAIR( 0.8890483558, 0.4578133036),00039500
PAIR( 0.8876396204, 0.4605387110),PAIR( 0.8862225301, 0.4632597836),00039600
PAIR( 0.8847970984, 0.4659764958),PAIR( 0.8833633387, 0.4686888220),00039700
```

```
PAIR( 0.8819212644, 0.4713967368),PAIR( 0.8804708891, 0.4741002146),00039800
PAIR( 0.8790122264, 0.4767992301),PAIR( 0.8775452902, 0.4794937577),00039900
PAIR( 0.8760700942, 0.4821837721),PAIR( 0.8745866523, 0.4848692480),00040000
PAIR( 0.8730949784, 0.4875501601),PAIR( 0.8715950867, 0.4902264833),00040100
PAIR( 0.8700869911, 0.4928981922),PAIR( 0.8685707060, 0.4955652618),00040200
PAIR( 0.8670462455, 0.4982276670),PAIR( 0.8655136241, 0.5008853826),00040300
PAIR( 0.8639728561, 0.5035383837),PAIR( 0.8624239561, 0.5061866453),00040400
PAIR( 0.8608669386, 0.5088301425),PAIR( 0.8593018184, 0.5114688504),00040500
PAIR( 0.8577286100, 0.5141027442),PAIR( 0.8561473284, 0.5167317990),00040600
PAIR( 0.8545579884, 0.5193559902),PAIR( 0.8529606049, 0.5219752929),00040700
PAIR( 0.8513551931, 0.5245896827),PAIR( 0.8497417680, 0.5271991348),00040800
PAIR( 0.8481203448, 0.5298036247),PAIR( 0.8464909388, 0.5324031279),00040900
PAIR( 0.8448535653, 0.5349976199),PAIR( 0.8432082396, 0.5375870763),00041000
PAIR( 0.8415549774, 0.5401714727),PAIR( 0.8398937942, 0.5427507849),00041100
PAIR( 0.8382247056, 0.5453249884),PAIR( 0.8365477272, 0.5478940592),00041200
PAIR( 0.8348628750, 0.5504579729),PAIR( 0.8331701647, 0.5530167056),00041300
PAIR( 0.8314696123, 0.5555702330),PAIR( 0.8297612338, 0.5581185312),00041400
PAIR( 0.8280450453, 0.5606615762),PAIR( 0.8263210628, 0.5631993440),00041500
PAIR( 0.8245893028, 0.5657318108),PAIR( 0.8228497814, 0.5682589527),00041600
PAIR( 0.8211025150, 0.5707807459),PAIR( 0.8193475201, 0.5732971667),00041700
PAIR( 0.8175848132, 0.5758081914),PAIR( 0.8158144108, 0.5783137964),00041800
PAIR( 0.8140363297, 0.5808139581),PAIR( 0.8122505866, 0.5833086529),00041900
PAIR( 0.8104571983, 0.5857978575),PAIR( 0.8086561816, 0.5882815482),00042000
PAIR( 0.8068475535, 0.5907597019),PAIR( 0.8050313311, 0.5932322950),00042100
PAIR( 0.8032075315, 0.5956993045),PAIR( 0.8013761717, 0.5981607070),00042200
PAIR( 0.7995372691, 0.6006164794),PAIR( 0.7976908409, 0.6030665985),00042300
PAIR( 0.7958369046, 0.6055110414),PAIR( 0.7939754776, 0.6079497850),00042400
PAIR( 0.7921065773, 0.6103828063),PAIR( 0.7902302214, 0.6128100824),00042500
PAIR( 0.7883464276, 0.6152315906),PAIR( 0.7864552136, 0.6176473079),00042600
PAIR( 0.7845565972, 0.6200572118),PAIR( 0.7826505962, 0.6224612794),00042700
PAIR( 0.7807372286, 0.6248594881),PAIR( 0.7788165124, 0.6272518155),00042800
PAIR( 0.7768884657, 0.6296382389),PAIR( 0.7749531066, 0.6320187359),00042900
PAIR( 0.7730104534, 0.6343932842),PAIR( 0.7710605243, 0.6367618612),00043000
PAIR( 0.7691033376, 0.6391244449),PAIR( 0.7671389119, 0.6414810128),00043100
PAIR( 0.7651672656, 0.6438315429),PAIR( 0.7631884173, 0.6461760130),00043200
PAIR( 0.7612023855, 0.6485144010),PAIR( 0.7592091890, 0.6508466850),00043300
PAIR( 0.7572088465, 0.6531728430),PAIR( 0.7552013769, 0.6554928530),00043400
PAIR( 0.7531867990, 0.6578066933),PAIR( 0.7511651319, 0.6601143421),00043500
PAIR( 0.7491363945, 0.6624157776),PAIR( 0.7471006060, 0.6647109782),00043600
PAIR( 0.7450577854, 0.6669999223),PAIR( 0.7430079521, 0.6692825883),00043700
PAIR( 0.7409511254, 0.6715589548),PAIR( 0.7388873245, 0.6738290004),00043800
PAIR( 0.7368165689, 0.6760927036),PAIR( 0.7347388781, 0.6783500431),00043900
PAIR( 0.7326542717, 0.6806009978),PAIR( 0.7305627692, 0.6828455464),00044000
PAIR( 0.7284643904, 0.6850836678),PAIR( 0.7263591551, 0.6873153409),00044100
PAIR( 0.7242470830, 0.6895405447),PAIR( 0.7221281939, 0.6917592584),00044200
PAIR( 0.7200025080, 0.6939714609),PAIR( 0.7178700451, 0.6961771315),00044300
PAIR( 0.7157308253, 0.6983762494),PAIR( 0.7135848688, 0.7005687939),00044400
PAIR( 0.7114321957, 0.7027547445),PAIR( 0.7092728264, 0.7049340804),00044500
PAIR( 0.7071067812, 0.7071067812),PAIR( 0.7049340804, 0.7092728264),00044600
PAIR( 0.7027547445, 0.7114321957),PAIR( 0.7005687939, 0.7135848688),00044700
PAIR( 0.6983762494, 0.7157308253),PAIR( 0.6961771315, 0.7178700451),00044800
PAIR( 0.6939714609, 0.7200025080),PAIR( 0.6917592584, 0.7221281939),00044900
PAIR( 0.6895405447, 0.7242470830),PAIR( 0.6873153409, 0.7263591551),00045000
PAIR( 0.6850836678, 0.7284643904),PAIR( 0.6828455464, 0.7305627692),00045100
PAIR( 0.6806009978, 0.7326542717),PAIR( 0.6783500431, 0.7347388781),00045200
PAIR( 0.6760927036, 0.7368165689),PAIR( 0.6738290004, 0.7388873245),00045300
PAIR( 0.6715589548, 0.7409511254),PAIR( 0.6692825883, 0.7430079521),00045400
```

```
PAIR( 0.6669999223, 0.7450577854),PAIR( 0.6647109782, 0.7471006060),00045500
PAIR( 0.6624157776, 0.7491363945),PAIR( 0.6601143421, 0.7511651319),00045600
PAIR( 0.6578066933, 0.7531867990),PAIR( 0.6554928530, 0.7552013769),00045700
PAIR( 0.6531728430, 0.7572088465),PAIR( 0.6508466850, 0.7592091890),00045800
PAIR( 0.6485144010, 0.7612023855),PAIR( 0.6461760130, 0.7631884173),00045900
PAIR( 0.6438315429, 0.7651672656),PAIR( 0.6414810128, 0.7671389119),00046000
PAIR( 0.6391244449, 0.7691033376),PAIR( 0.6367618612, 0.7710605243),00046100
PAIR( 0.6343932842, 0.7730104534),PAIR( 0.6320187359, 0.7749531066),00046200
PAIR( 0.6296382389, 0.7768884657),PAIR( 0.6272518155, 0.7788165124),00046300
PAIR( 0.6248594881, 0.7807372286),PAIR( 0.6224612794, 0.7826505962),00046400
PAIR( 0.6200572118, 0.7845565972),PAIR( 0.6176473079, 0.7864552136),00046500
PAIR( 0.6152315906, 0.7883464276),PAIR( 0.6128100824, 0.7902302214),00046600
PAIR( 0.6103828063, 0.7921065773),PAIR( 0.6079497850, 0.7939754776),00046700
PAIR( 0.6055110414, 0.7958369046),PAIR( 0.6030665985, 0.7976908409),00046800
PAIR( 0.6006164794, 0.7995372691),PAIR( 0.5981607070, 0.8013761717),00046900
PAIR( 0.5956993045, 0.8032075315),PAIR( 0.5932322950, 0.8050313311),00047000
PAIR( 0.5907597019, 0.8068475535),PAIR( 0.5882815482, 0.8086561816),00047100
PAIR( 0.5857978575, 0.8104571983),PAIR( 0.5833086529, 0.8122505866),00047200
PAIR( 0.5808139581, 0.8140363297),PAIR( 0.5783137964, 0.8158144108),00047300
PAIR( 0.5758081914, 0.8175848132),PAIR( 0.5732971667, 0.8193475201),00047400
PAIR( 0.5707807459, 0.8211025150),PAIR( 0.5682589527, 0.8228497814),00047500
PAIR( 0.5657318108, 0.8245893028),PAIR( 0.5631993440, 0.8263210628),00047600
PAIR( 0.5606615762, 0.8280450453),PAIR( 0.5581185312, 0.8297612338),00047700
PAIR( 0.5555702330, 0.8314696123),PAIR( 0.5530167056, 0.8331701647),00047800
PAIR( 0.5504579729, 0.8348628750),PAIR( 0.5478940592, 0.8365477272),00047900
PAIR( 0.5453249884, 0.8382247056),PAIR( 0.5427507849, 0.8398937942),00048000
PAIR( 0.5401714727, 0.8415549774),PAIR( 0.5375870763, 0.8432082396),00048100
PAIR( 0.5349976199, 0.8448535653),PAIR( 0.5324031279, 0.8464909388),00048200
PAIR( 0.5298036247, 0.8481203448),PAIR( 0.5271991348, 0.8497417680),00048300
PAIR( 0.5245896827, 0.8513551931),PAIR( 0.5219752929, 0.8529606049),00048400
PAIR( 0.5193559902, 0.8545579884),PAIR( 0.5167317990, 0.8561473284),00048500
PAIR( 0.5141027442, 0.8577286100),PAIR( 0.5114688504, 0.8593018184),00048600
PAIR( 0.5088301425, 0.8608669386),PAIR( 0.5061866453, 0.8624239561),00048700
