M- 2054

No type written
copies made.

Rough Draft available
in this copy. (M-2000)

6673
Memorandum M-2000

Air Traffic Control Project
Servomechanisms Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts

SUBJECT: MAINTAINING PRIVATE LINE COMMUNICATION (PART I)

To: 6673 Group

From: W. Gordon Welchman

Date: April 27, 1949

Object

1. The object of this memorandum is to provide a starting point in the study of the problem of maintaining private line communications. The program has been hurriedly prepared, without much attempt at refinement, in order to obtain some idea of the magnitude of the job and in order to have something with which alternative methods can be compared. The memorandum is in two parts so that the reader can look at the data storage on the last page of the first part while he is also looking at the program which is given in the second part.

The Problem

2. I assume that private line communication for control purposes has been established with a number of aircraft (I shall use the abbreviation a/c) which will be said to be "in the system". The problem is to maintain contact with these a/c. In this note I am not considering how a/c are brought into the system or how they leave.

3. Communication is by means of a directional beam which rotates at a constant speed ω which I assume to be 2 seconds per revolution. Every time contact is made with a particular a/c in the system the computer has both to estimate at what azimuth contact can be made again during the next sweep of the beam and also to store any information that may be needed for this future contact.

General Assumptions

4. It is assumed that each a/c has an identity code by which it can be distinguished from other a/c that may be in the beam at the same time. Positional data are assumed to be available in the a/c so that the computer can obtain them by interrogation. At

each contact therefore the computer obtains positional data for an a/c, does the necessary calculations, and sends control signals to the a/c. We are not at present concerned with the control computation but only with the computation that is needed to maintain contact.

5. We assume 100 a/c in the system at distances of 10 to 100 miles. We must allow for a/c speeds from 100 to 500 miles per hour. We can neglect the tangential acceleration of an a/c and assume that the rate of turn will not be more than 5 degrees a second.

6. We shall assume a beam width of between 2 and 3 degrees and define within the beam a leading edge and a trailing edge separated by 2 degrees. We shall attempt to make contact on the leading edge and it will be assumed that this leading edge is sufficiently far inside the beam to allow for errors in prediction. At 2 seconds per revolution, the leading edge sweeps 0.18 degrees per millisecond.

Outline of Method

7. I have simplified the problem by assuming a blind sector of 30 degrees. I assume that no a/c in the system will ever have an azimuth between $-\frac{\pi}{12}$ and $+\frac{\pi}{12}$. I use ϕ to denote the measured azimuth of the leading edge of the beam and assume that this can be obtained at any time from an analog input device by giving the order rf 1. I use θ to denote the azimuth of the a/c and φ to denote the predicted azimuth of the leading edge of the beam at which it is expected to hit the a/c. To save writing I take:

$$\phi' = \phi + 2\pi, \qquad \varphi' = \varphi + 2\pi$$

As a result of the blind sector assumption, θ, ϕ, and φ will all lie between $\frac{\pi}{12}$ and $\frac{23\pi}{12}$. In considering any particular contact I use subscripts 0 and 1 to indicate data relating to the previous (actual) contact and the next (predicted) contact respectively.

8. The necessary information about the a/c in the system is stored in four sets of B-class registers which will be called the a/c blocks. There are $2^7 = 128$ registers in each set and they are denoted by B(c+x), B(d+x), B(e+x), and B(f+x), where x runs from 0 to $2^7-1 = 127$.

9. We will suppose that the computer is dealing with some main program (MP) and that at convenient intervals in MP, at times when the content of the accumulator is not required, a test is applied by an sp order which leads to subprogram A (SPA). We assume that the sp order puts the address of the next order of MP in AR and that this address can be read directly from AR into storage by the order ta.

10. Consider the state of affairs at the beginning of the sweep, that is, when $\phi = \frac{\pi}{12}$. The predicted ϕ' for all a/c in the system (suppose the number of a/c is n) are arranged in ascending order in n consecutive B-class registers running from $B(c+x)$ to $B(c+\overline{x+n-1})$ and the related data are in $B(d+x)$, $B(e+x)$, $B(f+x)$, to $B(d+\overline{x+n-1})$, $B(e+\overline{x+n-1})$, $B(f+\overline{x+n-1})$ where the bar denotes modulo 2^7.

11. The test consists in finding out whether it is time to make contact with the a/c whose ϕ' is in $B(c+x)$. When contact time is reached, the computer switches to SPB, the prediction subprogram. The predicted ϕ' for contact on the next sweep is put in $B(c+y)$ where $y = \overline{x+n}$ and the related data are put in $B(d+y)$, $B(e+y)$, $B(f+y)$. Both x and y are increased by one modulo 2^7 to prepare for the next a/c. This next a/c is dealt with at once if its contact time has been reached, otherwise the computer returns to MP.

12. At the end of the sweep when $\phi = \frac{23\pi}{12}$, the data required for the next sweep are again stored in sets of n consecutive registers in the a/c blocks, the new initial value of x being $\overline{x_0+n}$ where x_0 is the old initial value. The ϕ, however, will not necessarily be in ascending order since some a/c may have changed places. Consequently, the computer goes to SPC, the shuffling subprogram, which rearranges the ϕ and the related data in ascending order for the next sweep before returning to MP.

