## CONTENTS

<table>
<thead>
<tr>
<th>N.I.O. PROGRAMS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Card sorting</td>
<td>March 1958</td>
</tr>
<tr>
<td>2 Cross band spectra</td>
<td>September 1958</td>
</tr>
<tr>
<td>3 Fourier analysis, R.A.B 528 modified</td>
<td>December 1959</td>
</tr>
<tr>
<td>4 Card sorting, Southport</td>
<td>December 1959</td>
</tr>
<tr>
<td>5 Correlation, R.A.B 506 modified</td>
<td>September 1959</td>
</tr>
<tr>
<td>6 Pitch and roll buoy predominant azimuth angles</td>
<td>March 1960</td>
</tr>
<tr>
<td>7 Pitch and roll buoy filtered directional spectrum</td>
<td>March 1960</td>
</tr>
<tr>
<td>8 Smoothing</td>
<td>March 1960</td>
</tr>
<tr>
<td>9 Dover Straits</td>
<td>May 1960</td>
</tr>
<tr>
<td>10 Combining two series</td>
<td>March 1959</td>
</tr>
<tr>
<td>11 Sixth differences</td>
<td>October 1959</td>
</tr>
<tr>
<td>12 Slope correction to wave heights</td>
<td>November 1959</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R.A.E. PROGRAMS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>174 Correlation</td>
<td>21</td>
</tr>
<tr>
<td>300 Fourier analysis</td>
<td>24</td>
</tr>
<tr>
<td>528 Fourier analysis</td>
<td>25</td>
</tr>
</tbody>
</table>
This report contains a collection of programs written for Deuce between January 1958 and June 1960. Programs 5, 10 and part of 2 are due to Mrs. Pamela Edwards, who has also done a considerable amount of work on the final editing of the program cards and on preparing the descriptions.

The programs were developed (and are now used for production runs) on the two computers at the Royal Aircraft Establishment, Farnborough; we should like to thank the Mathematical Services Department for their help.

The descriptions of the programs (including a few frequently used standard ones) are complete only when read in conjunction with the general remarks under the appropriate heading; this is particularly important when preparing data and operating the computer.

The report is intended primarily as a manual for the operator, and so only brief comments on the mathematics and programming methods are given.

Title This includes the source of the program, if it is not original. Most of the original programs use standard subroutines.

Code The present possibilities are

Deuce: (a) Deuce basic
(b) Alphacode
(c) Programs under the control of a general interpretive program (GIP), for example, Scheme B.

Purpose This amplifies the title.

Order of cards This gives the complete assembly for input to Deuce.

Data These comments apply also to the parameter cards and end cards. Unless otherwise stated, the following conventions are observed:

1. Unspecified columns may be left blank, and this usually includes the whole of the β-field. Anything punched on an unspecified column in the α-field is ignored, but this is not necessarily true for the β-field.

2. All specified columns must be punched, zeros being inserted where necessary. Decimal numbers are punched from left to right in the usual way. Positive signs (that is, holes in the Y-row) need not be punched.

3. The columns containing data are not stated explicitly when the layout is obvious. For example, 8 signed 2-digit numbers fit into the α-field in only one way, and the signs must be overpunched; that is, in the same column as the first digit.

4. In general the data is in decimal form, but layout parameters, for example, are in binary.

5. When the position of the data is specified by a layout parameter, it is not necessary to include all the digits appearing on the card, provided that the specified digits (and the sign) are consecutive.

Operation Before starting, make sure that the stop-normal key is up - that is, at normal. Unless otherwise stated, all other keys should be in the horizontal position. The field switches on the reader and the punch must be at the 64 column setting.

After reading the program cards with the initial input key, wait until the program stops before running in the data (that is, reading with the run in key). (Large sets of data may be read in sections.)

The stopper is sometimes specified, but in most cases it is either N, 16 - 1
or N, N - 13
where 1 ≤ N ≤ 8, and depends on where the read subroutine is stored.
All cards must be placed in the reader with the Y row downwards, that is, leading, and with the face of the card away from the operator.

The program may be restarted by reading the last card (if both \(\alpha\) and \(\beta\)-fields are punched) or the last two cards (if only the \(\alpha\)-field is punched) with the initial input key. This is not true for programs which only use the high speed store. Unless otherwise stated, it may be assumed that the last card restarts the program.

Output
Unspecified columns are left blank. A few of the programs fit standard listing boards, but in general a board must be plugged to fit the output.

Parameters
These are either set to the most commonly required values, or the position is left blank. Careful checking is essential. Parameters are often binary numbers (see Note 1).

Ideally, there should be nothing in this section, as variables are more conveniently set by a special parameter card. But it is sometimes easier to set those variables which are constant for most sets of data within the program, leaving the parameter card for more variable quantities.

Restrictions
These have been specified and tested as fully as possible, but it should be remembered that some deficiency in a program might come to light only when a particular case is encountered - perhaps after many hours of production. All results should be checked as carefully as possible.

Failures
Besides the specified failures, a program may not work because of the unforeseen cases mentioned above (see Restrictions). Also older programs may not work because of later modifications to the computer and/or the timing connected with the reader and the punch.

Time
In general this excludes reading the program, but includes reading data, the calculation and the punching time. The approximate times given are intended as a guide, and should not be relied on too much.

Method
This section contains definitions of the symbols used elsewhere in the description, together with a summary of the method. In general,

\[
\begin{align*}
n &= \text{number of terms in a series} \\
N &= \text{other counts.}
\end{align*}
\]

Notes
(i) A number \(x = a \cdot 2^b\) is said to be in standard floating binary (s.f.b.) if

- either \(-1 < a < -\frac{1}{2}\),
- \(\frac{1}{2} < a < 1\),
- \(a = 0\),

and \(b\) is an integer. The store of Deuce also demands that \(a\) is specified to 30 bp (binary places) and that \(-2^{31} < b < 2^{31}\).
Title Card sorting (written for Job 459, MS, RAE). RAE Program 566.

Code Deuce basic.

Purpose To separate 4 series.

Order of cards Program (cards 0-27), data.

Parameters None.

Data 8 positive 4 digit numbers per card in the $\alpha$-field. The number of cards must be a multiple of 4, and so up to 3 extra cards filled with zeros may be necessary. Unused $\alpha$-field columns on the last data card must be filled with zeros.

End card None.

Operation Read the program with the initial input key and then run in the data. Run in more data when required. The last card alone will not restart the program.

Output 8 positive 4 digit numbers per card in the $\alpha$-field. Signs are punched.

Parameters None.

Restrictions The number of data cards must be a multiple of 4.

Failures None.

Time About $N/36$ minutes, where $N$ is the number of data cards.