PAIR( 0.5035383837, 0.8639728561),PAIR( 0.5008853826, 0.8655136241),00048800
PAIR( 0.4982276670, 0.8670462455),PAIR( 0.4955652618, 0.8685707060),00048900
PAIR( 0.4928981922, 0.8700869911),PAIR( 0.4902264833, 0.8715950867),00049000
PAIR( 0.4875501601, 0.8730949784),PAIR( 0.4848692480, 0.8745866523),00049100
PAIR( 0.4821837721, 0.8760700942),PAIR( 0.4794937577, 0.8775452902),00049200
PAIR( 0.4767992301, 0.8790122264),PAIR( 0.4741002146, 0.8804708891),00049300
PAIR( 0.4713967368, 0.8819212644),PAIR( 0.4686888220, 0.8833633387),00049400
PAIR( 0.4659764958, 0.8847970984),PAIR( 0.4632597836, 0.8862225301),00049500
PAIR( 0.4605387110, 0.8876396204),PAIR( 0.4578133036, 0.8890483558),00049600
PAIR( 0.4550835871, 0.8904487232),PAIR( 0.4523495872, 0.8918407094),00049700
PAIR( 0.4496113297, 0.8932243012),PAIR( 0.4468688402, 0.8945994856),00049800
PAIR( 0.4441221446, 0.8959662498),PAIR( 0.4413712687, 0.8973245807),00049900
PAIR( 0.4386162385, 0.8986744657),PAIR( 0.4358570799, 0.9000158920),00050000
PAIR( 0.4330938189, 0.9013488470),PAIR( 0.4303264813, 0.9026733182),00050100
PAIR( 0.4275550934, 0.9039892931),PAIR( 0.4247796812, 0.9052967593),00050200
PAIR( 0.4220002708, 0.9065957045),PAIR( 0.4192168884, 0.9078861165),00050300
PAIR( 0.4164295601, 0.9091679831),PAIR( 0.4136383122, 0.9104412923),00050400
PAIR( 0.4108431711, 0.9117060320),PAIR( 0.4080441629, 0.9129621904),00050500
PAIR( 0.4052413140, 0.9142097557),PAIR( 0.4024346509, 0.9154487161),00050600
PAIR( 0.3996241998, 0.9166790599),PAIR( 0.3968099874, 0.9179007756),00050700
PAIR( 0.3939920401, 0.9191138517),PAIR( 0.3911703843, 0.9203182767),00050800
PAIR( 0.3883450467, 0.9215140393),PAIR( 0.3855160538, 0.9227011283),00050900
PAIR( 0.3826834324, 0.9238795325),PAIR( 0.3798472089, 0.9250492408),00051000
PAIR( 0.3770074102, 0.9262102421),PAIR( 0.3741640630, 0.9273625256),00051100
```

```
PAIR( 0.3713171940, 0.9285060805),PAIR( 0.3684668300, 0.9296408958),00051200
PAIR( 0.3656129978, 0.9307669611),PAIR( 0.3627557244, 0.9318842656),00051300
PAIR( 0.3598950365, 0.9329927988),PAIR( 0.3570309612, 0.9340925504),00051400
PAIR( 0.3541635254, 0.9351835099),PAIR( 0.3512927561, 0.9362656672),00051500
PAIR( 0.3484186802, 0.9373390119),PAIR( 0.3455413250, 0.9384035341),00051600
PAIR( 0.3426607173, 0.9394592236),PAIR( 0.3397768844, 0.9405060706),00051700
PAIR( 0.3368898534, 0.9415440652),PAIR( 0.3339996514, 0.9425731976),00051800
PAIR( 0.3311063058, 0.9435934582),PAIR( 0.3282098436, 0.9446048373),00051900
PAIR( 0.3253102922, 0.9456073254),PAIR( 0.3224076788, 0.9466009131),00052000
PAIR( 0.3195020308, 0.9475855910),PAIR( 0.3165933756, 0.9485613499),00052100
PAIR( 0.3136817404, 0.9495281806),PAIR( 0.3107671527, 0.9504860739),00052200
PAIR( 0.3078496400, 0.9514350210),PAIR( 0.3049292297, 0.9523750127),00052300
PAIR( 0.3020059493, 0.9533060404),PAIR( 0.2990798263, 0.9542280951),00052400
PAIR( 0.2961508882, 0.9551411683),PAIR( 0.2932191627, 0.9560452513),00052500
PAIR( 0.2902846773, 0.9569403357),PAIR( 0.2873474595, 0.9578264130),00052600
PAIR( 0.2844075372, 0.9587034749),PAIR( 0.2814649379, 0.9595715131),00052700
PAIR( 0.2785196894, 0.9604305194),PAIR( 0.2755718193, 0.9612804858),00052800
PAIR( 0.2726213555, 0.9621214043),PAIR( 0.2696683256, 0.9629532669),00052900
PAIR( 0.2667127575, 0.9637760658),PAIR( 0.2637546790, 0.9645897933),00053000
PAIR( 0.2607941179, 0.9653944417),PAIR( 0.2578311022, 0.9661900034),00053100
PAIR( 0.2548656596, 0.9669764710),PAIR( 0.2518978182, 0.9677538371),00053200
PAIR( 0.2489276057, 0.9685220943),PAIR( 0.2459550503, 0.9692812354),00053300
PAIR( 0.2429801799, 0.9700312532),PAIR( 0.2400030224, 0.9707721407),00053400
PAIR( 0.2370236060, 0.9715038910),PAIR( 0.2340419586, 0.9722264971),00053500
PAIR( 0.2310581083, 0.9729399522),PAIR( 0.2280720832, 0.9736442497),00053600
PAIR( 0.2250839114, 0.9743393828),PAIR( 0.2220936210, 0.9750253451),00053700
PAIR( 0.2191012402, 0.9757021300),PAIR( 0.2161067971, 0.9763697313),00053800
PAIR( 0.2131103199, 0.9770281427),PAIR( 0.2101118369, 0.9776773578),00053900
PAIR( 0.2071113762, 0.9783173707),PAIR( 0.2041089661, 0.9789481753),00054000
PAIR( 0.2011046348, 0.9795697657),PAIR( 0.1980984107, 0.9801821360),00054100
PAIR( 0.1950903220, 0.9807852804),PAIR( 0.1920803970, 0.9813791933),00054200
PAIR( 0.1890686641, 0.9819638691),PAIR( 0.1860551517, 0.9825393023),00054300
PAIR( 0.1830398880, 0.9831054874),PAIR( 0.1800229014, 0.9836624192),00054400
PAIR( 0.1770042204, 0.9842100924),PAIR( 0.1739838734, 0.9847485018),00054500
PAIR( 0.1709618888, 0.9852776424),PAIR( 0.1679382950, 0.9857975092),00054600
PAIR( 0.1649131205, 0.9863080972),PAIR( 0.1618863938, 0.9868094018),00054700
PAIR( 0.1588581433, 0.9873014182),PAIR( 0.1558283977, 0.9877841416),00054800
PAIR( 0.1527971853, 0.9882575677),PAIR( 0.1497645347, 0.9887216920),00054900
PAIR( 0.1467304745, 0.9891765100),PAIR( 0.1436950332, 0.9896220175),00055000
PAIR( 0.1406582393, 0.9900582103),PAIR( 0.1376201216, 0.9904850843),00055100
PAIR( 0.1345807085, 0.9909026354),PAIR( 0.1315400287, 0.9913108598),00055200
PAIR( 0.1284981108, 0.9917097537),PAIR( 0.1254549834, 0.9920993131),00055300
PAIR( 0.1224106752, 0.9924795346),PAIR( 0.1193652148, 0.9928504145),00055400
PAIR( 0.1163186309, 0.9932119492),PAIR( 0.1132709522, 0.9935641355),00055500
PAIR( 0.1102222073, 0.9939069700),PAIR( 0.1071724250, 0.9942404495),00055600
PAIR( 0.1041216339, 0.9945645707),PAIR( 0.1010698628, 0.9948793308),00055700
PAIR( 0.0980171403, 0.9951847267),PAIR( 0.0949634953, 0.9954807555),00055800
PAIR( 0.0919089565, 0.9957674145),PAIR( 0.0888535526, 0.9960447009),00055900
PAIR( 0.0857973123, 0.9963126122),PAIR( 0.0827402645, 0.9965711458),00056000
PAIR( 0.0796824380, 0.9968202993),PAIR( 0.0766238614, 0.9970600703),00056100
PAIR( 0.0735645636, 0.9972904567),PAIR( 0.0705045734, 0.9975114561),00056200
PAIR( 0.0674439196, 0.9977230666),PAIR( 0.0643826309, 0.9979252862),00056300
PAIR( 0.0613207363, 0.9981181129),PAIR( 0.0582582645, 0.9983015449),00056400
PAIR( 0.0551952443, 0.9984755806),PAIR( 0.0521317047, 0.9986402182),00056500
PAIR( 0.0490676743, 0.9987954562),PAIR( 0.0460031821, 0.9989412932),00056600
PAIR( 0.0429382569, 0.9990777278),PAIR( 0.0398729276, 0.9992047586),00056700
PAIR( 0.0368072229, 0.9993223846),PAIR( 0.0337411719, 0.9994306046),00056800
```

```
PAIR( 0.0306748032, 0.9995294175),PAIR( 0.0276081458, 0.9996188225),00056900
PAIR( 0.0245412285, 0.9996988187),PAIR( 0.0214740803, 0.9997694053),00057000
PAIR( 0.0184067299, 0.9998305818),PAIR( 0.0153392063, 0.9998823475),00057100
PAIR( 0.0122715383, 0.9999247018),PAIR( 0.0092037548, 0.9999576446),00057200
PAIR( 0.0061358846, 0.9999811753),PAIR( 0.0030679568, 0.9999952938),00057300
 PAIR(1.0,0.0),PAIR(.9999988235,.0015339802),                        00057400
 PAIR(.9999952938,.0030679568),                                      00057500
PAIR( 0.9999894111, 0.0046019261),PAIR( 0.9999811753, 0.0061358846),00057600
PAIR( 0.9999705864, 0.0076698287),PAIR( 0.9999576446, 0.0092037548),00057700
PAIR( 0.9999423497, 0.0107376592),PAIR( 0.9999247018, 0.0122715383),00057800
PAIR( 0.9999047011, 0.0138053885),PAIR( 0.9998823475, 0.0153392063),00057900