<u>Method of Prediction</u>

13. In predicting the ϕ_1' for the next sweep, I have neglected the angular acceleration of the a/c. Taking:

t_0 = time from previous contact to current contact

t_1 = time from current contact to predicted contact we have the equations

$$\omega t_0 = \phi' - \phi_0$$

$$\dot{\theta} t_0 = \theta - \theta_0$$

$$\omega t_1 = \phi_1' - \phi$$

$$\dot{\theta} t_1 = \phi_1 - \theta = \phi_1' - \theta - 2\pi$$

which give

$$\phi_1' = \frac{(\theta + 2\pi)(\phi' - \phi_0) - \phi\,(\theta - \theta_0)}{(\phi' - \phi_0) - (\theta - \theta_0)}$$

Method of Shuffling

14. In rearranging the predicted ϕ we can take advantage of the fact that they will be almost in ascending order. I have, therefore, done the shuffling by running through the consecutive pairs of ϕ, interchanging a pair and the related data when they are not in order. The shuffle is complete when a run through all consecutive pairs produces no interchange. Very few runs should be required.

15. At the beginning of a shuffle, values of x and y will be in storage. To deal with the difficulty of working modulo 2^7 I have introduced two new numbers, x' and y'. Of these, y' remains fixed during the shuffle and is equal to y or y + 2^7, according as y is greater than or less than x. The effect of this is to make y' = x+n. The other number, x', is initially equal to x and varies between x and y'- 1 as the shuffle proceeds. When two consecutive ϕ are being compared, they will be denoted by ϕ_j and ϕ_{j+1}, the position of the former being determined by the current value of x'. The same subscripts j and j+1 are used for the data corresponding to ϕ_j and ϕ_{j+1}.

Aircraft Data

16. The quantity i that is stored in B(d+x) is something that will identify an a/c in the system. It may be the identity code that will be used in communications or, alternatively, a register address that will enable the computer to find information about the a/c. In the program the data shown in B(c+x), B(d+x), B(e+x), B(f+x) were put there at the last contact while the data shown in B(c+y), B(d+y), B(e+y), B(f+y) are put there during the current contact.

Requirements of Storage and Time

17. The storage requirements are as follows:

SPA	12
SPB	65
SPC	89
B-class	527
C-class	8
Total	701

In estimating time the computer is assumed to average 20 µseconds per operation. When no action follows, SPA takes about one-quarter of a millisecond. To deal with prediction for one a/c SPB takes about one and one-half milliseconds which is equivalent to 0.27 degrees of sweep. In the shuffling process of SPC, each run through takes about 2 milliseconds and each interchange about 1 millisecond. Thus 5 runs and 40 interchanges, which is more than one would normally expect, would require 50 milliseconds which is equivalent to about 9 degrees of sweep.

The Program

18. In the program it should be understood that π, wherever it occurs, and all the angles θ, ϕ, ψ, are multiplied by a suitable scale factor that is not shown.

19. By reserving B4, Ab1, and Ac1, I have made it easy to change the program so that after SPB or SPC a return is made to SPA rather than the MP. However, as the program is written at present both SPB and SPC lead back to MP.

20. The cp order is assumed to be made effective by a negative sign digit. It seemed reasonable that we should use work on the Air Traffic problem, which is likely to be somewhat segregated from other work in Project Whirlwind, as a field in which to try out cp- with a view to deciding whether it is more convenient to use than cp+.

Questions for Further Study

21. The following questions seem to arise directly from this memorandum:

(a) What risk of losing contact is involved in the simple method of prediction that I have adopted? What errors in the measurement of θ and ϕ can be tolerated?

(b) What risk is there that the clustering of a/c at any one azimuth will not give the computer enough time to deal with them before the trailing edge of the beam has passed?

(c) Does this code give a reasonable measure of the storage and time requirements or could the code be considerably shortened?

(d) How will a/c be brought into the system and taken out of it?

(e) In the shuffling process would it be better to take each a/c in turn and move it up the line until no a/c ahead of it has a greater predicted ϕ? (This method should simplify the introduction and removal of a/c. Further, it would make it possible to do the shuffling intermittently during the sweep.)

(f) Could we get rid of the blind sector without lengthening the program unduly?

(g) Alternatively, would it be better to use two sectors, each of 180 degrees, passing a/c from one sector to the other when necessary, and doing the shuffling for each sector while the other is being swept?

(h) How great an improvement in the speed and flexibility of the system could be achieved if there were 32 final beams each covering 11 1/4 degrees instead of one rotating beam? It would then be possible to switch from one beam to the next as soon as all a/c in the first beam had been dealt with. (This idea may well prove to be impractical, but it is worth finding out whether in theory it would provide a more satisfactory system.)

Constant Data Storage

C 1	2π	C 5	$(e-d) \times 2^{-15}$
2	2^{-15}	6	$(f-e) \times 2^{-15}$
3	$c \times 2^{-15}$	7	$2^{7} \times 2^{-15}$
4	$(d-c) \times 2^{-15}$	8	$\frac{23\pi}{12}$

Transient Data Storage

B 1 Next ϕ at which a/c should be interrogated.

