THE BINARY EDITOR

AN "S"-LEVEL PROJECT

FOR M.E.I. APPLIED MATHEMATICS

BY J.M.PINNOCK

BEDFORD SCHOOL

LANGUAGE: PAL III

THE BINARY EDITOR.

A Binary Editor is a program which will enable the user to edit or produce Binary tapes. There are two main uses to which such a program could be put. Firstly, if a machine code program had been assembled, andd the Binary tape loaded, there might be a small fault somewhere inside it, such as an "SNA" (code: 745Ø) instead of an "SZA" (code: 744Ø). Obviously, if this were a large program, to use a Symbolic Editor to correct the fault on the original symbolic source tape, and then a two-pass assembler, would be exceedingly tedious. In this case, the Binary Editor would be used to read in the Binary tape, locate the offending code, alter it, and punch out a corrected Binary tape.

Secondly, the Binary Editor could be used to produce a Binary tape of a short machine code program, if the instructiond were input as octal codes rather than standard mnemonics (i.e. 745Ø would be input instead of SNA). This would again reduce a two-stage process to a one-stage process. This method could also be used to produce tapes of sections of programs containing pure data, which would be in octal number format in the symbolic source tape in any case.

A copy of the explanatory pamphlet distributed to potential users of the Binary Editor is enclosed. This explains how to use Binary Editor Mk.I. Since the production of this pamphlet, Binary Editor Mk.II has been developed. The only differences in this version are that it uses an additional 4K of core for text storage, and that the command "N" can now be treated entirely as a high-speed tape version of the command "R".

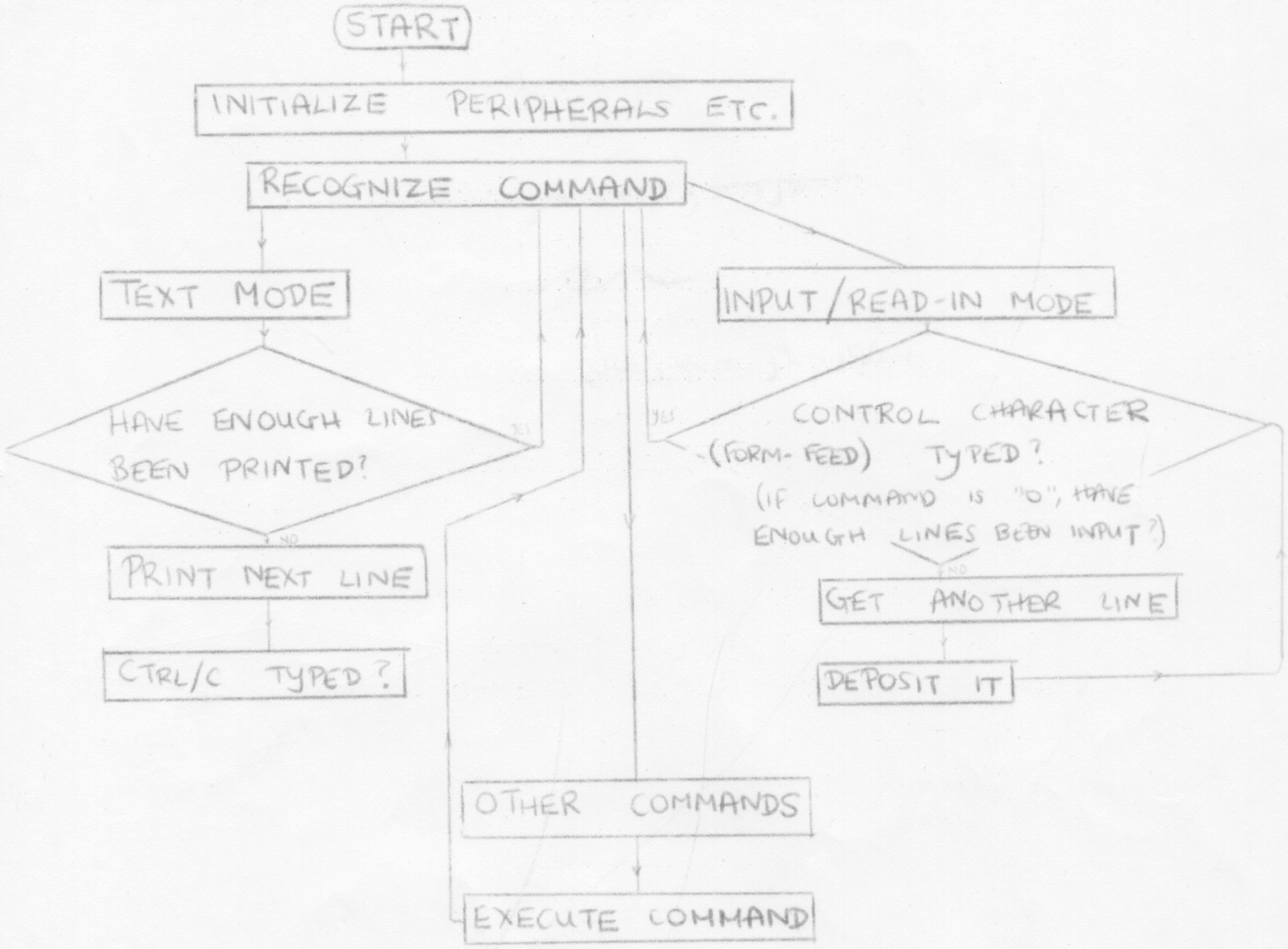
Also enclosed are four flow-diagrams. These illustrate the overall plan of the program, the main command program (Command mode), the command "R" (Input mode) and the command "P" (Text mode). Listings of the programs (Binary Editors I & II) are also included. Binary Editor I is well-tested and is known to be working, apart from one small error (pencilled in on the listing) which has only come to light very recently. Binary Editor II is relatively untested, but as Binary Editor I works, it is reasonable to assume that Binary Editor II is operational.