Method The program alternately reads 4 cards and punches 4 cards. The numbers are rearranged as follows:

Input: Card 1 $x_0$ $x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$
2 $x_8$ $x_9$ $x_{10}$ $x_{11}$ $x_{12}$ $x_{13}$ $x_{14}$ $x_{15}$
3 $x_{16}$ $x_{17}$ $x_{18}$ $x_{19}$ $x_{20}$ $x_{21}$ $x_{22}$ $x_{23}$
4 $x_{24}$ $x_{25}$ $x_{26}$ $x_{27}$ $x_{28}$ $x_{29}$ $x_{30}$ $x_{31}$

Output: Card 1 $x_0$ $x_4$ $x_8$ $x_{12}$ $x_{16}$ $x_{20}$ $x_{24}$ $x_{28}$
2 $x_1$ $x_5$ $x_9$ $x_{13}$ $x_{17}$ $x_{21}$ $x_{25}$ $x_{29}$
3 $x_2$ $x_6$ $x_{10}$ $x_{14}$ $x_{18}$ $x_{22}$ $x_{26}$ $x_{30}$
4 $x_3$ $x_7$ $x_{11}$ $x_{15}$ $x_{19}$ $x_{23}$ $x_{27}$ $x_{31}$

Notes
1) The original data came from a film containing 4 traces, which were read concurrently.

2) The 4 packs of cards are finally separated by the Hollerith collator or by the sorter.
N.I.O. PROGRAM 2

Title Cross band spectra, for use with output of RAE 300 (periodogram). In its present form it cannot be used with output of RAE 528.

Code Deuce basic.

Purpose To condense and combine the periodogram outputs of two related sets of data.

Order of cards Program (cards 0-52), series 1 of data, end card, series 2 of data, end card.

Parameter card None.

Data Output of periodogram program RAE 300.

Columns 17 - 20 s; sign in column 17
21 - 30 $I_a$; 21
31 - 38 $A_a$; 31
39 - 46 $B_a$; 39

All the numbers are integers. I may not have more than eight significant digits, and $A$ and $B$ may not have more than six significant digits.

End card $P_1$ on row 1; $P_1$ on row 2.

Operation (1) Read the program with the initial input key.
(2) Run in series 1 of data, followed by an end card.
(3) Run in series 2 of data, followed by an end card.
(4) The machine will stop on 5, 0-15, when the value $N.P$ should be set up on the I.D. switches in binary;
(5) Single shot. This will allow the program to continue and to punch the first output card. It stops a sin on 5, 0-13.
(6) Either leave the former value of $N$ on the I.D. switches or set up a new value of $N$.

Repeat (5) and (6) until all the output cards have been punched, when the machine will hoot.

To use the program with new data, read in the last card of the program with the initial input key. Then run in the data as usual.

Output Columns 17 - 20 $s$;
21 - 24 $N$
25 - 28 $s_N$
29 - 32 $b$
33 - 40 $E_i$; sign in column 33
41 - 46 $E_l$; 41
49 - 56 $10^{-2}E_a$; 49
57 - 54 $10^{-2}E_b$; 57
65 - 72 $10^{-2}E_i^2$
73 - 80 $10^{-2}E_l^2$

All these numbers are integers.

Parameters The scaling factors are chosen according to the magnitude of the input and the required accuracy of the output. Full details may be found in Computing Folder 1, which is kept by the person in charge of the N.I.O. library programs.

(1) $k_1, \delta_1$ on card number 25, row 3; $k_1 = 7$ is set
(2) $k_2, \delta_2$ on card number 25, rows 4, 5; $k_2 = 4$ is set.
These values allow I and i to have seven significant digits, and A, a, B and β to have four (and probably some terms of five) significant digits.

Restrictions
1. \( n_1 = n_2 < 511 \). If the input data exceeds this number it must be broken up into several batches; in this case the last card of the program must be run in in front of each batch of data.

2. \( b = a - a/N-1 < 19 \). If \( b > 19 \) the calculation will be carried out correctly but \( b_1 \) and \( b_2 \) will be wrongly punched.

Failures
1. 4, 7 - 2\( \frac{1}{2} \) x \( s \) (series 1) \( \neq s \) (series 2)
2. 8, 7 - 2\( \frac{1}{2} \) x \( n_1 \neq n_2 \)
3. 2, 0 - 1\( \frac{1}{2} \) x more data required
4. 6, 7 - 2\( \frac{1}{2} \) x \( b \) is too large, i.e. \( b > 20 \). Give a single shot to continue.
5. 5, 0 - 1\( \frac{1}{2} \) x (after hook) \( EN > n \) (see note 2)
6. 1, 7 - 2\( \frac{1}{2} \) x overflow in one of \( EI, Ei, EA, EB \). A single shot allows the program to continue. A large overflow may not be recorded. See separate note for further details.

Time
About \( n(N+8)/N \) seconds.

Method
\( s = a(b)c \), as defined in RAE 300.
\( n = \) total number of input cards in one series.
\( N = \) number of terms in one group.
\( s_1 \) and \( s_N \) are the first and last values of \( s \) in any one group of \( N \).
I, A and B are \( I_1, A_1 \) and \( B_1 \) as defined in RAE 300.
i, a and \( \beta \) are similarly for a second series of data.
\( k_1 \) and \( k_2 \) are scaling factor constants.

Notes
(1) If \( N \) is to remain constant the stopper on card number 37, row 2\( \frac{1}{2} \), can be removed (the instruction here is 0-13 x); in this case \( N,P \) must be set up on the I.D. switches before the data are run in.
(2) If the last group of data has fewer than \( N \) values, \( N \) may be "counted down" on the I.D. switches until a single shot has effect, that is, until the new value of \( N \) is appropriate.
(3) The following convention may be followed for the "optional" output columns:
- Columns 5 - 7 job number
- 8 - 10 run and parameter numbers
- 11, 12 scaling factors for (1) I and i, and (2) A, a, B and \( \beta \).
(4) This program is going to be rewritten.
N.I.O. PROGRAM 5

Title Fourier analysis (periodogram), RAE 528 modified. RAE Program 567.

Code Deuce basic.

Purpose To calculate the harmonic components $A_s$, $B_s$ for selected $s$.

Order of cards Program (cards 0-31), $s$-parameter card, $s$-data, parameter card, data.

Parameter cards (1) $s$-parameter card: columns 17-20 $N + 1$
(2) $s$-data: the $N$ values of $s$ are punched as 8 positive 4 digit integers per card in the $s$-field. An x (minus) must be punched in the column after the last value of $s$. This may mean starting a fresh card.
(3) Parameter card: columns 17-20 $n$, $2^1-2^4$ $k$.

Data The $n$ values of $y_r$ are punched as 8 signed 4 digit integers per card, in the $s$-field.

End card None.