2 Latest observed ϕ, azimuth of leading edge of beam.

3 Address for return to MP.

4 Address for return to SPA. (Not used in present form of code.)

5	θ	B 11	ϕ'_j
6	$\theta - \theta_o$	12	i_j
also	$(\phi'-\phi_o)-(\theta-\theta_o)$	13	$(\theta_o+2\pi)_j$
7	$\phi(\theta-\theta_o)$	14	$(\phi'-\phi_o)_j$
8	$x \cdot 2^{-15}$	15	Interchange count (initially -0)
9	$y \cdot 2^{-15}$	16	$x' \cdot 2^{-15}$
10	not used	17	$y' \cdot 2^{-15}$

B c+x	ϕ'	B c+y	ϕ'_1
d+x	i	d+y	i
e+x	$\theta_o+2\pi$	e+y	$\theta+2\pi$
f+x	$\phi'-\phi_o$ (changed to $\phi_o-\phi'$)	f+y	$\phi'_1-\phi$

$x,y = 0,1,2,\ldots 2^{7}-1$

6673
Memorandum M-2000

Air Traffic Control Project
Servomechanisms Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts

SUBJECT: MAINTAINING PRIVATE LINE COMMUNICATION (PART II)

To: 6673 Group

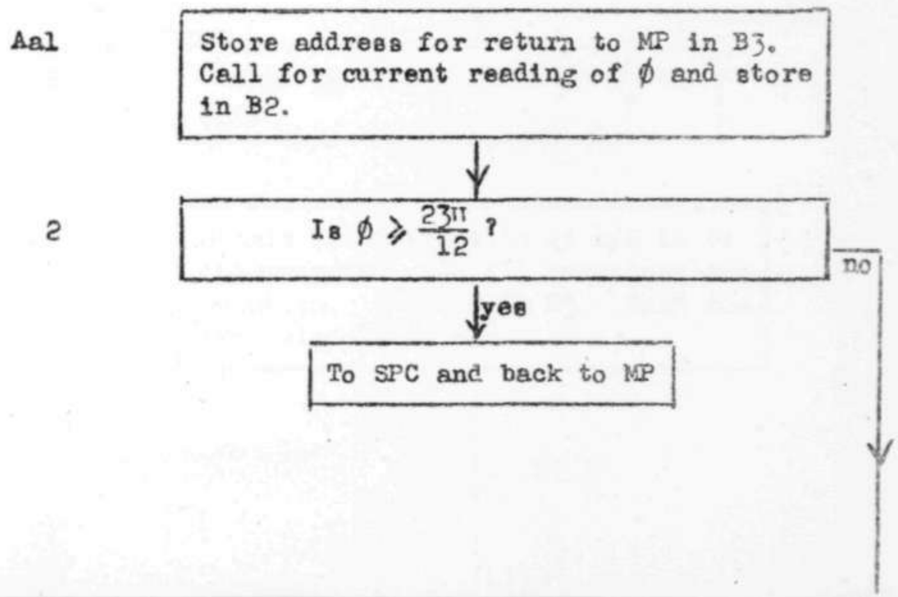
From: W. Gordon Welchman

Date: April 27, 1949

THIS CODE USES cp-.

Subprogram A. (SPA) The Test Program

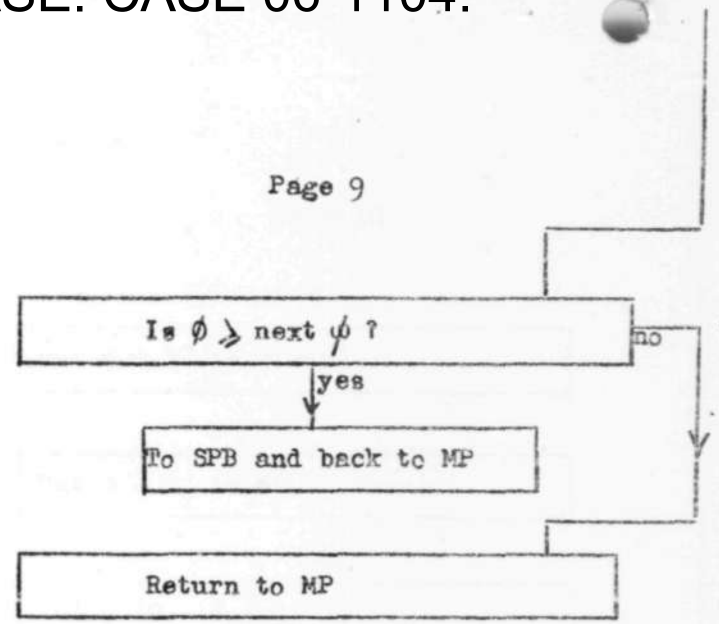
```
Aa1.1  rf  1
    2  ta  B3  (Transfer direct from AR)
    3  rd  B2   Puts φ in B2.

Aa2.1  ca  C8    AC  23π/12
    2  su  B2        23π/12 - φ
    3  cp  Ac2.1
```



```
Aa3.1  ca  B1      φ
    2  su  B2      φ - φ
    3  cp  Ab2.1

Aa4.1  ca  B3      Address for return to MP
    2  td  Aa4.3
    3  sp  --
```