PAIR( 0.9998576410, 0.0168729879),PAIR( 0.9998305818, 0.0184067299),00058000
PAIR( 0.9998011699, 0.0199404286),PAIR( 0.9997694053, 0.0214740803),00058100
PAIR( 0.9997352883, 0.0230076815),PAIR( 0.9996988187, 0.0245412285),00058200
PAIR( 0.9996599967, 0.0260747178),PAIR( 0.9996188225, 0.0276081458),00058300
PAIR( 0.9995752961, 0.0291415088),PAIR( 0.9995294175, 0.0306748032),00058400
PAIR( 0.9994811870, 0.0322080254),PAIR( 0.9994306046, 0.0337411719),00058500
PAIR( 0.9993776704, 0.0352742389),PAIR( 0.9993223846, 0.0368072229),00058600
PAIR( 0.9992647473, 0.0383401204),PAIR( 0.9992047586, 0.0398729276),00058700
PAIR( 0.9991424187, 0.0414056410),PAIR( 0.9990777278, 0.0429382569),00058800
PAIR( 0.9990106859, 0.0444707719),PAIR( 0.9989412932, 0.0460031821),00058900
PAIR( 0.9988695499, 0.0475354842),PAIR( 0.9987954562, 0.0490676743),00059000
PAIR( 0.9987190122, 0.0505997490),PAIR( 0.9986402182, 0.0521317047),00059100
PAIR( 0.9985590742, 0.0536635377),PAIR( 0.9984755806, 0.0551952443),00059200
PAIR( 0.9983897374, 0.0567268212),PAIR( 0.9983015449, 0.0582582645),00059300
PAIR( 0.9982110034, 0.0597895707),PAIR( 0.9981181129, 0.0613207363),00059400
PAIR( 0.9980228738, 0.0628517576),PAIR( 0.9979252862, 0.0643826309),00059500
PAIR( 0.9978253504, 0.0659133528),PAIR( 0.9977230666, 0.0674439196),00059600
PAIR( 0.9976184351, 0.0689743276),PAIR( 0.9975114561, 0.0705045734),00059700
PAIR( 0.9974021299, 0.0720346532),PAIR( 0.9972904567, 0.0735645636),00059800
PAIR( 0.9971764367, 0.0750943008),PAIR( 0.9970600703, 0.0766238614),00059900
PAIR( 0.9969413578, 0.0781532416),PAIR( 0.9968202993, 0.0796824380),00060000
PAIR( 0.9966968952, 0.0812114468),PAIR( 0.9965711458, 0.0827402645),00060100
PAIR( 0.9964430514, 0.0842688876),PAIR( 0.9963126122, 0.0857973123),00060200
PAIR( 0.9961798286, 0.0873255352),PAIR( 0.9960447009, 0.0888535526),00060300
PAIR( 0.9959072294, 0.0903813609),PAIR( 0.9957674145, 0.0919089565),00060400
PAIR( 0.9956252564, 0.0934363358),PAIR( 0.9954807555, 0.0949634953),00060500
PAIR( 0.9953339121, 0.0964904314),PAIR( 0.9951847267, 0.0980171403),00060600
PAIR( 0.9950331994, 0.0995436187),PAIR( 0.9948793308, 0.1010698628),00060700
PAIR( 0.9947231211, 0.1025958690),PAIR( 0.9945645707, 0.1041216339),00060800
PAIR( 0.9944036801, 0.1056471537),PAIR( 0.9942404495, 0.1071724250),00060900
PAIR( 0.9940748793, 0.1086974440),PAIR( 0.9939069700, 0.1102222073),00061000
PAIR( 0.9937367219, 0.1117467112),PAIR( 0.9935641355, 0.1132709522),00061100
PAIR( 0.9933892111, 0.1147949266),PAIR( 0.9932119492, 0.1163186309),00061200
PAIR( 0.9930323502, 0.1178420615),PAIR( 0.9928504145, 0.1193652148),00061300
PAIR( 0.9926661425, 0.1208880872),PAIR( 0.9924795346, 0.1224106752),00061400
PAIR( 0.9922905913, 0.1239329751),PAIR( 0.9920993131, 0.1254549834),00061500
PAIR( 0.9919057004, 0.1269766965),PAIR( 0.9917097537, 0.1284981108),00061600
PAIR( 0.9915114733, 0.1300192227),PAIR( 0.9913108598, 0.1315400287),00061700
PAIR( 0.9911079137, 0.1330605252),PAIR( 0.9909026354, 0.1345807085),00061800
PAIR( 0.9906950254, 0.1361005752),PAIR( 0.9904850843, 0.1376201216),00061900
PAIR( 0.9902728124, 0.1391393442),PAIR( 0.9900582103, 0.1406582393),00062000
PAIR( 0.9898412785, 0.1421768035),PAIR( 0.9896220175, 0.1436950332),00062100
PAIR( 0.9894004278, 0.1452129247),PAIR( 0.9891765100, 0.1467304745),00062200
PAIR( 0.9889502645, 0.1482476790),PAIR( 0.9887216920, 0.1497645347),00062300
PAIR( 0.9884907929, 0.1512810380),PAIR( 0.9882575677, 0.1527971853),00062400
PAIR( 0.9880220171, 0.1543129730),PAIR( 0.9877841416, 0.1558283977),00062500
```

```
PAIR( 0.9875439418, 0.1573434556),PAIR( 0.9873014182, 0.1588581433),00062600
PAIR( 0.9870565713, 0.1603724572),PAIR( 0.9868094018, 0.1618863938),00062700
PAIR( 0.9865599103, 0.1633999494),PAIR( 0.9863080972, 0.1649131205),00062800
PAIR( 0.9860539633, 0.1664259035),PAIR( 0.9857975092, 0.1679382950),00062900
PAIR( 0.9855387353, 0.1694502912),PAIR( 0.9852776424, 0.1709618888),00063000
PAIR( 0.9850142310, 0.1724730840),PAIR( 0.9847485018, 0.1739838734),00063100
PAIR( 0.9844804554, 0.1754942534),PAIR( 0.9842100924, 0.1770042204),00063200
PAIR( 0.9839374134, 0.1785137709),PAIR( 0.9836624192, 0.1800229014),00063300
PAIR( 0.9833851103, 0.1815316083),PAIR( 0.9831054874, 0.1830398880),00063400
PAIR( 0.9828235512, 0.1845477369),PAIR( 0.9825393023, 0.1860551517),00063500
PAIR( 0.9822527414, 0.1875621286),PAIR( 0.9819638691, 0.1890686641),00063600
PAIR( 0.9816726862, 0.1905747548),PAIR( 0.9813791933, 0.1920803970),00063700
PAIR( 0.9810833912, 0.1935855873),PAIR( 0.9807852804, 0.1950903220),00063800
PAIR( 0.9804848618, 0.1965945977),PAIR( 0.9801821360, 0.1980984107),00063900
PAIR( 0.9798771037, 0.1996017576),PAIR( 0.9795697657, 0.2011046348),00064000
PAIR( 0.9792601226, 0.2026070388),PAIR( 0.9789481753, 0.2041089661),00064100
PAIR( 0.9786339244, 0.2056104131),PAIR( 0.9783173707, 0.2071113762),00064200
PAIR( 0.9779985149, 0.2086118520),PAIR( 0.9776773578, 0.2101118369),00064300
PAIR( 0.9773539001, 0.2116113274),PAIR( 0.9770281427, 0.2131103199),00064400
PAIR( 0.9767000861, 0.2146088110),PAIR( 0.9763697313, 0.2161067971),00064500
PAIR( 0.9760370790, 0.2176042746),PAIR( 0.9757021300, 0.2191012402),00064600
PAIR( 0.9753648851, 0.2205976901),PAIR( 0.9750253451, 0.2220936210),00064700
PAIR( 0.9746835107, 0.2235890292),PAIR( 0.9743393828, 0.2250839114),00064800
PAIR( 0.9739929622, 0.2265782638),PAIR( 0.9736442497, 0.2280720832),00064900
PAIR( 0.9732932461, 0.2295653658),PAIR( 0.9729399522, 0.2310581083),00065000
PAIR( 0.9725843689, 0.2325503070),PAIR( 0.9722264971, 0.2340419586),00065100
PAIR( 0.9718663375, 0.2355330594),PAIR( 0.9715038910, 0.2370236060),00065200
PAIR( 0.9711391585, 0.2385135948),PAIR( 0.9707721407, 0.2400030224),00065300
PAIR( 0.9704028387, 0.2414918853),PAIR( 0.9700312532, 0.2429801799),00065400
PAIR( 0.9696573851, 0.2444679027),PAIR( 0.9692812354, 0.2459550503),00065500
PAIR( 0.9689028048, 0.2474416192),PAIR( 0.9685220943, 0.2489276057),00065600
PAIR( 0.9681391047, 0.2504130066),PAIR( 0.9677538371, 0.2518978182),00065700
PAIR( 0.9673662922, 0.2533820370),PAIR( 0.9669764710, 0.2548656596),00065800
PAIR( 0.9665843745, 0.2563486825),PAIR( 0.9661900034, 0.2578311022),00065900
PAIR( 0.9657933589, 0.2593129151),PAIR( 0.9653944417, 0.2607941179),00066000
PAIR( 0.9649932529, 0.2622747070),PAIR( 0.9645897933, 0.2637546790),00066100
PAIR( 0.9641840640, 0.2652340303),PAIR( 0.9637760658, 0.2667127575),00066200
PAIR( 0.9633657998, 0.2681908571),PAIR( 0.9629532669, 0.2696683256),00066300
PAIR( 0.9625384680, 0.2711451595),PAIR( 0.9621214043, 0.2726213555),00066400
PAIR( 0.9617020765, 0.2740969099),PAIR( 0.9612804858, 0.2755718193),00066500
PAIR( 0.9608566331, 0.2770460803),PAIR( 0.9604305194, 0.2785196894),00066600
PAIR( 0.9600021457, 0.2799926431),PAIR( 0.9595715131, 0.2814649379),00066700
PAIR( 0.9591386225, 0.2829365705),PAIR( 0.9587034749, 0.2844075372),00066800
PAIR( 0.9582660714, 0.2858778347),PAIR( 0.9578264130, 0.2873474595),00066900
PAIR( 0.9573845008, 0.2888164082),PAIR( 0.9569403357, 0.2902846773),00067000
PAIR( 0.9564939189, 0.2917522632),PAIR( 0.9560452513, 0.2932191627),00067100