Operation (1) Read the program with the initial input key.
(2) Run in the $s$-parameter card followed by the $s$-data.
(3) Run in the parameter card followed by the data.

When a run is complete the program stops, and operation (3) may be repeated. If new $s$-data is required the program must be restarted.

Output Columns 18 - 21 $s$ ; sign in column 17
$24 - 30$ $A_s$ ; $23$
$33 - 39$ $B_s$ ; $32$
$42 - 48$ $I_s$ ; $41$

There are three more decimal places in $A_s$, $B_s$ and $I_s$ than in $y_r$.

Parameters None

Restrictions (1) $0 < s < 9999$ (not tested for $s > 4428$)
(2) $1 < N < 95$
(3) $n^2 M < 2 \cdot 10^9$ where $|y_r| < M$, $r = 0(1) n-1$

Failures None

Time About $nN/200$ seconds

Method

\[ A_0 = \frac{1}{n} \sum_{r=0}^{n-1} y_r \]  \quad $B_0 = 0$ \quad when $s = 0$

\[ A_s = \frac{2}{n} \sum_{r=0}^{n-1} y_r \cos \left( \frac{2\pi rs}{kn} \right) \]  \quad when $s > 0$

\[ B_s = \frac{2}{n} \sum_{r=0}^{n-1} y_r \sin \left( \frac{2\pi rs}{kn} \right) \]  \quad when $s > 0$

\[ I_s = \sqrt{A_s^2 + B_s^2} \quad \text{for all} \ s,

where $s$ is an integer. If $0 < c < 13$ is an integer, then $k = 2^c$.

Notes (1) The method of calculation is described in The Computer Journal, 1, 4, p. 162.
N.I.O. PROGRAM 4

Title Card sorting, Southport. Written for Job 852 (MS, RAE). RAE Program 569.

Code Deuce basic.

Purpose Given one value of each of N series per card, to form N packs of cards, one pack per series, and with eight values per card.

Order of cards Program (cards 0-31), parameter card, data.

Parameter card Columns 17-20 m

Data N signed 4 digit numbers in the α-field, punched consecutively, and with overpunched signs, with the first sign and digit in column 17.

End card None.

Operation (1) Read in the program with the initial input key. The program stops on 7, 5-5.
(2) Run in the parameter card and data. The program is still at 7, 5-5.
(3) Single shot. The program reads the parameter card and data and stops on 7, 13-12.
(4) Single shot. The first set of results is punched. The program stops at 7, 13-12.
(5) Repeat (4), making N times in all, and N sets of output. The program does not stop at 7, 13-12 after the Nth time.
(6) The hooter sounds and the program stops at 7, 5-5. After clearing the hooter, operations (2) to (6) may be repeated with more data.

Output N packs of cards, with 8 signed 4 digit numbers per card. If m/64 is an integer, each pack will contain m/64 cards. Otherwise, each pack will contain the integral part of (m/64 + 1) cards.

Parameters (N-1),R7 on card number 5, row 9%. This is set for N = 6.

Restrictions (1) 1 ≤ N ≤ 8
(2) 8 ≤ m ≤ 7520, that is, 1 ≤ number of cards ≤ 940

Failures None

Time About Nm/40 seconds.

Method The values on columns 17-20 of the input data are punched successively on the first pack of output cards, with eight values per card. Then the values from columns 21-24 form the second pack of output, and so on for the N series, giving N packs of output.

We define m as 8 x the number of data cards.

Notes (1) The stopper after stage (3) of Operation allows time for the manual setting of new series numbers, zeroising card numbers, running out cards and so on.
(2) If any column in the input is left blank the corresponding output column contains a nine.
(3) If m/64 is not an integer (that is, if the number of cards is not a multiple of 8), the last output card will be completed with zeros.
Title  Correlation, based on RAE 506 (unavailable) (M3, RAE). RAE Program 548.

Purpose To find the autocorrelation coefficients of a time series or the cross correlation coefficients of two series.

Order of cards Program (cards 0-55), parameter card, data.

Parameter card Ya row: data layout. A one in each digit (or sign) position except the last digit of a number, and zeros everywhere else.

The rest of the card is exactly as for RAE 174 except that zeros must be punched instead of blank columns. No positive signs may be punched.

Data Either 1 or 2 signed numbers per card. If 2 are specified the first is series 1 and the second series 2; if only 1 is specified it is called series 2, in which case the autocorrelation parameter is $K_2$. Columns other than those specified are ignored. The sign is in the column preceding the first digit, and the first sign must be in column 17.

End card None.

Operation (1) Read the program with the initial input key.
(2) Run in the parameter card, followed by the data. When one run is complete, (2) may be repeated.

Output See RAE 174.

Parameters (1) $c.P$, in binary on card 8, row $Y_a$. Here $c = 5000$.
(2) $r.P$, in binary on card 7, row $Y_a$. Here $r = 5$.
(3) $10.P$, in binary on card 8, row $6_x$.

Restrictions (1) $0 < n < 3488$
(2) $0 < K_1, K_2 < 159$
(3) $0 < L_1, L_2 < 127$
(4) $-5005 < x_i, y_i < 4995$, $i = 0(1)n-1$

Failures See RAE 174. There are also 2 failures peculiar to this program, namely:
(1) 6, 6-6, with hooter; $x_i$ and/or $y_i > 4994$
(2) 6, 7-7, with hooter; $x_i$ and/or $y_i < -5005$.

Time About $N/60 + n$ seconds, where $N$ is any of $K_1, K_2, L_1, L_2$.

Method The main calculation is exactly the same as for RAE 174. Before entering the main section the data are made to satisfy certain conditions. If the original number $x_i$ satisfies Restriction (4), then the program calculates $Z_i$, where

$$Z_i = \frac{x_i + c + r}{10}$$

and $0 \leq Z_i \leq 999$.

Notes (1) The checks are the same as for RAE 174.
(2) It is clear that by juggling with Parameters (1), (2) and (3) and Restriction (4), and taking account of the relation between $x_i$ and $Z_i$, almost any data may be dealt with and made to satisfy $0 \leq Z_i \leq 999$. The failures depend on $c$ rather than on the actual values named in Restriction (4).
N.I.O. PROGRAM 6

Title Pitch and roll buoy predominant azimuth angles. Written for job 832 part 1 (MS, MAR).

Code Deuce basic.

Purpose To find the two main directions of wave propagation from the first five Fourier coefficients.

Order of cards Program (cards 0-37), data.

Parameter card None.

Data Columns 17 - 18 10σ ; sign in column 17
19 - 24 10^3a_0 ; 19
25 - 30 10^3a_1 ; 25
31 - 36 10^3b_1 ; 31
37 - 42 10^3a_2 ; 37
43 - 48 10^3b_2 ; 43

All the numbers are integers, and 6 digits (2 for 10σ) must be punched.