3

4



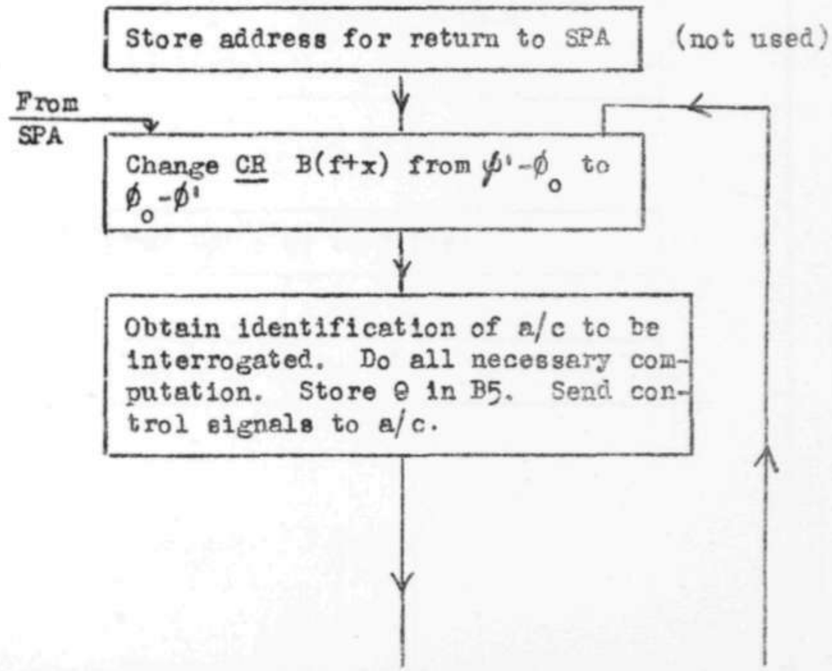
Subprogram B. (SPB) The Prediction Program

```
Ab1.1  ta  B4      (Transfer direct from AR)

Ab2.1  su  B(f+x)  φ-φ-(φ'-φ_o)=φ_o-φ'
    2  ts  B(f+x)

Ab3.1  ca  B(d+x)      i
       Remainder omitted for the present.
```