PAIR( 0.9555943341, 0.2946853722),PAIR( 0.9551411683, 0.2961508882),00067200
PAIR( 0.9546857549, 0.2976157074),PAIR( 0.9542280951, 0.2990798263),00067300
PAIR( 0.9537681899, 0.3005432414),PAIR( 0.9533060404, 0.3020059493),00067400
PAIR( 0.9528416476, 0.3034679466),PAIR( 0.9523750127, 0.3049292297),00067500
PAIR( 0.9519061368, 0.3063897954),PAIR( 0.9514350210, 0.3078496400),00067600
PAIR( 0.9509616663, 0.3093087603),PAIR( 0.9504860739, 0.3107671527),00067700
PAIR( 0.9500082450, 0.3122248139),PAIR( 0.9495281806, 0.3136817404),00067800
PAIR( 0.9490458819, 0.3151379288),PAIR( 0.9485613499, 0.3165933756),00067900
PAIR( 0.9480745859, 0.3180480774),PAIR( 0.9475855910, 0.3195020308),00068000
PAIR( 0.9470943664, 0.3209552324),PAIR( 0.9466009131, 0.3224076788),00068100
PAIR( 0.9461052324, 0.3238593665),PAIR( 0.9456073254, 0.3253102922),00068200
```

```
PAIR( 0.9451071933, 0.3267604523),PAIR( 0.9446048373, 0.3282098436),00068300
PAIR( 0.9441002585, 0.3296584625),PAIR( 0.9435934582, 0.3311063058),00068400
PAIR( 0.9430844375, 0.3325533699),PAIR( 0.9425731976, 0.3339996514),00068500
PAIR( 0.9420597398, 0.3354451471),PAIR( 0.9415440652, 0.3368898534),00068600
PAIR( 0.9410261751, 0.3383337670),PAIR( 0.9405060706, 0.3397768844),00068700
PAIR( 0.9399837530, 0.3412192023),PAIR( 0.9394592236, 0.3426607173),00068800
PAIR( 0.9389324835, 0.3441014260),PAIR( 0.9384035341, 0.3455413250),00068900
PAIR( 0.9378723764, 0.3469804108),PAIR( 0.9373390119, 0.3484186802),00069000
PAIR( 0.9368034417, 0.3498561298),PAIR( 0.9362656672, 0.3512927561),00069100
PAIR( 0.9357256895, 0.3527285558),PAIR( 0.9351835099, 0.3541635254),00069200
PAIR( 0.9346391298, 0.3555976617),PAIR( 0.9340925504, 0.3570309612),00069300
PAIR( 0.9335437730, 0.3584634206),PAIR( 0.9329927988, 0.3598950365),00069400
PAIR( 0.9324394293, 0.3613258056),PAIR( 0.9318842656, 0.3627557244),00069500
PAIR( 0.9313267091, 0.3641847896),PAIR( 0.9307669611, 0.3656129978),00069600
PAIR( 0.9302050229, 0.3670403457),PAIR( 0.9296408958, 0.3684668300),00069700
PAIR( 0.9290745813, 0.3698924471),PAIR( 0.9285060805, 0.3713171940),00069800
PAIR( 0.9279353948, 0.3727410670),PAIR( 0.9273625256, 0.3741640630),00069900
PAIR( 0.9267874743, 0.3755861785),PAIR( 0.9262102421, 0.3770074102),00070000
PAIR( 0.9256308305, 0.3784277548),PAIR( 0.9250492408, 0.3798472089),00070100
PAIR( 0.9244654743, 0.3812657692),PAIR( 0.9238795325, 0.3826834324),00070200
PAIR( 0.9232914167, 0.3841001950),PAIR( 0.9227011283, 0.3855160538),00070300
PAIR( 0.9221086687, 0.3869310055),PAIR( 0.9215140393, 0.3883450467),00070400
PAIR( 0.9209172415, 0.3897581741),PAIR( 0.9203182767, 0.3911703843),00070500
PAIR( 0.9197171463, 0.3925816741),PAIR( 0.9191138517, 0.3939920401),00070600
PAIR( 0.9185083943, 0.3954014789),PAIR( 0.9179007756, 0.3968099874),00070700
PAIR( 0.9172909970, 0.3982175622),PAIR( 0.9166790599, 0.3996241998),00070800
PAIR( 0.9160649658, 0.4010298972),PAIR( 0.9154487161, 0.4024346509),00070900
PAIR( 0.9148303122, 0.4038384576),PAIR( 0.9142097557, 0.4052413140),00071000
PAIR( 0.9135870479, 0.4066432169),PAIR( 0.9129621904, 0.4080441629),00071100
PAIR( 0.9123351846, 0.4094441487),PAIR( 0.9117060320, 0.4108431711),00071200
PAIR( 0.9110747341, 0.4122412267),PAIR( 0.9104412923, 0.4136383122),00071300
PAIR( 0.9098057081, 0.4150344245),PAIR( 0.9091679831, 0.4164295601),00071400
PAIR( 0.9085281187, 0.4178237158),PAIR( 0.9078861165, 0.4192168884),00071500
PAIR( 0.9072419779, 0.4206090744),PAIR( 0.9065957045, 0.4220002708),00071600
PAIR( 0.9059472978, 0.4233904741),PAIR( 0.9052967593, 0.4247796812),00071700
PAIR( 0.9046440906, 0.4261678887),PAIR( 0.9039892931, 0.4275550934),00071800
PAIR( 0.9033323685, 0.4289412921),PAIR( 0.9026733182, 0.4303264813),00071900
PAIR( 0.9020121439, 0.4317106580),PAIR( 0.9013488470, 0.4330938189),00072000
PAIR( 0.9006834292, 0.4344759606),PAIR( 0.9000158920, 0.4358570799),00072100
PAIR( 0.8993462370, 0.4372371737),PAIR( 0.8986744657, 0.4386162385),00072200
PAIR( 0.8980005797, 0.4399942713),PAIR( 0.8973245807, 0.4413712687),00072300
PAIR( 0.8966464702, 0.4427472276),PAIR( 0.8959662498, 0.4441221446),00072400
PAIR( 0.8952839210, 0.4454960165),PAIR( 0.8945994856, 0.4468688402),00072500
PAIR( 0.8939129451, 0.4482406123),PAIR( 0.8932243012, 0.4496113297),00072600
PAIR( 0.8925335554, 0.4509809890),PAIR( 0.8918407094, 0.4523495872),00072700
PAIR( 0.8911457648, 0.4537171210),PAIR( 0.8904487232, 0.4550835871),00072800
PAIR( 0.8897495864, 0.4564489824),PAIR( 0.8890483558, 0.4578133036),00072900
PAIR( 0.8883450333, 0.4591765475),PAIR( 0.8876396204, 0.4605387110),00073000
PAIR( 0.8869321188, 0.4618997907),PAIR( 0.8862225301, 0.4632597836),00073100
PAIR( 0.8855108561, 0.4646186863),PAIR( 0.8847970984, 0.4659764958),00073200
PAIR( 0.8840812587, 0.4673332087),PAIR( 0.8833633387, 0.4686888220),00073300
PAIR( 0.8826433400, 0.4700433325),PAIR( 0.8819212644, 0.4713967368),00073400
PAIR( 0.8811971135, 0.4727490320),PAIR( 0.8804708891, 0.4741002146),00073500
PAIR( 0.8797425928, 0.4754502817),PAIR( 0.8790122264, 0.4767992301),00073600
PAIR( 0.8782797917, 0.4781470564),PAIR( 0.8775452902, 0.4794937577),00073700
PAIR( 0.8768087238, 0.4808393306),PAIR( 0.8760700942, 0.4821837721),00073800
PAIR( 0.8753294031, 0.4835270789),PAIR( 0.8745866523, 0.4848692480),00073900
```

```
PAIR( 0.8738418435, 0.4862102761),PAIR( 0.8730949784, 0.4875501601),00074000
PAIR( 0.8723460589, 0.4888888969),PAIR( 0.8715950867, 0.4902264833),00074100
PAIR( 0.8708420635, 0.4915629161),PAIR( 0.8700869911, 0.4928981922),00074200
PAIR( 0.8693298714, 0.4942323085),PAIR( 0.8685707060, 0.4955652618),00074300
PAIR( 0.8678094968, 0.4968970490),PAIR( 0.8670462455, 0.4982276670),00074400
PAIR( 0.8662809540, 0.4995571125),PAIR( 0.8655136241, 0.5008853826),00074500
PAIR( 0.8647442575, 0.5022124740),PAIR( 0.8639728561, 0.5035383837),00074600
PAIR( 0.8631994217, 0.5048631085),PAIR( 0.8624239561, 0.5061866453),00074700
PAIR( 0.8616464611, 0.5075089911),PAIR( 0.8608669386, 0.5088301425),00074800
PAIR( 0.8600853904, 0.5101500967),PAIR( 0.8593018184, 0.5114688504),00074900
PAIR( 0.8585162243, 0.5127864006),PAIR( 0.8577286100, 0.5141027442),00075000
PAIR( 0.8569389774, 0.5154178780),PAIR( 0.8561473284, 0.5167317990),00075100
PAIR( 0.8553536647, 0.5180445041),PAIR( 0.8545579884, 0.5193559902),00075200
PAIR( 0.8537603011, 0.5206662541),PAIR( 0.8529606049, 0.5219752929),00075300
PAIR( 0.8521589016, 0.5232831035),PAIR( 0.8513551931, 0.5245896827),00075400
PAIR( 0.8505494813, 0.5258950275),PAIR( 0.8497417680, 0.5271991348),00075500
PAIR( 0.8489320552, 0.5285020015),PAIR( 0.8481203448, 0.5298036247),00075600
PAIR( 0.8473066387, 0.5311040012),PAIR( 0.8464909388, 0.5324031279),00075700
PAIR( 0.8456732470, 0.5337010018),PAIR( 0.8448535653, 0.5349976199),00075800
PAIR( 0.8440318955, 0.5362929791),PAIR( 0.8432082396, 0.5375870763),00075900
PAIR( 0.8423825996, 0.5388799085),PAIR( 0.8415549774, 0.5401714727),00076000
PAIR( 0.8407253750, 0.5414617659),PAIR( 0.8398937942, 0.5427507849),00076100
PAIR( 0.8390602371, 0.5440385267),PAIR( 0.8382247056, 0.5453249884),00076200
PAIR( 0.8373872016, 0.5466101669),PAIR( 0.8365477272, 0.5478940592),00076300
PAIR( 0.8357062844, 0.5491766622),PAIR( 0.8348628750, 0.5504579729),00076400