End card None.

Operation Read the program with the initial input key and then run in the data. The program alternately reads 1 card and punches 1 card until no more data is left. Further data cards may then be run in.

Output Columns 17 - 19 10σ ; sign in column 17
21 - 24 θ (degrees) ; 21
25 - 26 (minutes) ; 25
28 - 31 ψ (degrees) ; 28
32 - 33 (minutes) ; 32
35 - 40 10^3(c_0-I) ; 35
42 - 47 10^3I ; 42

Parameters None.

Restrictions (1) 0 < |σ| < 9.9
(2) 0 < |a_0| < 999.999
(3) 0 < |a_1| < 999.999, and similarly for b_1, a_2, b_2.

Failures If the data are incompatible (for example, if they would give |cos ψ| > 1) the program will fail. It may stop on 4, 24-14 (that is, a number > 1 is about to be produced during the floating to fixed point conversion) or there may be other stoppers in the standard subroutines, or there may be no indication at all.

Time About N/14 minutes, where N is the number of data cards.

Method σ = reference number
θ = \cos^{-1} \sqrt{\frac{x^2 + y^2}{2}} ; 0 ≤ θ ≤ 2π
ψ = \cos^{-1} \sqrt{\frac{z^2}{2}} ; 0 ≤ ψ ≤ 2π
a_0I = a_0^2 - a_1^2 - b_1^2 - (p^2 + q^2)^{\frac{1}{2}}
where
x = (a_0 - I)a_1 - (a_1a_2 + b_1b_2)
y = (a_0 - I)b_1 - (a_1b_2 - a_2b_1)
z = (a_0 - I)a_2^2 - (a_2^2 + b_2^2)
\[ P = a_0 a_2 - a_1^2 + b_1^2 \]
\[ Q = a_0 b_2 - 2a_1 b_1 \]

**Notes**


(2) The data is the same as in N.I.O., Program 7.
N.I.O. PROGRAM 7

Title Pitch and roll buoy filtered directional spectrum. Written for job 832 part 2 (NG, RAE).

Code Deuce basic.

Purpose To obtain a smoothed estimate of the directional energy spectrum of the waves from the first five Fourier coefficients.

Order of cards Program (cards 0-34), data.

Parameter card None.

Data Columns

<table>
<thead>
<tr>
<th>Column Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17-18</td>
<td>$10^e$</td>
</tr>
<tr>
<td>19-24</td>
<td>$10^a_e$</td>
</tr>
<tr>
<td>25-30</td>
<td>$10^a_1$</td>
</tr>
<tr>
<td>31-36</td>
<td>$10^b_1$</td>
</tr>
<tr>
<td>37-42</td>
<td>$10^a_2$</td>
</tr>
<tr>
<td>43-48</td>
<td>$10^b_2$</td>
</tr>
</tbody>
</table>

Sign in column 17.

All the numbers are integers, and 6 digits (2 for $10^e$) must be punched.

End card None.

Operation Read the program with the initial input key and then run in the data. The program alternately reads 1 card and punches 12 cards until no more data is left. Further data cards may then be run in.

Output Columns

<table>
<thead>
<tr>
<th>Column Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17-19</td>
<td>$10^e$</td>
</tr>
<tr>
<td>21-24</td>
<td>$\phi$ (degrees)</td>
</tr>
<tr>
<td>26-32</td>
<td>$10^{-r}$ $F$</td>
</tr>
</tbody>
</table>

Parameters $10^{-r}$ in standard floating binary on card number 29, rows 6, 7, where $r$ is defined in Method. Normally $r = 1$.

Restrictions

1. $0 \leq \sigma \leq 9\cdot9$
2. $0 < |a_0| \leq 999\cdot999$
3. $0 < |a_1| \leq 999\cdot999$, and similarly for $a_2$, $b_2$.
4. $10^{-r}|F| < 1 \quad r \geq 0$ See note (1)

Failures None.

Time About N/2 minutes, where N is the number of data cards.

Method

$6F = 3a_0 + 4(a_4 \cos \phi + b_4 \sin \phi) + (a_2 \cos 2\phi + b_2 \sin 2\phi)$,

where $\phi = -180^\circ$ or $+30^\circ$ to $150^\circ$.

An upper bound for $|F|$ can be derived.

Suppose $|\max (a_0, a_1, b_1, a_2, b_2)| = M \leq 999\cdot999$.

Then $|F| < \frac{15M}{67^2} < 0\cdot7 M$.

We must choose the parameter $r > 0$ so that

$|10^{-r} F| < 1$

but is as large as possible subject to this restriction.
N.I.O. PROGRAM 8

Title Smoothing. Written for Job 670 (N3, RAE). RAE Program 570.

Code Deuce basic.

Purpose To scale and smooth a series.

Order of cards Program (cards 0-43), parameter card, data.

Parameter card Columns 25 - 28 a
33 - 36 b
37 - 40 k
41 - 44 p

Data Columns 21 - 30 I_s ; sign in column 21.
See parameter (1).

End card None.

Operation (1) Read the program with the initial input key. It stops at 6, 9-8.
(2) Run in the parameter card, followed by the data. The program is still at 6, 9-8.
(3) Give a single shot.
(4) When one run is completed the program returns to 6, 9-8. Operations (2) and (3) may then be repeated.

Output Columns 17 - 21 a ; sign in column 17
23 - 32 q'_s ; 23
33 - 42 q''_s ; 33

Parameters (1) Data layout. A one in each digit (or sign) position except the last digit of a number, and zeros everywhere else. Column 17 must be zero. This is set as described under Data. The layout parameter is on card 11, row 3_a.
(2) q_s in standard floating binary on card 6, rows 3_a, 3_b.
(3) q_s in standard floating binary on card 6, rows 4_a, 5_a.

Restrictions (1) 0 < c - a + 1 < 1824, a > 0
(2) 1 < p < 912, but the smoothing is useful only when 0 < 2p - 1 ≤ c - a + 1
(3) 0 < |k| < 9999.
(4) |k| N_0 / p^2 < 1, 4|k| N_0 / < 1, where M = max. |I_s / s^2|, and q_s are the largest possible multiples of 10 or s^2 subject to the above restrictions. Normally q_s = q_s = 1. See Note (2).
(5) |I_s| < 2.14 x 10^9.

Failures None.

Time About \( \frac{2}{a} (c - a + 1) \) seconds.