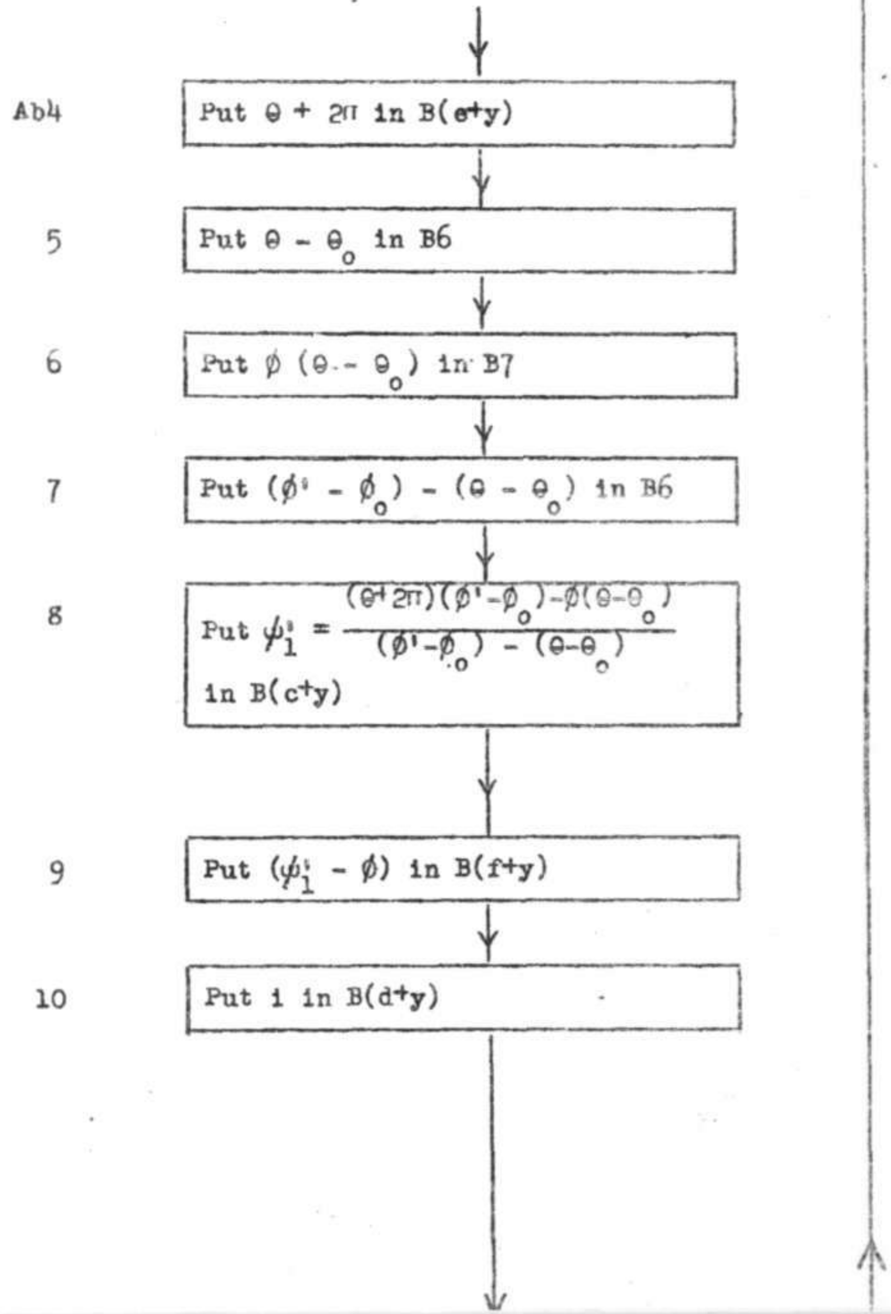
Ab1

2

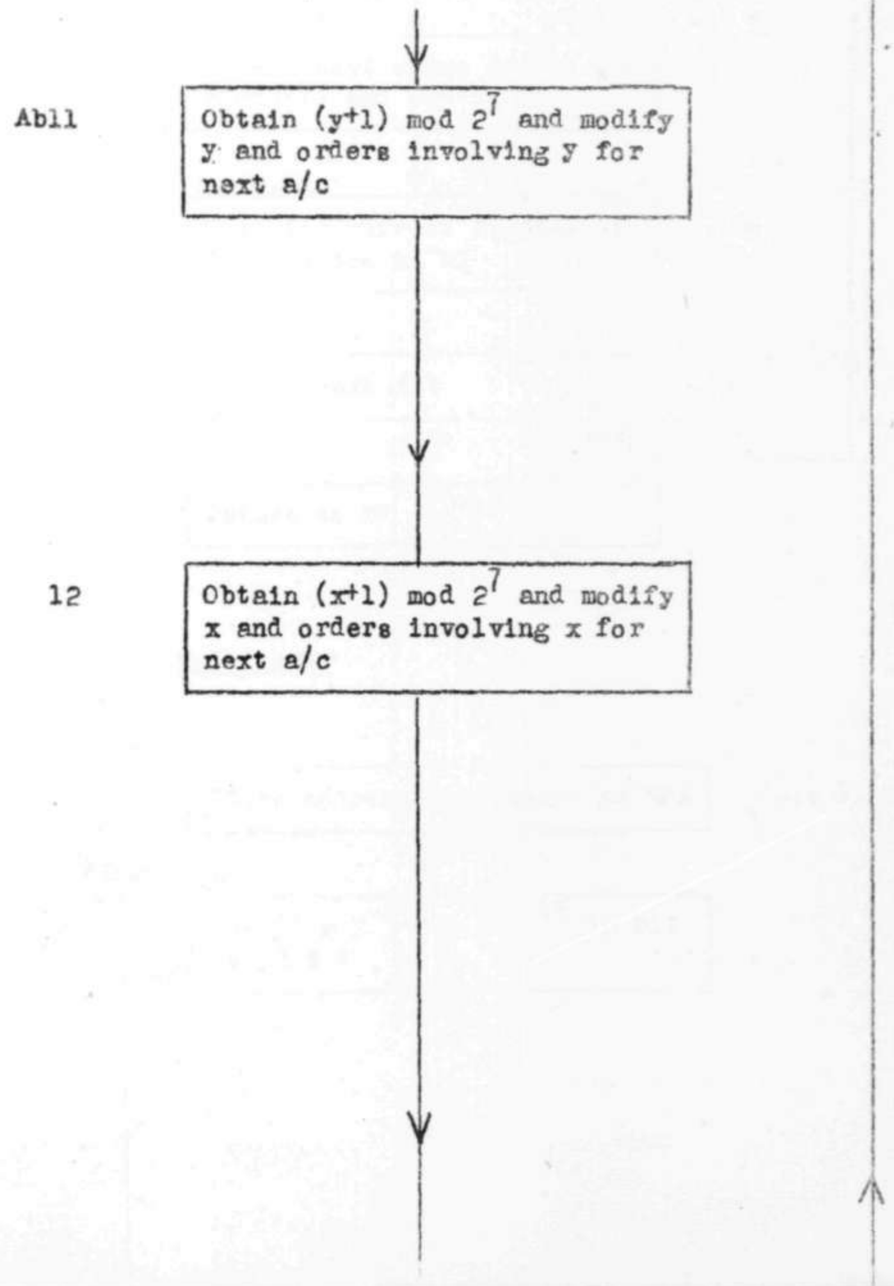
3



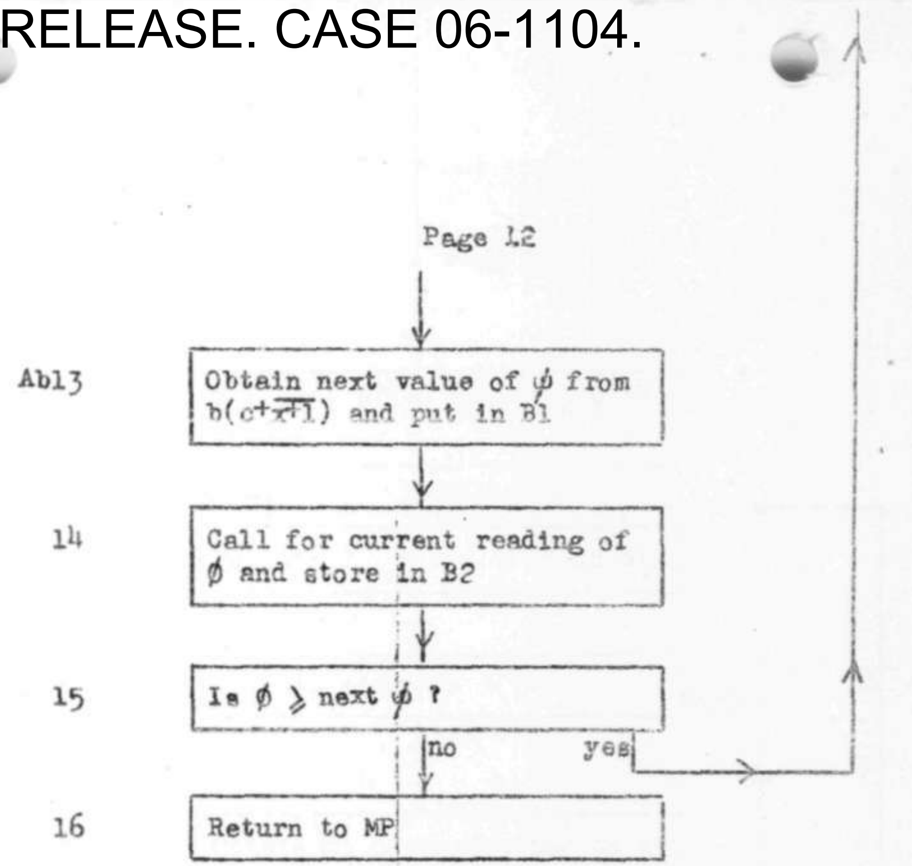
Ab4.1	ca	B5	θ
2	ad	C1	$\theta + 2\pi$
3	ts	Be^+y	
Ab5.1	su	Be^+x	$\theta - \theta_o$
2	ts	B6	
Ab6.1	mr	B2	$\phi(\theta - \theta_o)$
2	ts	B7	
Ab7.1	cs	B(f^+x)	$\phi' - \phi$
2	su	B6	$(\phi' - \phi_o^o) - (\theta - \theta_o)$
3	ts	B6	
Ab8.1	cs	B(f^+x)	$\phi' - \phi_o$
2	mr	B(e^+y)	$(\theta + 2\pi)(\phi' - \phi_o)$
3	su	B7	$(\theta + 2\pi)(\phi' - \phi_o) - \phi(\theta - \theta_o)$
4	dv	B6	
5	sl	15	ϕ_1'
6	ts	B(c^+y)	
Ab9.1	su	B2	$\phi_1' - \phi$
2	ts	B(f^+y)	
Ab10.1	ca	B(d^+x)	i
2	ts	B(d^+y)	