FUTURE DEVELOPMENT.

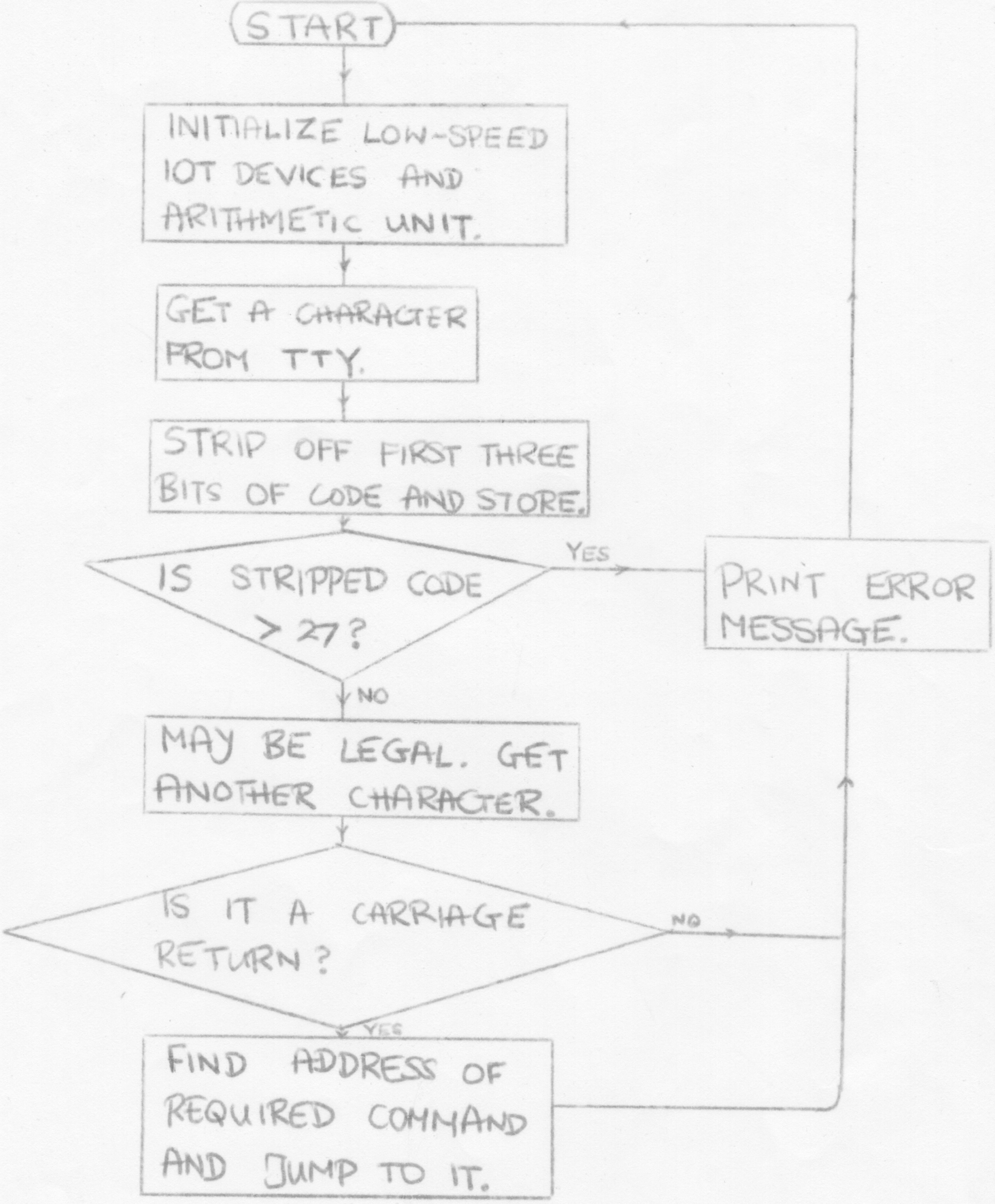
The present Binary Editor is as economical on storage as possible, but with the ability to store large programs comes ~~the~~ shortcomings , such as the necessity to store a starting location as 7777, followed by the location. This, of course, means that 7777 cannot be used as part of text. To get over this limitation, later Binary Editors will use two stores for each line of text (6753 will be represented as ØØ67,ØØ53 and *Ø2ØØ as Ø1Ø2, ØØØØ). There will also be a Binary Editor with four extra new commands, for short programs. This will use the first field of core for storage of the Binary Editor itself and text, and will keep the second field clear for the new commands "A" and "G"."A" (Address) will load the program in text into the correct place in field 1, whilst "G" (Go) will enable the user to start the program thus loaded at a specified point. Further additions include "S"(Search) which will permit the user to search for any particular word (e.g. 3476,*Ø2ØØ) and the computer will tell him which lines that particular word occurs. Use of the command "Q" will give the operator the number of lines in his program.

Further plans include experiments with the interrupt system, leading to a two-user Binary Editor. The Binary Editor could also be used as a basis for a machine code tutor program, and could be improved to accept mnemonics as well as octal numbers.