Method The smoothed value \( I_s'' \) is defined by

\[ I_s'' = I_s - p + 2I_{2p-2} + \ldots + (p-1)I_{2p-1} + \ldots + 2I_{s+p-2} + I_{s+p-1} \]
where \( I_s' = \frac{k}{p} I_s / s^2 \), and \( s = a(1)c \).

\[ I_s'' = \frac{k}{p} I_s / s^2 \] (2)

\( I_s \) is given for \( s = a(1)c \) and \( k, p \) are given constant integers.
We define \( I_s'' = 0 \) for \( s = a + p - 2 \) and \( c - p + 2 < s < c \). It is easy to show that

\[ \forall s \neq a + p - 1 \quad I_s'' = I_s - I_s' - \frac{s}{a + p - 1} (I_s' - I_{s-p}) \]

\[ \forall s = a + p - 1 \quad I_s'' = I_s' - 2I_{s-p} + I_{s-p-1} \] (4)

The program has three stages:
Stage 1. $\text{I}_{0}^p$, $\text{I}_{a}^p$ and $\text{I}_{0}^p$ (and hence $\text{VI}_a^p$) are calculated from (1).

Stage 2. $\sqrt{\text{I}_0^p}$ is calculated from (4) and $\text{VI}_0^p$ and $\text{I}_0^p$ are built up for $s = (a + p + 1)(c - p + 1)$. This involves addition and subtraction only, and is far quicker than direct evaluation of (1).

Stage 3. $\text{I}_{0-p+1}^p$ is calculated from (1) and compared with the last value obtained in Stage 2.

Notes

(1) The program was originally intended to accept the output from the periodogram program HAE 300; that is, $s$, $I_s$, $A_g$, and $B_g$ for $s = a(1)c$. The essential number $I_g$ (regarded as an integer) is in columns 21-30, with sign in column 21.

(2) A brief derivation of restriction (4) is as follows:

The stored quantities which must not overflow are $s$, $I_g$, $I_g'$, $I_g''$, $|\text{VI}_g^p|$ and $|\sqrt{\text{I}_0^p}|$, of which the first two are integers with upper limit $2^{24}$ - 1, and the others are fractions with upper limit unity.

By definition, $s$ and $I_g$ are in range.

Let $M = \max \frac{|I_g|}{p^2}$

Then $|I_g'| \leq \frac{|k|M}{p^2}$ (5)

From equations (1) and (2) we see that

$|I_g''| \leq \frac{|k|M}{p^2} [1 + 2 + \ldots + (p - 1) + p + (p - 1) + \ldots + 2 + 1]$

$= |k|M$

Also $|\text{VI}_g''| \leq 2|I_g''|$, $|\sqrt{\text{I}_0^p}| \leq 4|I_g''|$, so the inequality which covers $I_g''$, $\text{VI}_g''$ and $\sqrt{\text{I}_0^p}$ is

$4|I_g''| \leq 4|k|M$ (6)

Equations (5) and (6) give the inequalities of restriction (4), namely,

$\frac{|k|M}{p^2} q_1 \leq 1$

$4|k|M q_2 \leq 1$

where $q_1$ and $q_2$ are independent constants.

(3) It is not possible to give an adequate analysis of the rounding errors here, but it is clear that there may be such errors in the two initial values of $I_g$, and also that the error in the last value of $I_g$ will depend to some extent on the number of steps. The check value of $I_g$ (see Method, Stage 3) will give an indication of this error, and if it is too large to be accepted, then either the range of $s$ must be reduced or the number of decimal places increased by increasing $q_1$ and/or $q_2$.

(4) The program could be adapted to smooth $I_g$ in other ways, for example, by setting $s = 1$ in the appropriate routine. However, it is a special purpose program, and would probably be inefficient in any other context.
Title Dover Straits. Written for job 885 (MS, RAE).

Code Deuce basic.

Purpose Given the barometric pressure at Felixstowe, Tangmere (or Thorney Island) and Calais, to calculate the difference in water level between Ramsgate and Dunkirk due to the pressure difference and wind stress.

Order of cards Program (cards 0-25), data, end card.

Parameter card None.

Data (1) Tangmere.

Columns 17 - 20 reference number; sign in column 17
22 - 26 $p_1$ ; 22
27 - 31 $p_2$ ; 27
37 - 41 $p_3$ ; 37

(2) Thorney Island

Columns 17 - 20 reference number; sign in column 17
22 - 26 $p_1$ ; 22
27 $x$ (minus)
32 - 36 $p_2'$ ; 32
37 - 41 $p_3$ ; 37

End card $x$ (minus) in column 17

Operation (1) Read the program with the initial input key. It stops on 6, 3-13
(2) Run in the data, followed by the end card.
(3) When the end card has been read the program stops at 8, 0-0.
    Clearing the hooter and giving a single shot causes the program to return to 6, 3-13. Operation (2) may then be repeated.

Output Columns 17 - 21 reference number; sign in column 17
22 - 25 $d$ ; 22
26 - 29 $d'$ ; 26
30 - 33 $d_1$ ; 30
34 - 37 $d_2$ ; 34
38 - 41 $d_1'$ ; 38
42 - 45 $d_2$ ; 42

All the numbers (except the reference number, which is an integer) are given to 2D.

Parameters The $a$, $b$, $c$ are punched in binary (to 30 by) on card 22, rows $a$-$a_9$, and card 23, rows $b$-$b_9$, as follows:

Card 22, row $a_9$

$a_1$ now set as + 1797/5960
$3a$ $a_2$ + 1780/5960
$4a$ $a_3$ - 3577/5960
$5a$ $a_4$ + 0.0327
$6a$ $a_5$ + 1995/6480
$7a$ $a_6$ + 1780/6480
$8a$ $a_7$ - 3775/6480
$9a$ $a_8$ - 0.0824

Card 23, row $b_9$

$b_1$ + 3283/5960
$2a$ $b_2$ - 1660/5960
### Restrictions

1. $0 \leq \text{reference number} \leq 9999$
2. $|p_1| < 9999 \cdot 9$, and similarly for $p_2$, $p_3$, and $p_5$.
3. $|d| < 9 \cdot 99$, and similarly for $d'$, $\delta_1$, $\delta_2$, $\delta_3$, and $\delta_4$.
4. $N > 5$, where $N$ is the number of data cards.

### Failures

None.

### Time

N/29 minutes, where $N$ is the number of data cards.

### Method

- $p_1$ = pressure at Felixstowe
- $p_2$ = pressure at Tangmere
- $p_2'$ = pressure at Thorney Island
- $p_3$ = pressure at Calais

### Notes

1. In practice, the pressures $p_1$, $p_2$, $p_2'$, $p_3$ had a constant (900.0) subtracted before they were punched. This simplification does not affect the program provided $a_1 + a_2 + a_3 = 0$, and similarly for the $a'$, $b$ and $b'$.
2. Only 4 signs can be obtained comfortably on the Hollerith 805 lister, so in this case the signs of $d_1$ and $d_2$ (which have the same signs as $\delta_1$ and $\delta_2$ respectively) were not listed.
N.I.O. PROGRAM 10

Title Combining two series. Written for Job 538 (M3, RAB). RAB Program 569.