PAIR( 0.8340175011, 0.5517379884),PAIR( 0.8331701647, 0.5530167056),00076500
PAIR( 0.8323208678, 0.5542941215),PAIR( 0.8314696123, 0.5555702330),00076600
PAIR( 0.8306164003, 0.5568450373),PAIR( 0.8297612338, 0.5581185312),00076700
PAIR( 0.8289041148, 0.5593907119),PAIR( 0.8280450453, 0.5606615762),00076800
PAIR( 0.8271840273, 0.5619311212),PAIR( 0.8263210628, 0.5631993440),00076900
PAIR( 0.8254561540, 0.5644662415),PAIR( 0.8245893028, 0.5657318108),00077000
PAIR( 0.8237205112, 0.5669960488),PAIR( 0.8228497814, 0.5682589527),00077100
PAIR( 0.8219771153, 0.5695205193),PAIR( 0.8211025150, 0.5707807459),00077200
PAIR( 0.8202259826, 0.5720396293),PAIR( 0.8193475201, 0.5732971667),00077300
PAIR( 0.8184671296, 0.5745533550),PAIR( 0.8175848132, 0.5758081914),00077400
PAIR( 0.8167005729, 0.5770616729),PAIR( 0.8158144108, 0.5783137964),00077500
PAIR( 0.8149263291, 0.5795645591),PAIR( 0.8140363297, 0.5808139581),00077600
PAIR( 0.8131444148, 0.5820619903),PAIR( 0.8122505866, 0.5833086529),00077700
PAIR( 0.8113548470, 0.5845539430),PAIR( 0.8104571983, 0.5857978575),00077800
PAIR( 0.8095576424, 0.5870403935),PAIR( 0.8086561816, 0.5882815482),00077900
PAIR( 0.8077528179, 0.5895213186),PAIR( 0.8068475535, 0.5907597019),00078000
PAIR( 0.8059403906, 0.5919966950),PAIR( 0.8050313311, 0.5932322950),00078100
PAIR( 0.8041203774, 0.5944664992),PAIR( 0.8032075315, 0.5956993045),00078200
PAIR( 0.8022927955, 0.5969307081),PAIR( 0.8013761717, 0.5981607070),00078300
PAIR( 0.8004576622, 0.5993892984),PAIR( 0.7995372691, 0.6006164794),00078400
PAIR( 0.7986149946, 0.6018422471),PAIR( 0.7976908409, 0.6030665985),00078500
PAIR( 0.7967648102, 0.6042895309),PAIR( 0.7958369046, 0.6055110414),00078600
PAIR( 0.7949071263, 0.6067311270),PAIR( 0.7939754776, 0.6079497850),00078700
PAIR( 0.7930419605, 0.6091670123),PAIR( 0.7921065773, 0.6103828063),00078800
PAIR( 0.7911693302, 0.6115971639),PAIR( 0.7902302214, 0.6128100824),00078900
PAIR( 0.7892892532, 0.6140215589),PAIR( 0.7883464276, 0.6152315906),00079000
PAIR( 0.7874017470, 0.6164401745),PAIR( 0.7864552136, 0.6176473079),00079100
PAIR( 0.7855068296, 0.6188529880),PAIR( 0.7845565972, 0.6200572118),00079200
PAIR( 0.7836045186, 0.6212599765),PAIR( 0.7826505962, 0.6224612794),00079300
PAIR( 0.7816948321, 0.6236611175),PAIR( 0.7807372286, 0.6248594881),00079400
PAIR( 0.7797777879, 0.6260563884),PAIR( 0.7788165124, 0.6272518155),00079500
PAIR( 0.7778534042, 0.6284457666),PAIR( 0.7768884657, 0.6296382389),00079600
```

```
PAIR( 0.7759216990, 0.6308292296),PAIR( 0.7749531066, 0.6320187359),00079700
PAIR( 0.7739826906, 0.6332067550),PAIR( 0.7730104534, 0.6343932842),00079800
PAIR( 0.7720363971, 0.6355783205),PAIR( 0.7710605243, 0.6367618612),00079900
PAIR( 0.7700828370, 0.6379439036),PAIR( 0.7691033376, 0.6391244449),00080000
PAIR( 0.7681220285, 0.6403034822),PAIR( 0.7671389119, 0.6414810128),00080100
PAIR( 0.7661539902, 0.6426570340),PAIR( 0.7651672656, 0.6438315429),00080200
PAIR( 0.7641787405, 0.6450045368),PAIR( 0.7631884173, 0.6461760130),00080300
PAIR( 0.7621962981, 0.6473459686),PAIR( 0.7612023855, 0.6485144010),00080400
PAIR( 0.7602066816, 0.6496813074),PAIR( 0.7592091890, 0.6508466850),00080500
PAIR( 0.7582099098, 0.6520105311),PAIR( 0.7572088465, 0.6531728430),00080600
PAIR( 0.7562060014, 0.6543336178),PAIR( 0.7552013769, 0.6554928530),00080700
PAIR( 0.7541949753, 0.6566505457),PAIR( 0.7531867990, 0.6578066933),00080800
PAIR( 0.7521768504, 0.6589612930),PAIR( 0.7511651319, 0.6601143421),00080900
PAIR( 0.7501516458, 0.6612658378),PAIR( 0.7491363945, 0.6624157776),00081000
PAIR( 0.7481193805, 0.6635641586),PAIR( 0.7471006060, 0.6647109782),00081100
PAIR( 0.7460800735, 0.6658562337),PAIR( 0.7450577854, 0.6669999223),00081200
PAIR( 0.7440337442, 0.6681420414),PAIR( 0.7430079521, 0.6692825883),00081300
PAIR( 0.7419804117, 0.6704215604),PAIR( 0.7409511254, 0.6715589548),00081400
PAIR( 0.7399200955, 0.6726947691),PAIR( 0.7388873245, 0.6738290004),00081500
PAIR( 0.7378528148, 0.6749616461),PAIR( 0.7368165689, 0.6760927036),00081600
PAIR( 0.7357785892, 0.6772221701),PAIR( 0.7347388781, 0.6783500431),00081700
PAIR( 0.7336974381, 0.6794763199),PAIR( 0.7326542717, 0.6806009978),00081800
PAIR( 0.7316093812, 0.6817240742),PAIR( 0.7305627692, 0.6828455464),00081900
PAIR( 0.7295144381, 0.6839654118),PAIR( 0.7284643904, 0.6850836678),00082000
PAIR( 0.7274126286, 0.6862003117),PAIR( 0.7263591551, 0.6873153409),00082100
PAIR( 0.7253039724, 0.6884287528),PAIR( 0.7242470830, 0.6895405447),00082200
PAIR( 0.7231884893, 0.6906507141),PAIR( 0.7221281939, 0.6917592584),00082300
PAIR( 0.7210661993, 0.6928661748),PAIR( 0.7200025080, 0.6939714609),00082400
PAIR( 0.7189371224, 0.6950751140),PAIR( 0.7178700451, 0.6961771315),00082500
PAIR( 0.7168012785, 0.6972775108),PAIR( 0.7157308253, 0.6983762494),00082600
PAIR( 0.7146586879, 0.6994733446),PAIR( 0.7135848688, 0.7005687939),00082700
PAIR( 0.7125093706, 0.7016625947),PAIR( 0.7114321957, 0.7027547445),00082800
PAIR( 0.7103533469, 0.7038452405),PAIR( 0.7092728264, 0.7049340804),00082900
PAIR( 0.7081906370, 0.7060212614),PAIR( 0.7071067812, 0.7071067812),00083000
PAIR( 0.7060212614, 0.7081906370),PAIR( 0.7049340804, 0.7092728264),00083100
PAIR( 0.7038452405, 0.7103533469),PAIR( 0.7027547445, 0.7114321957),00083200
PAIR( 0.7016625947, 0.7125093706),PAIR( 0.7005687939, 0.7135848688),00083300
PAIR( 0.6994733446, 0.7146586879),PAIR( 0.6983762494, 0.7157308253),00083400
PAIR( 0.6972775108, 0.7168012785),PAIR( 0.6961771315, 0.7178700451),00083500
PAIR( 0.6950751140, 0.7189371224),PAIR( 0.6939714609, 0.7200025080),00083600
PAIR( 0.6928661748, 0.7210661993),PAIR( 0.6917592584, 0.7221281939),00083700
PAIR( 0.6906507141, 0.7231884893),PAIR( 0.6895405447, 0.7242470830),00083800
PAIR( 0.6884287528, 0.7253039724),PAIR( 0.6873153409, 0.7263591551),00083900
PAIR( 0.6862003117, 0.7274126286),PAIR( 0.6850836678, 0.7284643904),00084000
PAIR( 0.6839654118, 0.7295144381),PAIR( 0.6828455464, 0.7305627692),00084100
PAIR( 0.6817240742, 0.7316093812),PAIR( 0.6806009978, 0.7326542717),00084200
PAIR( 0.6794763199, 0.7336974381),PAIR( 0.6783500431, 0.7347388781),00084300
PAIR( 0.6772221701, 0.7357785892),PAIR( 0.6760927036, 0.7368165689),00084400
PAIR( 0.6749616461, 0.7378528148),PAIR( 0.6738290004, 0.7388873245),00084500
PAIR( 0.6726947691, 0.7399200955),PAIR( 0.6715589548, 0.7409511254),00084600
PAIR( 0.6704215604, 0.7419804117),PAIR( 0.6692825883, 0.7430079521),00084700
PAIR( 0.6681420414, 0.7440337442),PAIR( 0.6669999223, 0.7450577854),00084800
PAIR( 0.6658562337, 0.7460800735),PAIR( 0.6647109782, 0.7471006060),00084900
PAIR( 0.6635641586, 0.7481193805),PAIR( 0.6624157776, 0.7491363945),00085000
PAIR( 0.6612658378, 0.7501516458),PAIR( 0.6601143421, 0.7511651319),00085100
PAIR( 0.6589612930, 0.7521768504),PAIR( 0.6578066933, 0.7531867990),00085200
PAIR( 0.6566505457, 0.7541949753),PAIR( 0.6554928530, 0.7552013769),00085300
```

```
PAIR( 0.6543336178, 0.7562060014),PAIR( 0.6531728430, 0.7572088465),00085400
PAIR( 0.6520105311, 0.7582099098),PAIR( 0.6508466850, 0.7592091890),00085500
PAIR( 0.6496813074, 0.7602066816),PAIR( 0.6485144010, 0.7612023855),00085600