Ab11.1	ca	B9	$y \times 2^{-15}$
2	ad	C2	$(y+1) \times 2^{-15}$
3	sl	8	
4	sr	8	$\overline{y+1} \times 2^{-15}$
5	ts	B9	
6	ad	C3	$(c+\overline{y+1}) \times 2^{-15}$
7	td	Ab8.6	
8	ad	C4	$(d+\overline{y+1}) \times 2^{-15}$
9	td	Ab10.2	
10	ad	C5	$(e+y+1) \times 2^{-15}$
11	td	Ab4.3	
12	td	Ab8.2	
13	ad	C6	$(f+\overline{y+1}) \times 2^{-15}$
14	td	Ab9.2	
Ab12.1	ca	B8	$x \cdot 2^{-15}$
2	ad	C2	$(x+1) \times 2^{-15}$
3	sl	8	
4	sr	8	$\overline{x+1} \times 2^{-15}$
5	ts	B8	
6	ad	C3	$(c+\overline{x+1}) \times 2^{-15}$
7	td	Ab13.1	
8	ad	C4	$(d+\overline{x+1}) \times 2^{-15}$
9	td	Ab3.1	
10	td	Ab10.1	
11	ad	C5	$(e+\overline{x+1}) \times 2^{-15}$
12	td	Ab5.1	
13	ad	C6	$(f+\overline{x+1}) \times 2^{-15}$
14	td	Ab2.1	
15	td	Ab2.2	
16	td	Ab7.1	
17	td	Ab8.1	

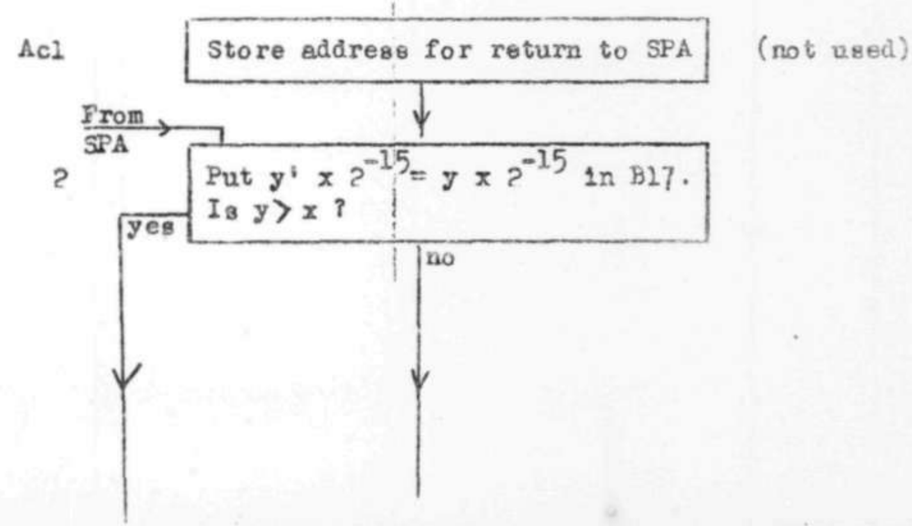


```
Ab13.1  ca  --       Next ϕ'
     2  su  C1       Next ϕ
     3  ts  B1

Ab14.1  rf   1
     2  rd  B2       B2 contains new ∅

Ab15.1  ca  B1       New ϕ
     2  su  B2       New (ϕ - ∅)
     3  cp  Ab2.1

Ab16.1  ca  B3       Address for return to MP
     2  td  Ab16.3
     3  sp  --
```