Code Deuce basic.

Purpose To difference corresponding terms of two series, and to sum the resulting series.

Order of cards Program (cards 0-18), data (in pairs of cards).

Parameter card None.

Data Columns 18 - 27 \( x_i \) or \( y_i \); sign in column 17.

There must be n pairs of cards.

End card None.

Operation (1) Read in the program with the initial input key.
(2) Run in the data, taking care not to split up pairs of data cards.

The last card alone will not restart the program.

Output Columns 18 - 27 \( w_i \); sign in column 17.

Parameters \( n, p \), on card number 01, row 9.

Restrictions (1) \( 0 < n \leq 499 \)
(2) \( |x_i| \leq 999,999, |y_i| \leq 999,999 \), for \( i = 1(1)n \).

See also Note (2).

Failures None.

Time About 2n seconds.

Method If there are n pairs of data cards, denoted by \( (x_1, y_1), (x_2, y_2) \ldots \)
\( (x_n, y_n) \ldots \), then the program computes \( W_i = \frac{1}{n}(x_i - y_i) \) for
\( i = 1(1)n \), and also \( \sum_{i=1}^{n} W_i \).

Notes (1) If \( W_i = \frac{1}{n}(x_i + y_i) \) is required instead of \( W_i \), the destination number
on card number 19, row 18 should be changed from 26 to 25. (This has
not been tested.)

(2) The program has been fully tested only for values within the limits
given under Restrictions, but it may well work for larger numbers.
Care must be taken to ensure that \( \sum_{i=1}^{n} W_i \) (in 17) does not overflow.

(3) Given two packs of cards, A and B, the corresponding terms of the
two series can be paired by placing the A pack in front of the B pack
and sorting the whole lot on the sorter (sorting on the card number and
starting at the units end of the card number as usual).
Title  Sixth differences.  RAE Program 54.9.

Code  Douce basis.

Purpose  To obtain central differences of a given numerical function.

Order of cards  Program (cards 0-31), parameter card, data, end card.

Parameter card  Y row: a one in each digit (or sign) position except the last
digit of a number, and zeros everywhere else.

Data  Two integers in the α-field. Both must have signs in the same column as
the first digit (see Parameters) and the first number must begin in
column 17. Nothing may be punched in the Y row of column 48.

End card  Yα row: P, only.

Operation  (1) Read the program with the initial input key. It stops on 3,0-16.
(2) Run in the parameter card, data and end card.
(3) When one difference table is complete, the hooter sounds and the
program returns to 3,0-16. The hooter may then be cleared and
operation (2) repeated.

Output  8 signed 8 digit integers per card. Non-significant zeros are
suppressed. The cards are in pairs as follows:—
Card 1  x  f(x)  space  δ^2  space  δ^4  space  δ^6
Card 2  space  space  δ  space  δ^3  space  δ^5  space

Card 1  columns 17 - 24  x  ;  sign in column 17
25 - 32  f(x)  ;  25
41 - 48  δ^2  ;  41
57 - 64  δ^4  ;  57
73 - 80  δ^6  ;  73

Card 2  columns 33 - 40  δ  ;  sign in column 33
49 - 56  δ^3  ;  49
65 - 72  δ^5  ;  65

The missing differences at the beginning and end of the table are
replaced by spaces.

Parameters  If the data has signs in the column before the first digit, then
the timing number on card 16, row 9, must be changed from 0 to δ.

Restrictions  (1) |f(x)| < 9999 9999. If any function value or difference
exceeds this limit, wrong results will be punched and values
depending on the incorrect one may be affected.
(2) There must be at least 6 data cards.

Failures  None.

Time  About n/15 minutes, where n is the number of input cards.

Method  The first integer on each data card is regarded as the argument, and the
second is differenced. Except at the beginning and end of the difference
table, the program alternately reads 1 card and punches 2 cards.

Notes  (1) The program makes use of two apparently unpublished subroutines both
due to G. Cork and J.M. Watt (MS, RAE) and originally designed for use
with stirling. Both subroutines have been slightly modified, and the
relevant parts of these versions are briefly described below. Together
they occupy DL 6, 7, 8.
(a) Binary to 4 digit binary coded decimal conversion.

First order, entry 82
Stores used 13 14 15 16 19 20 21
Entry link marker x divisor - - - - - x coded
Exit - - - - - - -

The marker consists of many 3 digit groups, the first one to be interpreted being in P_{20-32}. These groups are interpreted as follows:-

First group: 0 Code first digit
1 Code first digit suppressing non-significant zeros.

Other groups: 0,1 Code space
2,3 Multiply by 10 and code next digit.

If the value of any of these groups is even, non-significant zeros will not be suppressed from that point.

The divisor = 16N-1 where N is the number of units corresponding to a most significant digit of 1. For example, if the number to be coded is x, and x = 432, then N = 100.

If x has 8 digits or less the coded result is entirely in 212.

(b) Punch decimal from 4-digit binary.

First order, entry 82
Stores used 13 14 15 16 17, 17, 17, 18 18 21 21
Entry link - - - 0 ; X g - - 4 digit binary
Exit - - - - - - - - - - - 4 digit binary

If any digit is d, then if d = 0, 1, 2, ..., 9, an appropriate hole is punched in that column and if d > 10, the corresponding column is left blank.

The layout of binary coded digits ready for punching in columns 17-32 is as follows:-

<table>
<thead>
<tr>
<th>DL 10</th>
<th>P_{1-4}</th>
<th>P_{5-8}</th>
<th>P_{9-12}</th>
<th>P_{13-16}</th>
<th>P_{17-20}</th>
<th>P_{21-24}</th>
<th>P_{25-28}</th>
<th>P_{29-32}</th>
</tr>
</thead>
<tbody>
<tr>
<td>me 24</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>25 32</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td>26</td>
<td>25</td>
</tr>
<tr>
<td>26 40</td>
<td>40</td>
<td>39</td>
<td>38</td>
<td>37</td>
<td>36</td>
<td>35</td>
<td>34</td>
<td>33</td>
</tr>
<tr>
<td>27 43</td>
<td>43</td>
<td>42</td>
<td>41</td>
<td>40</td>
<td>39</td>
<td>38</td>
<td>37</td>
<td>36</td>
</tr>
<tr>
<td>28 56</td>
<td>56</td>
<td>55</td>
<td>54</td>
<td>53</td>
<td>52</td>
<td>51</td>
<td>50</td>
<td>49</td>
</tr>
<tr>
<td>29 64</td>
<td>64</td>
<td>63</td>
<td>62</td>
<td>61</td>
<td>60</td>
<td>59</td>
<td>58</td>
<td>57</td>
</tr>
<tr>
<td>30 72</td>
<td>72</td>
<td>71</td>
<td>70</td>
<td>69</td>
<td>68</td>
<td>67</td>
<td>66</td>
<td>65</td>
</tr>
<tr>
<td>31 80</td>
<td>80</td>
<td>79</td>
<td>78</td>
<td>77</td>
<td>76</td>
<td>75</td>
<td>74</td>
<td>73</td>
</tr>
</tbody>
</table>
Title: Slope correction to waveheights. Written for job 596 (MS, RAE).