PAIR( 0.6473459686, 0.7621962981),PAIR( 0.6461760130, 0.7631884173),00085700
PAIR( 0.6450045368, 0.7641787405),PAIR( 0.6438315429, 0.7651672656),00085800
PAIR( 0.6426570340, 0.7661539902),PAIR( 0.6414810128, 0.7671389119),00085900
PAIR( 0.6403034822, 0.7681220285),PAIR( 0.6391244449, 0.7691033376),00086000
PAIR( 0.6379439036, 0.7700828370),PAIR( 0.6367618612, 0.7710605243),00086100
PAIR( 0.6355783205, 0.7720363971),PAIR( 0.6343932842, 0.7730104534),00086200
PAIR( 0.6332067550, 0.7739826906),PAIR( 0.6320187359, 0.7749531066),00086300
PAIR( 0.6308292296, 0.7759216990),PAIR( 0.6296382389, 0.7768884657),00086400
PAIR( 0.6284457666, 0.7778534042),PAIR( 0.6272518155, 0.7788165124),00086500
PAIR( 0.6260563884, 0.7797777879),PAIR( 0.6248594881, 0.7807372286),00086600
PAIR( 0.6236611175, 0.7816948321),PAIR( 0.6224612794, 0.7826505962),00086700
PAIR( 0.6212599765, 0.7836045186),PAIR( 0.6200572118, 0.7845565972),00086800
PAIR( 0.6188529880, 0.7855068296),PAIR( 0.6176473079, 0.7864552136),00086900
PAIR( 0.6164401745, 0.7874017470),PAIR( 0.6152315906, 0.7883464276),00087000
PAIR( 0.6140215589, 0.7892892532),PAIR( 0.6128100824, 0.7902302214),00087100
PAIR( 0.6115971639, 0.7911693302),PAIR( 0.6103828063, 0.7921065773),00087200
PAIR( 0.6091670123, 0.7930419605),PAIR( 0.6079497850, 0.7939754776),00087300
PAIR( 0.6067311270, 0.7949071263),PAIR( 0.6055110414, 0.7958369046),00087400
PAIR( 0.6042895309, 0.7967648102),PAIR( 0.6030665985, 0.7976908409),00087500
PAIR( 0.6018422471, 0.7986149946),PAIR( 0.6006164794, 0.7995372691),00087600
PAIR( 0.5993892984, 0.8004576622),PAIR( 0.5981607070, 0.8013761717),00087700
PAIR( 0.5969307081, 0.8022927955),PAIR( 0.5956993045, 0.8032075315),00087800
PAIR( 0.5944664992, 0.8041203774),PAIR( 0.5932322950, 0.8050313311),00087900
PAIR( 0.5919966950, 0.8059403906),PAIR( 0.5907597019, 0.8068475535),00088000
PAIR( 0.5895213186, 0.8077528179),PAIR( 0.5882815482, 0.8086561816),00088100
PAIR( 0.5870403935, 0.8095576424),PAIR( 0.5857978575, 0.8104571983),00088200
PAIR( 0.5845539430, 0.8113548470),PAIR( 0.5833086529, 0.8122505866),00088300
PAIR( 0.5820619903, 0.8131444148),PAIR( 0.5808139581, 0.8140363297),00088400
PAIR( 0.5795645591, 0.8149263291),PAIR( 0.5783137964, 0.8158144108),00088500
PAIR( 0.5770616729, 0.8167005729),PAIR( 0.5758081914, 0.8175848132),00088600
PAIR( 0.5745533550, 0.8184671296),PAIR( 0.5732971667, 0.8193475201),00088700
PAIR( 0.5720396293, 0.8202259826),PAIR( 0.5707807459, 0.8211025150),00088800
PAIR( 0.5695205193, 0.8219771153),PAIR( 0.5682589527, 0.8228497814),00088900
PAIR( 0.5669960488, 0.8237205112),PAIR( 0.5657318108, 0.8245893028),00089000
PAIR( 0.5644662415, 0.8254561540),PAIR( 0.5631993440, 0.8263210628),00089100
PAIR( 0.5619311212, 0.8271840273),PAIR( 0.5606615762, 0.8280450453),00089200
PAIR( 0.5593907119, 0.8289041148),PAIR( 0.5581185312, 0.8297612338),00089300
PAIR( 0.5568450373, 0.8306164003),PAIR( 0.5555702330, 0.8314696123),00089400
PAIR( 0.5542941215, 0.8323208678),PAIR( 0.5530167056, 0.8331701647),00089500
PAIR( 0.5517379884, 0.8340175011),PAIR( 0.5504579729, 0.8348628750),00089600
PAIR( 0.5491766622, 0.8357062844),PAIR( 0.5478940592, 0.8365477272),00089700
PAIR( 0.5466101669, 0.8373872016),PAIR( 0.5453249884, 0.8382247056),00089800
PAIR( 0.5440385267, 0.8390602371),PAIR( 0.5427507849, 0.8398937942),00089900
PAIR( 0.5414617659, 0.8407253750),PAIR( 0.5401714727, 0.8415549774),00090000
PAIR( 0.5388799085, 0.8423825996),PAIR( 0.5375870763, 0.8432082396),00090100
PAIR( 0.5362929791, 0.8440318955),PAIR( 0.5349976199, 0.8448535653),00090200
PAIR( 0.5337010018, 0.8456732470),PAIR( 0.5324031279, 0.8464909388),00090300
PAIR( 0.5311040012, 0.8473066387),PAIR( 0.5298036247, 0.8481203448),00090400
PAIR( 0.5285020015, 0.8489320552),PAIR( 0.5271991348, 0.8497417680),00090500
PAIR( 0.5258950275, 0.8505494813),PAIR( 0.5245896827, 0.8513551931),00090600
PAIR( 0.5232831035, 0.8521589016),PAIR( 0.5219752929, 0.8529606049),00090700
PAIR( 0.5206662541, 0.8537603011),PAIR( 0.5193559902, 0.8545579884),00090800
PAIR( 0.5180445041, 0.8553536647),PAIR( 0.5167317990, 0.8561473284),00090900
PAIR( 0.5154178780, 0.8569389774),PAIR( 0.5141027442, 0.8577286100),00091000
```

```
PAIR( 0.5127864006, 0.8585162243),PAIR( 0.5114688504, 0.8593018184),00091100
PAIR( 0.5101500967, 0.8600853904),PAIR( 0.5088301425, 0.8608669386),00091200
PAIR( 0.5075089911, 0.8616464611),PAIR( 0.5061866453, 0.8624239561),00091300
PAIR( 0.5048631085, 0.8631994217),PAIR( 0.5035383837, 0.8639728561),00091400
PAIR( 0.5022124740, 0.8647442575),PAIR( 0.5008853826, 0.8655136241),00091500
PAIR( 0.4995571125, 0.8662809540),PAIR( 0.4982276670, 0.8670462455),00091600
PAIR( 0.4968970490, 0.8678094968),PAIR( 0.4955652618, 0.8685707060),00091700
PAIR( 0.4942323085, 0.8693298714),PAIR( 0.4928981922, 0.8700869911),00091800
PAIR( 0.4915629161, 0.8708420635),PAIR( 0.4902264833, 0.8715950867),00091900
PAIR( 0.4888888969, 0.8723460589),PAIR( 0.4875501601, 0.8730949784),00092000
PAIR( 0.4862102761, 0.8738418435),PAIR( 0.4848692480, 0.8745866523),00092100
PAIR( 0.4835270789, 0.8753294031),PAIR( 0.4821837721, 0.8760700942),00092200
PAIR( 0.4808393306, 0.8768087238),PAIR( 0.4794937577, 0.8775452902),00092300
PAIR( 0.4781470564, 0.8782797917),PAIR( 0.4767992301, 0.8790122264),00092400
PAIR( 0.4754502817, 0.8797425928),PAIR( 0.4741002146, 0.8804708891),00092500
PAIR( 0.4727490320, 0.8811971135),PAIR( 0.4713967368, 0.8819212644),00092600
PAIR( 0.4700433325, 0.8826433400),PAIR( 0.4686888220, 0.8833633387),00092700
PAIR( 0.4673332087, 0.8840812587),PAIR( 0.4659764958, 0.8847970984),00092800
PAIR( 0.4646186863, 0.8855108561),PAIR( 0.4632597836, 0.8862225301),00092900
PAIR( 0.4618997907, 0.8869321188),PAIR( 0.4605387110, 0.8876396204),00093000
PAIR( 0.4591765475, 0.8883450333),PAIR( 0.4578133036, 0.8890483558),00093100
PAIR( 0.4564489824, 0.8897495864),PAIR( 0.4550835871, 0.8904487232),00093200
PAIR( 0.4537171210, 0.8911457648),PAIR( 0.4523495872, 0.8918407094),00093300
PAIR( 0.4509809890, 0.8925335554),PAIR( 0.4496113297, 0.8932243012),00093400
PAIR( 0.4482406123, 0.8939129451),PAIR( 0.4468688402, 0.8945994856),00093500
PAIR( 0.4454960165, 0.8952839210),PAIR( 0.4441221446, 0.8959662498),00093600
PAIR( 0.4427472276, 0.8966464702),PAIR( 0.4413712687, 0.8973245807),00093700
PAIR( 0.4399942713, 0.8980005797),PAIR( 0.4386162385, 0.8986744657),00093800
PAIR( 0.4372371737, 0.8993462370),PAIR( 0.4358570799, 0.9000158920),00093900
PAIR( 0.4344759606, 0.9006834292),PAIR( 0.4330938189, 0.9013488470),00094000
PAIR( 0.4317106580, 0.9020121439),PAIR( 0.4303264813, 0.9026733182),00094100
PAIR( 0.4289412921, 0.9033323685),PAIR( 0.4275550934, 0.9039892931),00094200
PAIR( 0.4261678887, 0.9046440906),PAIR( 0.4247796812, 0.9052967593),00094300
PAIR( 0.4233904741, 0.9059472978),PAIR( 0.4220002708, 0.9065957045),00094400
PAIR( 0.4206090744, 0.9072419779),PAIR( 0.4192168884, 0.9078861165),00094500
PAIR( 0.4178237158, 0.9085281187),PAIR( 0.4164295601, 0.9091679831),00094600
PAIR( 0.4150344245, 0.9098057081),PAIR( 0.4136383122, 0.9104412923),00094700
PAIR( 0.4122412267, 0.9110747341),PAIR( 0.4108431711, 0.9117060320),00094800