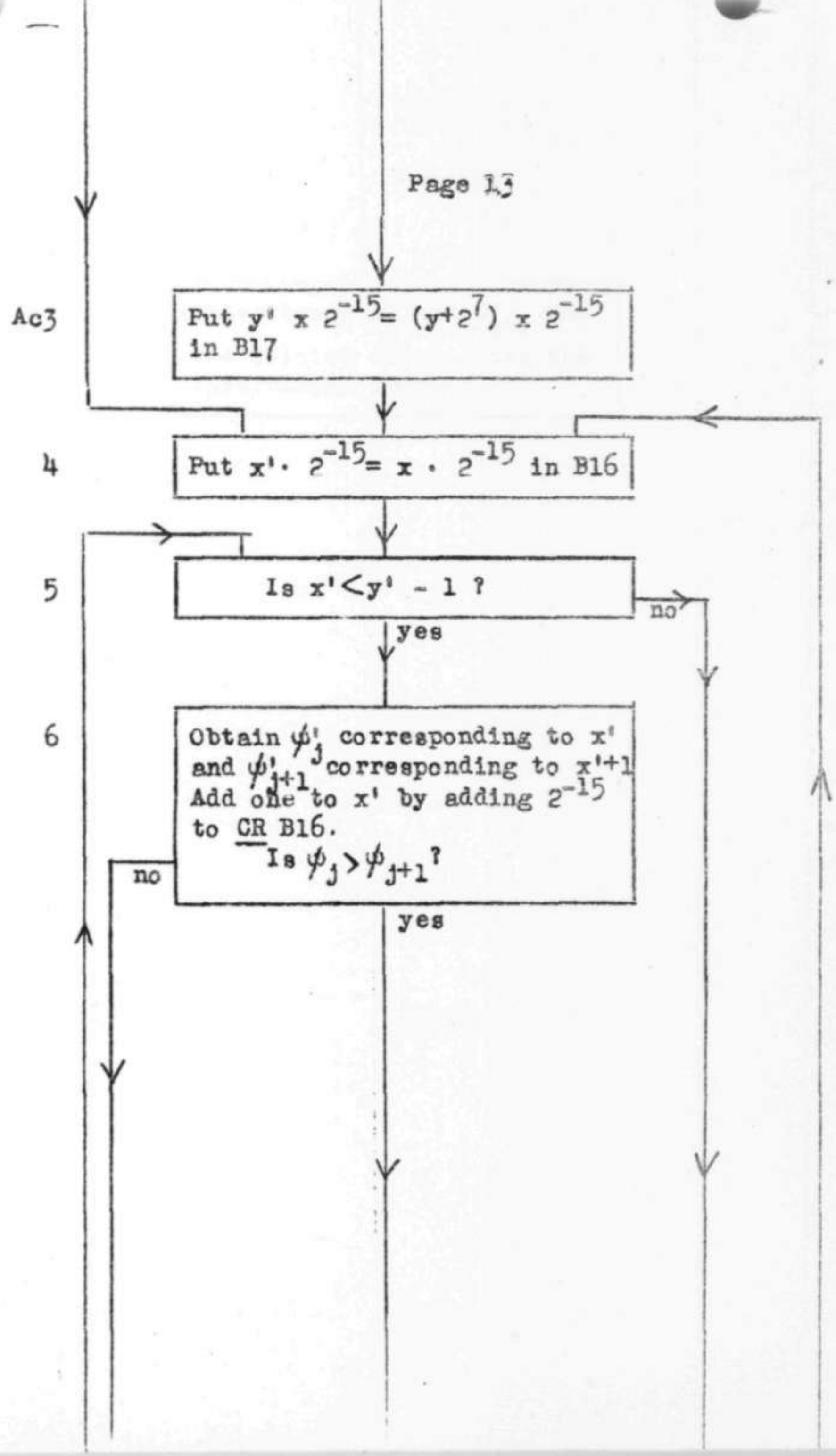


<u>Subprogram C. (SPC) The Shuffling Program</u>

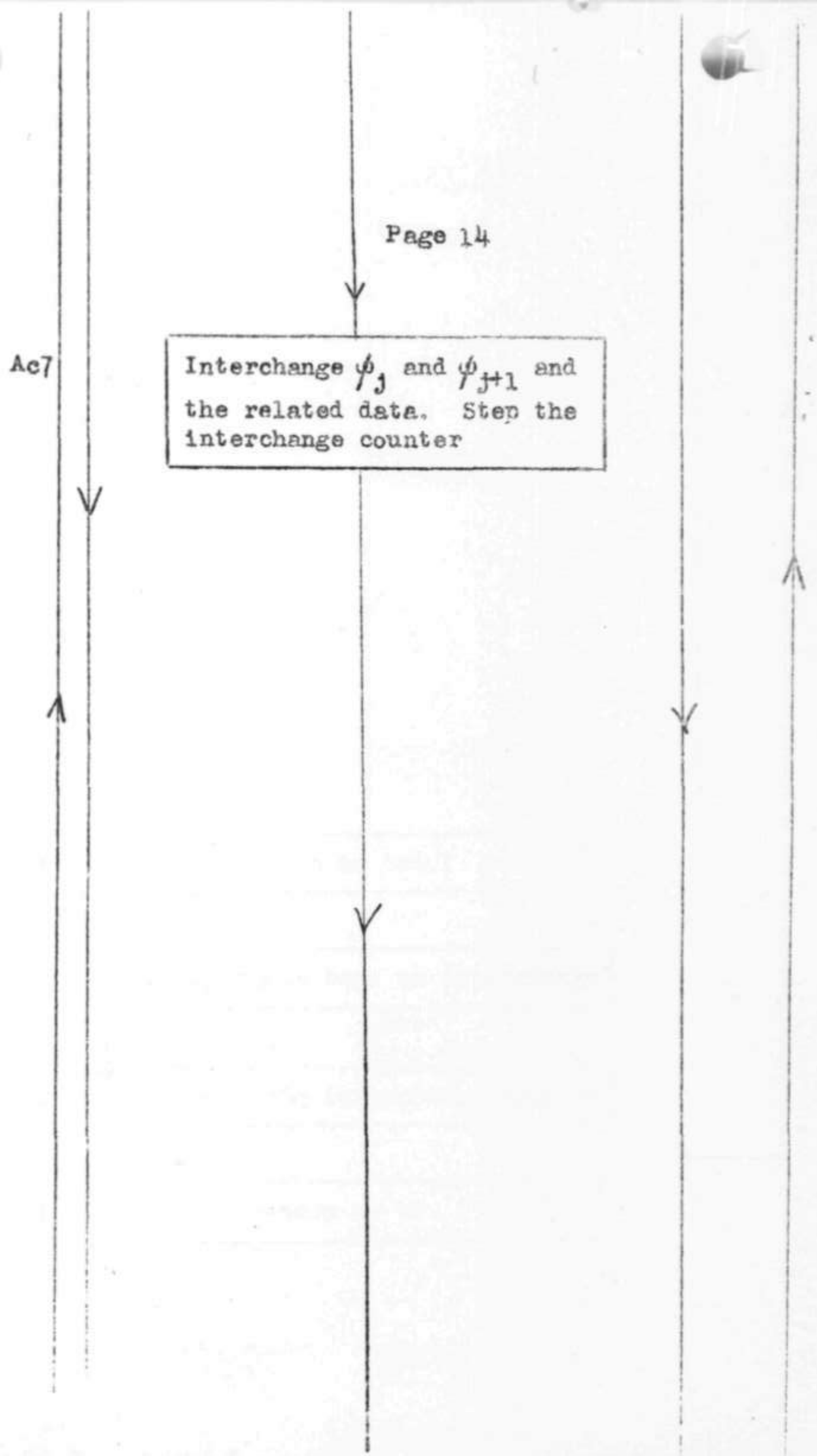
```
Ac1.1   ta  B4

Ac2.1   ca  B9       y x 2^-15
     2  ts  B17
     3  su  B8       (y-x)2^-15
     4  cp  Ac3.1
     5  sp  Ac4.1
```