Purpose: To modify a set of data, using two related sets of data and suitable constants.

Order of cards: Program (cards 0-40), parameter card, data.

Parameter card: Columns
- 17 - 20: n
- 21 - 24: \( a_1 \)
- 25 - 28: \( 10^7 a_2 \)
- 29 - 32: \( 10^7 a_3 \)
- 33 - 48: zeros

Data: 8 positive 4 digit numbers per card in the a-field. If \( n \) is not a multiple of 8, the last card must be filled with zeros.

End card: None.

Operation: (1) Read the program with the initial input key. The program stops on 3, 16-1.
(2) Run in the parameter card, followed by R, P, Q in that order.
(3) When one calculation is finished, the hooter sounds and the program stops on 3, 16-1. The hooter may then be cleared and operation (2) repeated.

Output: \( Q_r \), \( r = 0(1) n-1 \), punched as 8 positive 4 digit numbers per card in the a-field. No signs are punched. If the number of data cards is not a multiple of 4, up to 3 superfluous cards containing zeros will be punched.

Parameters: None.

Restrictions: (1) Each number \( x \) in the data must satisfy \( 0 < x < 999 \).
(2) \( 0 < n < 2560 \).
(3) \( Q_r - \bar{Q} + a_1 > 0 \) for \( 0 < r < n-1 \).
(4) \( 0 < a_2 (P_r - \bar{P})^2 + a_3 (R_r - \bar{R})^2 < 214 \) for \( 0 < r < n-1 \).
(5) \( 0 < a_1 < 9999 \), and similarly for \( 10^7 a_2 \) and \( 10^7 a_3 \).

Failures: None.

Time: About 3n/500 minutes.

Method: \( Q_r = Q_r - (Q_r - \bar{Q} + a_1) [a_2 (P_r - \bar{P})^2 + a_3 (R_r - \bar{R})^2] \) for \( 0 < r < n-1 \).

Notes: (1) The program was originally written for data obtained from the pitch and roll buoy. The aim was to obtain the vertical acceleration \( Q \) from the pitch \( P \), roll \( R \) and acceleration \( Q \).
(2) At the end of the program DL 12 contains the following integers, apart from constants and working space:
- 0 n
- 1 a_1
2  \(10^7c_2\)
3  \(10^7c_3\)
4  number of cards (either \(\left\lfloor \frac{n}{5} \right\rfloor\) or \(\left\lceil \frac{n}{5} \right\rceil + 1\), depending on \(n\))
9  \(\bar{J}\)
10  \(\bar{F}\)
11  \(\bar{R}\)
Title Correlation.

Code Deuce basic.

Purpose To find the autocorrelation coefficients of a time series or the cross correlation coefficients of two series.

Order of cards Program (cards 0-49), parameter card, series 1, series 2 (if required).

Parameter card Columns 17 - 20 \( n \)
21 - 24 \( K_1 \)
25 - 28 \( K_2 \)
29 - 32 \( L_1 \)
33 - 36 \( L_2 \)
41 - 44 \( p \)
45 - 48 must be left blank.

If any of \( K_1, K_2, L_1, L_2 \) are omitted (that is, the corresponding columns are blank), the appropriate section of the program is omitted. If both \( L_1 \) and \( L_2 \) are omitted, only one series is read and \( K_1 \) autocorrelation coefficients are found. If \( p \) is omitted, no blank cards are inserted.

Data The \( n \) terms of a series are punched as 8 positive 4 digit integers per card in the \( a \)-field. Series 2, if present, must have the same number of terms (\( n \)) as series 1.

End card None.

Operation (1) Read the program with the initial input key.
(2) Run in the parameter card, followed by series 1.
(3) Run in series 2, if present.
(4) When one run is complete, (2) and (3) may be repeated.

Output There are two types of output.

(1) Autocorrelation. Card 1 Mean
Card 2 Variance
Card 3 Blank
Card 4 onwards Correlation coefficients.

(2) Cross correlation. Card 1 Blank
Card 2 onwards Correlation coefficients.

The correlation coefficients are separated by a blank card after every \( p \) output cards, and there is a blank card at the end.

There are two types of card layout.

(1) Columns 18 - 26 \( a \); sign in column 17
34 - 36 \( b \); 33
47 - 48 Marker

If either the mean or the variance is denoted by \( c \), then \( c = a \cdot 10^{-b} \).

(2) Columns 18 - 26 \( r_j \); sign in column 17
34 - 36 \( j \); 33
47 - 48 Marker

Here \( r_j \) is the correlation coefficient for lag \( j \).

Both \( a \) and \( r_j \) are punched to 8D. The marker is defined under Method.

Parameters None.
Restrictions
(1) \(0 < n \leq 3488\)
(2) \(0 < K_1, K_2 \leq 159\)
(3) \(0 < L_1, L_2 \leq 127\)
(4) \(0 < x_i, y_i \leq 999, i = 0(1)n-1\).

Failures
(1) If any of Restrictions (1)-(3) are violated, the program hoots and stops on 1, 1-1. The output staticiser lights indicate the mistake, as follows:

- \(P_1\) : \(n > 3488\)
- \(P_{17}\) : \(K_1 > 159\) and/or \(K_2 > 159\)
- \(P_{1-3}\) : \(L_1 > 127\) and/or \(L_2 > 127\)

(2) There are also indications of machine failures. The program hoots and stops on 1, 1-1 (excluding the above), 6, 2-2 or 5, 3-3. A single shot usually repeats the appropriate section, but it is safer to restart the run completely.

Time
About \(3n/30\) seconds to read 1 series, and \(nV/210\) seconds to calculate and punch 1 set of correlation coefficients, where \(N\) is the number of lags.