PAIR( 0.4094441487, 0.9123351846),PAIR( 0.4080441629, 0.9129621904),00094900
PAIR( 0.4066432169, 0.9135870479),PAIR( 0.4052413140, 0.9142097557),00095000
PAIR( 0.4038384576, 0.9148303122),PAIR( 0.4024346509, 0.9154487161),00095100
PAIR( 0.4010298972, 0.9160649658),PAIR( 0.3996241998, 0.9166790599),00095200
PAIR( 0.3982175622, 0.9172909970),PAIR( 0.3968099874, 0.9179007756),00095300
PAIR( 0.3954014789, 0.9185083943),PAIR( 0.3939920401, 0.9191138517),00095400
PAIR( 0.3925816741, 0.9197171463),PAIR( 0.3911703843, 0.9203182767),00095500
PAIR( 0.3897581741, 0.9209172415),PAIR( 0.3883450467, 0.9215140393),00095600
PAIR( 0.3869310055, 0.9221086687),PAIR( 0.3855160538, 0.9227011283),00095700
PAIR( 0.3841001950, 0.9232914167),PAIR( 0.3826834324, 0.9238795325),00095800
PAIR( 0.3812657692, 0.9244654743),PAIR( 0.3798472089, 0.9250492408),00095900
PAIR( 0.3784277548, 0.9256308305),PAIR( 0.3770074102, 0.9262102421),00096000
PAIR( 0.3755861785, 0.9267874743),PAIR( 0.3741640630, 0.9273625256),00096100
PAIR( 0.3727410670, 0.9279353948),PAIR( 0.3713171940, 0.9285060805),00096200
PAIR( 0.3698924471, 0.9290745813),PAIR( 0.3684668300, 0.9296408958),00096300
PAIR( 0.3670403457, 0.9302050229),PAIR( 0.3656129978, 0.9307669611),00096400
PAIR( 0.3641847896, 0.9313267091),PAIR( 0.3627557244, 0.9318842656),00096500
PAIR( 0.3613258056, 0.9324396293),PAIR( 0.3598950365, 0.9329927988),00096600
PAIR( 0.3584634206, 0.9335437730),PAIR( 0.3570309612, 0.9340925504),00096700
```

```
PAIR( 0.3555976617, 0.9346391298),PAIR( 0.3541635254, 0.9351835099),00096800
PAIR( 0.3527285558, 0.9357256895),PAIR( 0.3512927561, 0.9362656672),00096900
PAIR( 0.3498561298, 0.9368034417),PAIR( 0.3484186802, 0.9373390119),00097000
PAIR( 0.3469804108, 0.9378723764),PAIR( 0.3455413250, 0.9384035341),00097100
PAIR( 0.3441014260, 0.9389324835),PAIR( 0.3426607173, 0.9394592236),00097200
PAIR( 0.3412192023, 0.9399837530),PAIR( 0.3397768844, 0.9405060706),00097300
PAIR( 0.3383337670, 0.9410261751),PAIR( 0.3368898534, 0.9415440652),00097400
PAIR( 0.3354451471, 0.9420597398),PAIR( 0.3339996514, 0.9425731976),00097500
PAIR( 0.3325533699, 0.9430844375),PAIR( 0.3311063058, 0.9435934582),00097600
PAIR( 0.3296584625, 0.9441002585),PAIR( 0.3282098436, 0.9446048373),00097700
PAIR( 0.3267604523, 0.9451071933),PAIR( 0.3253102922, 0.9456073254),00097800
PAIR( 0.3238593665, 0.9461052324),PAIR( 0.3224076788, 0.9466009131),00097900
PAIR( 0.3209552324, 0.9470943664),PAIR( 0.3195020308, 0.9475855910),00098000
PAIR( 0.3180480774, 0.9480745859),PAIR( 0.3165933756, 0.9485613499),00098100
PAIR( 0.3151379288, 0.9490458819),PAIR( 0.3136817404, 0.9495281806),00098200
PAIR( 0.3122248139, 0.9500082450),PAIR( 0.3107671527, 0.9504860739),00098300
PAIR( 0.3093087603, 0.9509616663),PAIR( 0.3078496400, 0.9514350210),00098400
PAIR( 0.3063897954, 0.9519061368),PAIR( 0.3049292297, 0.9523750127),00098500
PAIR( 0.3034679466, 0.9528416476),PAIR( 0.3020059493, 0.9533060404),00098600
PAIR( 0.3005432414, 0.9537681899),PAIR( 0.2990798263, 0.9542280951),00098700
PAIR( 0.2976157074, 0.9546857549),PAIR( 0.2961508882, 0.9551411683),00098800
PAIR( 0.2946853722, 0.9555943341),PAIR( 0.2932191627, 0.9560452513),00098900
PAIR( 0.2917522632, 0.9564939189),PAIR( 0.2902846773, 0.9569403357),00099000
PAIR( 0.2888164082, 0.9573845008),PAIR( 0.2873474595, 0.9578264130),00099100
PAIR( 0.2858778347, 0.9582660714),PAIR( 0.2844075372, 0.9587034749),00099200
PAIR( 0.2829365705, 0.9591386225),PAIR( 0.2814649379, 0.9595715131),00099300
PAIR( 0.2799926431, 0.9600021457),PAIR( 0.2785196894, 0.9604305194),00099400
PAIR( 0.2770460803, 0.9608566331),PAIR( 0.2755718193, 0.9612804858),00099500
PAIR( 0.2740969099, 0.9617020765),PAIR( 0.2726213555, 0.9621214043),00099600
PAIR( 0.2711451595, 0.9625384680),PAIR( 0.2696683256, 0.9629532669),00099700
PAIR( 0.2681908571, 0.9633657998),PAIR( 0.2667127575, 0.9637760658),00099800
PAIR( 0.2652340303, 0.9641840640),PAIR( 0.2637546790, 0.9645897933),00099900
PAIR( 0.2622747070, 0.9649932529),PAIR( 0.2607941179, 0.9653944417),00100000
PAIR( 0.2593129151, 0.9657933589),PAIR( 0.2578311022, 0.9661900034),00100100
PAIR( 0.2563486825, 0.9665843745),PAIR( 0.2548656596, 0.9669764710),00100200
PAIR( 0.2533820370, 0.9673662922),PAIR( 0.2518978182, 0.9677538371),00100300
PAIR( 0.2504130066, 0.9681391047),PAIR( 0.2489276057, 0.9685220943),00100400
PAIR( 0.2474416192, 0.9689028048),PAIR( 0.2459550503, 0.9692812354),00100500
PAIR( 0.2444679027, 0.9696573851),PAIR( 0.2429801799, 0.9700312532),00100600
PAIR( 0.2414918853, 0.9704028387),PAIR( 0.2400030224, 0.9707721407),00100700
PAIR( 0.2385135948, 0.9711391585),PAIR( 0.2370236060, 0.9715038910),00100800
PAIR( 0.2355330594, 0.9718663375),PAIR( 0.2340419586, 0.9722264971),00100900
PAIR( 0.2325503070, 0.9725843689),PAIR( 0.2310581083, 0.9729399522),00101000
PAIR( 0.2295653658, 0.9732932461),PAIR( 0.2280720832, 0.9736442497),00101100
PAIR( 0.2265782638, 0.9739929622),PAIR( 0.2250839114, 0.9743393828),00101200
PAIR( 0.2235890292, 0.9746835107),PAIR( 0.2220936210, 0.9750253451),00101300
PAIR( 0.2205976901, 0.9753648851),PAIR( 0.2191012402, 0.9757021300),00101400
PAIR( 0.2176042746, 0.9760370790),PAIR( 0.2161067971, 0.9763697313),00101500
PAIR( 0.2146088110, 0.9767000861),PAIR( 0.2131103199, 0.9770281427),00101600
PAIR( 0.2116113274, 0.9773539001),PAIR( 0.2101118369, 0.9776773578),00101700
PAIR( 0.2086118520, 0.9779985149),PAIR( 0.2071113762, 0.9783173707),00101800
PAIR( 0.2056104131, 0.9786339244),PAIR( 0.2041089661, 0.9789481753),00101900
PAIR( 0.2026070388, 0.9792601226),PAIR( 0.2011046348, 0.9795697657),00102000
PAIR( 0.1996017576, 0.9798771037),PAIR( 0.1980984107, 0.9801821360),00102100
PAIR( 0.1965945977, 0.9804848618),PAIR( 0.1950903220, 0.9807852804),00102200
PAIR( 0.1935855873, 0.9810833912),PAIR( 0.1920803970, 0.9813791933),00102300
PAIR( 0.1905747548, 0.9816726862),PAIR( 0.1890686641, 0.9819638691),00102400
```

```
PAIR( 0.1875621286, 0.9822527414),PAIR( 0.1860551517, 0.9825393023),00102500
PAIR( 0.1845477369, 0.9828235512),PAIR( 0.1830398880, 0.9831054874),00102600
PAIR( 0.1815316083, 0.9833851103),PAIR( 0.1800229014, 0.9836624192),00102700
PAIR( 0.1785137709, 0.9839374134),PAIR( 0.1770042204, 0.9842100924),00102800
PAIR( 0.1754942534, 0.9844804554),PAIR( 0.1739838734, 0.9847485018),00102900
PAIR( 0.1724730840, 0.9850142310),PAIR( 0.1709618888, 0.9852776424),00103000
PAIR( 0.1694502912, 0.9855387353),PAIR( 0.1679382950, 0.9857975092),00103100
PAIR( 0.1664259035, 0.9860539633),PAIR( 0.1649131205, 0.9863080972),00103200
PAIR( 0.1633999494, 0.9865599103),PAIR( 0.1618863938, 0.9868094018),00103300
PAIR( 0.1603724572, 0.9870565713),PAIR( 0.1588581433, 0.9873014182),00103400
PAIR( 0.1573434556, 0.9875439418),PAIR( 0.1558283977, 0.9877841416),00103500
PAIR( 0.1543129730, 0.9880220171),PAIR( 0.1527971853, 0.9882575677),00103600
PAIR( 0.1512810380, 0.9884907929),PAIR( 0.1497645347, 0.9887216920),00103700
PAIR( 0.1482476790, 0.9889502645),PAIR( 0.1467304745, 0.9891765100),00103800
PAIR( 0.1452129247, 0.9894004278),PAIR( 0.1436950332, 0.9896220175),00103900
PAIR( 0.1421768035, 0.9898412785),PAIR( 0.1406582393, 0.9900582103),00104000