Ac3.1	ca	B9	$y \times 2^{-15}$
2	ad	C7	$(y+2^7)\times 2^{-15}$
3	ts	B17	
Ac4.1	ca	B8	$x \cdot 2^{-15}$
2	ts	B16	
Ac5.1	ca	B17	$y' \times 2^{-15}$
2	su	C2	$(y'-1) \times 2^{-15}$
3	su	B16	$(y'-1-x') \times 2^{-15}$
4	cp	Ac9.1	
Ac6.1	ca	B16	$x' \cdot 2^{-15}$
2	sl	8	
3	sr	8	$\overline{x'} \cdot 2^{-15}$
4	ad	C3	$(c+\overline{x'}) \times 2^{-15}$
5	td	Ac6.13	
6	ca	B16	$x' \cdot 2^{-15}$
7	ad	C2	$(x'+1) \times 2^{-15}$
8	ts	B16	
9	sl	8	
10	sr	8	
11	ad	C3	$(c+\overline{x'+1}) \times 2^{-15}$
12	td	Ac6.14	
13	ca	--	ϕ'_j
14	su	--	$\phi'_j - \phi'_{j+1} = \phi_j - \phi_{j+1}$
15	cp	Ac8.1	



Ac7.1	ca	Ac6.14	su RC ϕ'_{j+1}
2	td	Ac7.33	
3	td	Ac7.42	
4	ad	C4	su RC i_{j+1}
5	td	Ac7.35	
6	td	Ac7.44	
7	ad	C5	su RC $(\theta_o + 2\pi)_{j+1}$
8	td	Ac7.37	
9	td	Ac7.46	
10	ad	C6	su RC $(\phi' - \phi_o)_{j+1}$
11	td	Ac7.39	
12	td	Ac7.48	
13	ca	Ac6.13	ca RC ϕ_j
14	td	Ac7.25	
15	td	Ac7.34	
16	ad	C4	ca RC i_j
17	td	Ac7.27	
18	td	Ac7.36	
19	ad	C5	ca RC $(\theta_o + 2\pi)_j$
20	td	Ac7.29	
21	td	Ac7.38	
22	ad	C6	ca RC $(\phi' - \phi_o)_j$
23	td	Ac7.31	
24	td	Ac7.40	
25	ca	--	ϕ'_j
26	ts	B11	
27	ca	--	i_j
28	ts	B12	
29	ca	--	$(\theta_o + 2\pi)_j$
30	ts	B13	
31	ca	--	$(\phi' - \phi_o)_j$
32	ts	B14	
33	ca	--	ϕ'_{j+1}
34	ts	--	
35	ca	--	i_{j+1}



Ac7.36	ts	--	
37	ca	--	$(\theta_o + 2\pi)_{j+1}$
38	ts	--	
39	ca	--	$(\phi' - \phi_o)_{j+1}$
40	ts	--	
41	ca	B11	ϕ'_j
42	ts	--	
43	ca	B12	1_j
44	ts	--	
45	ca	B13	$(\theta_o + 2\pi)_j$
46	ts	--	
47	ca	B14	$(\phi' - \phi_o)_j$
48	ts	--	
49	ca	B15	interchange count
50	ad	C2	2^{-15} + interchange count
51	ts	B15	
Ac8.1	sp	Ac5.1	
Ac9.1	ca	B15	
2	cp	Ac11.1	
Ac10.1	su	B15	-0
2	ts	B15	
3	sp	Ac4.1	
Ac11.1	ca	B3	
2	td	Ac11.3	
3	sp	--