Method
(1) If we are given 2 series, each with \(n\) terms, denoted by
\[x_0, x_1, \ldots, x_{n-1} \text{ (series 1)}\]
\[y_0, y_1, \ldots, y_{n-1} \text{ (series 2)}\]
then the cross correlation coefficient is defined by
\[r_{j} = \frac{c_{j}^2}{AB},\]
where
\[(n-j)c^2 = (n-j) \sum_{i=1}^{n-j} x_i y_{i+j} - \left[\sum_{i=1}^{n-j} x_i\right]\left[\sum_{j=1}^{n} y_j\right],\]
\[(n-j)A^2 = (n-j) \sum_{i=1}^{n-j} x_i^2 - \left[\sum_{i=1}^{n-j} x_i\right]^2,\]
\[(n-j)B^2 = (n-j) \sum_{j=1}^{n} y_j^2 - \left[\sum_{j=1}^{n} y_j\right]^2,\]
and where \(j > 0\) is the lag number.
The autocorrelation coefficient is obtained when \(x_i = y_i\) for all \(i\) in the above formulae. In this case we may also define the mean \(\mu\) and the variance \(\sigma^2\) as follows:
\[n \mu = \sum_{i=1}^{n} x_i,\]
\[(n-1) \sigma^2 = \sum_{i=1}^{n} x_i^2 - n \mu^2,\]

(2) The numbers \(K\) and \(L\) on the parameter card denote the number of lags in the 4 possible cases.
K_1 : series 1 autocorrelation, with marker 1,1
K_2 : series 2 autocorrelation, with marker 2,2.
L_1 : cross correlation, series 1 leading, with marker 1,2
L_2 : cross correlation, series 2 leading, with marker 2,1

Also p = punch spacing.

"Series 1 leading" means that x_{n-i} and y_i, i = 0(1)j-1, are excluded from the calculation of r_j.

Notes

There are two simple checks on the output.

(1) The first autocorrelation coefficient (that is, the one with zero lag) is unity.
(2) If r_0 and r_0' denote the cross correlation coefficients with zero lag for series 1 leading and series 2 leading respectively, then r_0 = -r_0'.
R.A.E. PROGRAM 300

Title Fourier analysis (periodogram)

Code Deuce basic.

Purpose To calculate the harmonic components \( A_s, B_s \) for \( s = a(b)c \).

Order of cards Program (cards 0-28), parameter card, data.

Parameter card

<table>
<thead>
<tr>
<th>Columns</th>
<th>17 - 20</th>
<th>21 - 24</th>
<th>25 - 28</th>
<th>29 - 32</th>
<th>33 - 36</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( n )</td>
<td>zeros</td>
<td>( a )</td>
<td>( b )</td>
<td>( c )</td>
</tr>
</tbody>
</table>

Data The \( n \) values of \( y_r \) are punched as 8 signed 4-digit integers per card, in the \( a \)-field.

End card None.

Operation

1. Read the program with the initial input key.
2. Run in the parameter card, followed by the data.
When a run is complete, (2) may be repeated.

Output

<table>
<thead>
<tr>
<th>Columns</th>
<th>17 - 20</th>
<th>21 - 30</th>
<th>31 - 38</th>
<th>39 - 46</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( s )</td>
<td>( I_s )</td>
<td>( A_s )</td>
<td>( B_s )</td>
</tr>
</tbody>
</table>

All these numbers are integers.

Parameters None.

Restrictions

1. \( 0 \leq s \leq 2047 \)
2. \( n M^2 \leq 1.08 \times 10^9 \), where \( |y_r| \leq M, r = O(1)n^{-1} \)
3. \( n^2 M \leq 2.14 \times 10^9 \)

Failures

1. \( s \) is punched incorrectly for \( s > 2047 \), but the other quantities may be correct.
2. If Restriction (3) is violated, the program might punch the corresponding \( I_s \) as -1 and sound the hooter. Rubbish may be punched with no indication.

Time About \( nN/200 \) seconds, where \( N \) is the number of values of \( s \).

Method

\[
A_0 = \sum_{r=0}^{n-1} y_r, \quad B_0 = 0 \quad \text{when} \ s = 0
\]

\[
A_s = \sum_{r=0}^{n-1} y_r \cos \left( \frac{2\pi rs}{n} \right) \quad \text{when} \ s > 0
\]

\[
B_s = \sum_{r=0}^{n-1} y_r \sin \left( \frac{2\pi rs}{n} \right) \quad \text{when} \ s > 0
\]

\[
I_s = \frac{2}{n} (A_s^2 + B_s^2) \quad \text{for all} \ s,
\]

where \( s \) is an integer.

Notes The method of calculation is described in The Computer Journal, 1, 4, p. 162.
R.A.E. PROGRAM 528

Title Fourier analysis (periodogram)

Code Deuce basic.

Purpose To calculate harmonic components $A_n$, $B_n$ for $s = 0(a_1)b_1(a_2)b_2(a_3)b_3$

Order of cards Program (cards 0-28), parameter card, data.

Parameter cards

<table>
<thead>
<tr>
<th>Column</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>17 - 20</td>
<td>n</td>
</tr>
<tr>
<td>21 - 24</td>
<td>k</td>
</tr>
<tr>
<td>25 - 28</td>
<td>$a_i$</td>
</tr>
<tr>
<td>29 - 32</td>
<td>$b_i$</td>
</tr>
<tr>
<td>33 - 36</td>
<td>$a_j$</td>
</tr>
<tr>
<td>37 - 40</td>
<td>$b_j$</td>
</tr>
<tr>
<td>41 - 44</td>
<td>$a_k$</td>
</tr>
<tr>
<td>45 - 48</td>
<td>$b_k$</td>
</tr>
</tbody>
</table>

No column may be left blank, so if $a_i$, $b_i$, $a_j$, $b_j$ are not needed, zeros should be punched in their place.

Data The n values of $y_t$ are punched as 8 signed 4 digit integers per card, in the $n$-field.

End card None.

Operation

(1) Read the program with the initial input key.
(2) Run in the parameter card, followed by the data.
When a run is complete, (2) may be repeated.

Output Columns 18 - 21 $s$ ; sign in column 17

<table>
<thead>
<tr>
<th>Column</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 - 30</td>
<td>$A_s$</td>
</tr>
<tr>
<td>33 - 39</td>
<td>$B_s$</td>
</tr>
<tr>
<td>42 - 48</td>
<td>$I_s$</td>
</tr>
</tbody>
</table>

There are three more decimal places in $A_s$, $B_s$, and $I_s$ than in $y_t$.

Parameters None.

Restrictions $n^3 M < 2 \cdot 10^9$, where $|y_t| < M$, $r = 0(1)n-1$.

Failures None.

Time About $n/200$ seconds, where $N$ is the number of values of $s$.

Method

$$A_s = \frac{1}{n} \sum_{r=0}^{n-1} y_r, \quad B_s = 0 \quad \text{when } s = 0$$

$$B_s = \frac{2}{n} \sum_{r=0}^{n-1} y_r \cos \left( \frac{r2\pi}{k} \right) \quad \text{when } s > 0$$

$$I_s = \sqrt{A^2 + B^2} \quad \text{for all } s$$

where $s$ is an integer. If $0 < c < 13$ is an integer, then $k = 2^c$.

Notes The method of calculation is described in The Computer Journal, 1, 4, p. 162.