PAIR( 0.1391393442, 0.9902728124),PAIR( 0.1376201216, 0.9904850843),00104100
PAIR( 0.1361005752, 0.9906950254),PAIR( 0.1345807085, 0.9909026354),00104200
PAIR( 0.1330605252, 0.9911079137),PAIR( 0.1315400287, 0.9913108598),00104300
PAIR( 0.1300192227, 0.9915114733),PAIR( 0.1284981108, 0.9917097537),00104400
PAIR( 0.1269766965, 0.9919057004),PAIR( 0.1254549834, 0.9920993131),00104500
PAIR( 0.1239329751, 0.9922905913),PAIR( 0.1224106752, 0.9924795346),00104600
PAIR( 0.1208880872, 0.9926661425),PAIR( 0.1193652148, 0.9928504145),00104700
PAIR( 0.1178420615, 0.9930323502),PAIR( 0.1163186309, 0.9932119492),00104800
PAIR( 0.1147949266, 0.9933892111),PAIR( 0.1132709522, 0.9935641355),00104900
PAIR( 0.1117467112, 0.9937367219),PAIR( 0.1102222073, 0.9939069700),00105000
PAIR( 0.1086974440, 0.9940748793),PAIR( 0.1071724250, 0.9942404495),00105100
PAIR( 0.1056471537, 0.9944036801),PAIR( 0.1041216339, 0.9945645707),00105200
PAIR( 0.1025958690, 0.9947231211),PAIR( 0.1010698628, 0.9948793308),00105300
PAIR( 0.0995436187, 0.9950331994),PAIR( 0.0980171403, 0.9951847267),00105400
PAIR( 0.0964904314, 0.9953339121),PAIR( 0.0949634953, 0.9954807555),00105500
PAIR( 0.0934363358, 0.9956252564),PAIR( 0.0919089565, 0.9957674145),00105600
PAIR( 0.0903813609, 0.9959072294),PAIR( 0.0888535526, 0.9960447009),00105700
PAIR( 0.0873255352, 0.9961798286),PAIR( 0.0857973123, 0.9963126122),00105800
PAIR( 0.0842688876, 0.9964430514),PAIR( 0.0827402645, 0.9965711458),00105900
PAIR( 0.0812114468, 0.9966968952),PAIR( 0.0796824380, 0.9968202993),00106000
PAIR( 0.0781532416, 0.9969413578),PAIR( 0.0766238614, 0.9970600703),00106100
PAIR( 0.0750943008, 0.9971764367),PAIR( 0.0735645636, 0.9972904567),00106200
PAIR( 0.0720346532, 0.9974021299),PAIR( 0.0705045734, 0.9975114561),00106300
PAIR( 0.0689743276, 0.9976184351),PAIR( 0.0674439196, 0.9977230666),00106400
PAIR( 0.0659133528, 0.9978253504),PAIR( 0.0643826309, 0.9979252862),00106500
PAIR( 0.0628517576, 0.9980228738),PAIR( 0.0613207363, 0.9981181129),00106600
PAIR( 0.0597895707, 0.9982110034),PAIR( 0.0582582645, 0.9983015449),00106700
PAIR( 0.0567268212, 0.9983897374),PAIR( 0.0551952443, 0.9984755806),00106800
PAIR( 0.0536635377, 0.9985590742),PAIR( 0.0521317047, 0.9986402182),00106900
PAIR( 0.0505997490, 0.9987190122),PAIR( 0.0490676743, 0.9987954562),00107000
PAIR( 0.0475354842, 0.9988695499),PAIR( 0.0460031821, 0.9989412932),00107100
PAIR( 0.0444707719, 0.9990106859),PAIR( 0.0429382569, 0.9990777278),00107200
PAIR( 0.0414056410, 0.9991424187),PAIR( 0.0398729276, 0.9992047586),00107300
PAIR( 0.0383401204, 0.9992647473),PAIR( 0.0368072229, 0.9993223846),00107400
PAIR( 0.0352742389, 0.9993776704),PAIR( 0.0337411719, 0.9994306046),00107500
PAIR( 0.0322080254, 0.9994811870),PAIR( 0.0306748032, 0.9995294175),00107600
PAIR( 0.0291415088, 0.9995752961),PAIR( 0.0276081458, 0.9996188225),00107700
PAIR( 0.0260747178, 0.9996599967),PAIR( 0.0245412285, 0.9996988187),00107800
PAIR( 0.0230076815, 0.9997352883),PAIR( 0.0214740803, 0.9997694053),00107900
PAIR( 0.0199404286, 0.9998011699),PAIR( 0.0184067299, 0.9998305818),00108000
PAIR( 0.0168729879, 0.9998576410),PAIR( 0.0153392063, 0.9998823475),00108100
```

```
    PAIR( 0.0138053885, 0.9999047011),PAIR( 0.0122715383, 0.9999247018),00108200
    PAIR( 0.0107376592, 0.9999423497),PAIR( 0.0092037548, 0.9999576446),00108300
    PAIR( 0.0076698287, 0.9999705864),PAIR( 0.0061358846, 0.9999811753),00108400
    PAIR( 0.0046019261, 0.9999894111),PAIR( 0.0030679568, 0.9999952938),00108500
    PAIR( 0.0015339802, 0.9999988235),                                   00108600
    FILL        128;                                                     00108700
REINDEX[ENTRY];;                                                         00108800
    DATA     0.10204102040;8,102041020;8,10306143060;8,1020410;8,        00108900
         10205122450;8,103061430;8,10307163470;8,10204;8,                00109000
 10204112244;8,102051224;8,10306153264;8,1030614;8,10205132654;8,        00109100
 103071634;8,10307173674;8,102;8,10204102142;8,102041122;8,              00109200
 10306143162;8,1020512;8,10205122552;8,103061532;8,10307163572;8,        00109300
 10306;8,10204112346;8,102051326;8,10306153366;8,1030716;8,              00109400
10205132756;8,103071736;8,10307173776;8,1,10204102041;8,102041021;8,     00109500
10306143061;8,1020411;8,10205122451;8,103061431;8,10307163471;8,         00109600
 10205;8,10204112245;8,102051225;8,10306153265;8,1030615;8,              00109700
 10205132655;8,103071635;8,10307173675;8,103;8,10204102143;8,            00109800
 102041123;8,10306143163;8,1020513;8,10205122553;8,103061533;8,          00109900
 10307163573;8,10307;8,10204112347;8,102051327;8,10306153367;8,          00110000
 1030717;8,10205132757;8,103071737;8,10307173777;8;                      00110100
END SIN.                                                                 00110200
```

REFERENCES

[1] Cooley, J. W., Lewis, P. A. W., Welch, P. D., "The Fast Fourier Transform Algorithm and its Application," IBM Watson Research Center, Yorktown Heights, New York, February, 1967.

[2] Barnes, G. H., Brown, R. M., Kato, M., Kuck, D. J., Slotnick, D. L., Stokes, R. A., "The ILLIAC IV Computer," IEEE Transactions on Computers C-17, No. 8 746 (1968).

[3] Ackins, G. M., "Fast Fourier Transform via ILLIAC IV," ILLIAC IV Document No. 198, July 12, 1968.

[4] Robinson, E. A., Multichannel Time Series Analysis with Digital Computer Programs, Holden-Day (1967), p. 290.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Center for Advanced Computation University of Illinois at Urbana-Champaign Urbana, Illinois 61801	**2a. REPORT SECURITY CLASSIFICATION** UNCLASSIFIED **2b. GROUP**

3. REPORT TITLE

A FAST FOURIER TRANSFORM SUBROUTINE FOR ILLIAC IV

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Research Report

5. AUTHOR(S) *(First name, middle initial, last name)*

James E. Stevens, Jr.

6. REPORT DATE October 15, 1971	**7a. TOTAL NO. OF PAGES** 67	**7b. NO. OF REFS** 4

8a. CONTRACT OR GRANT NO. DAHC04-72-C-0001 **b. PROJECT NO.** ARPA Order 1899 **c.** **d.**	**9a. ORIGINATOR'S REPORT NUMBER(S)** CAC Document No. 17 **9b. OTHER REPORT NO(S)** *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Copies may be obtained from the address given in (1) above.

11. SUPPLEMENTARY NOTES None.	**12. SPONSORING MILITARY ACTIVITY** U.S. Army Research Office-Durham Duke Station Durham, North Carolina

13. ABSTRACT

This report is a description of a Fast Fourier Transform (FFT) subroutine written in assembly language for the ILLIAC IV computer. The subroutine uses the Cooley-Tukey algorithm for performing discrete Fourier transforms. The parallel nature of the Cooley-Tukey method lends itself very well to a highly parallel machine like ILLIAC IV. Timing simulation results have shown that this program will perform Fourier transforms faster than they are being done on any existing computer system.

DD FORM 1 NOV 65 **1473**

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Seismic Data Processing						
ASK						
Fast Fourier Transform (FFT)						
Discrete Fourier Transform (DFT)						
Time Series						
Frequency Spectrum						