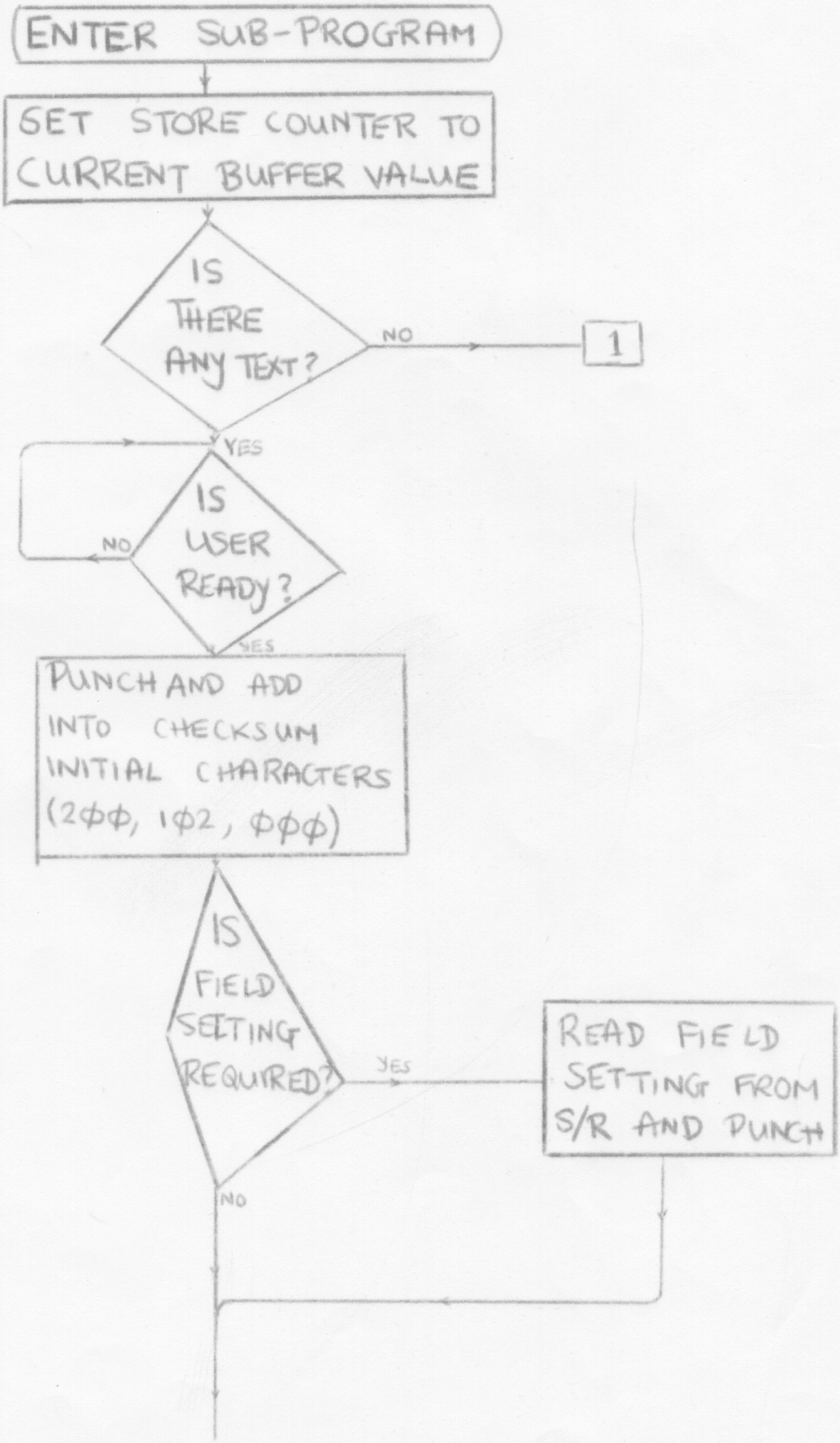
FLOW DIAGRAM OF PROGRAM

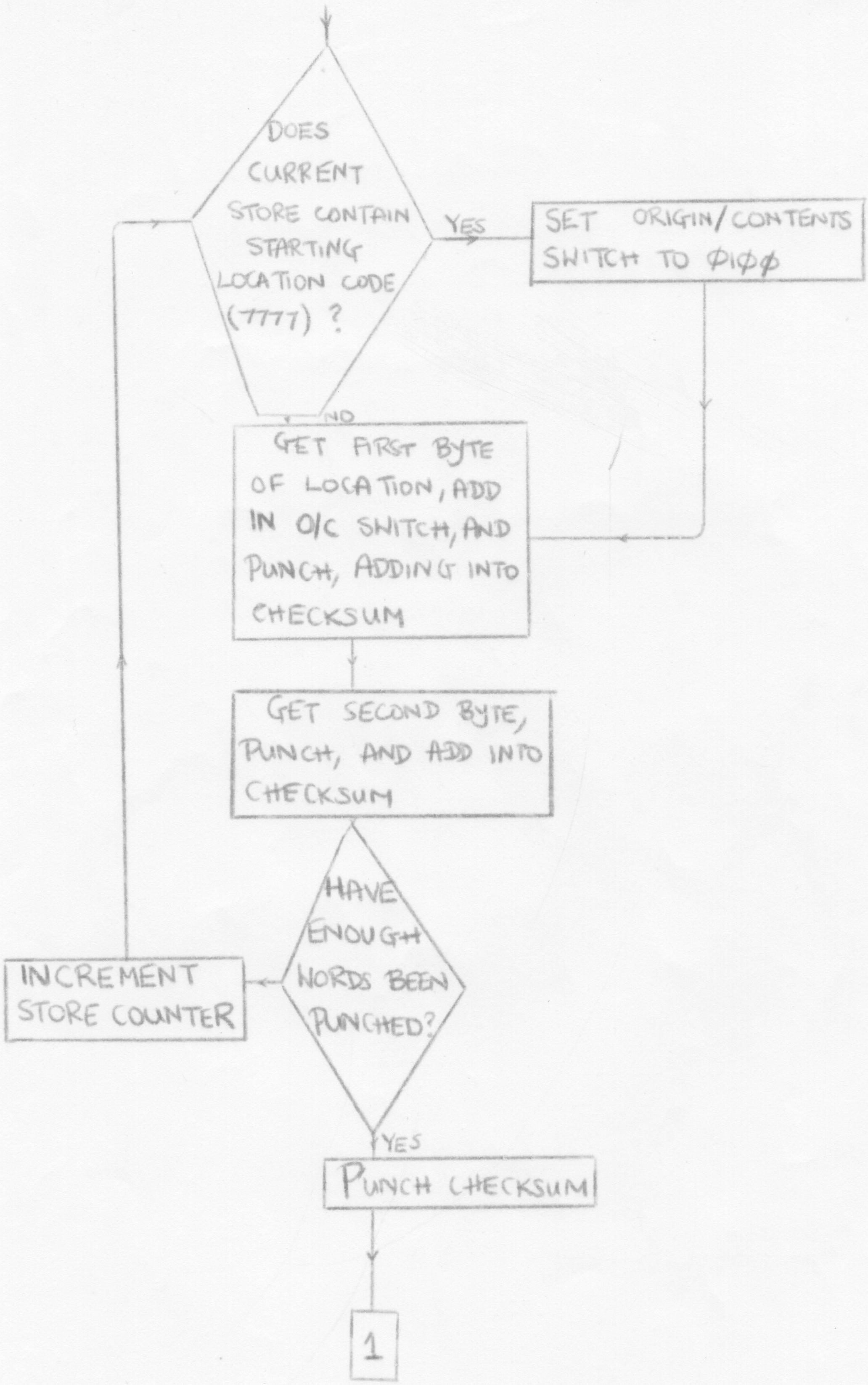

START
INITIALIZE PERIPHERALS ETC.
RECOGNIZE COMMAND
TEXT MODE
HAVE ENOUGH LINES BEEN PRINTED?
YES
NO
PRINT NEXT LINE
CTRL/C TYPED?
INPUT/READ-IN MODE
CONTROL CHARACTER (FORM-FEED) TYPED?
(IF COMMAND IS "O", HAVE ENOUGH LINES BEEN INPUT?)
YES
NO
GET ANOTHER LINE
DEPOSIT IT
OTHER COMMANDS
EXECUTE COMMAND

FLOW DIAGRAM ONE: CONTROL PROGRAM


START
INITIALIZE LOW-SPEED IOT DEVICES AND ARITHMETIC UNIT.
GET A CHARACTER FROM TTY.
STRIP OFF FIRST THREE BITS OF CODE AND STORE.
IS STRIPPED CODE > 27?
YES
PRINT ERROR MESSAGE.
NO
MAY BE LEGAL. GET ANOTHER CHARACTER.
IS IT A CARRIAGE RETURN?
NO
YES
FIND ADDRESS OF REQUIRED COMMAND AND JUMP TO IT.

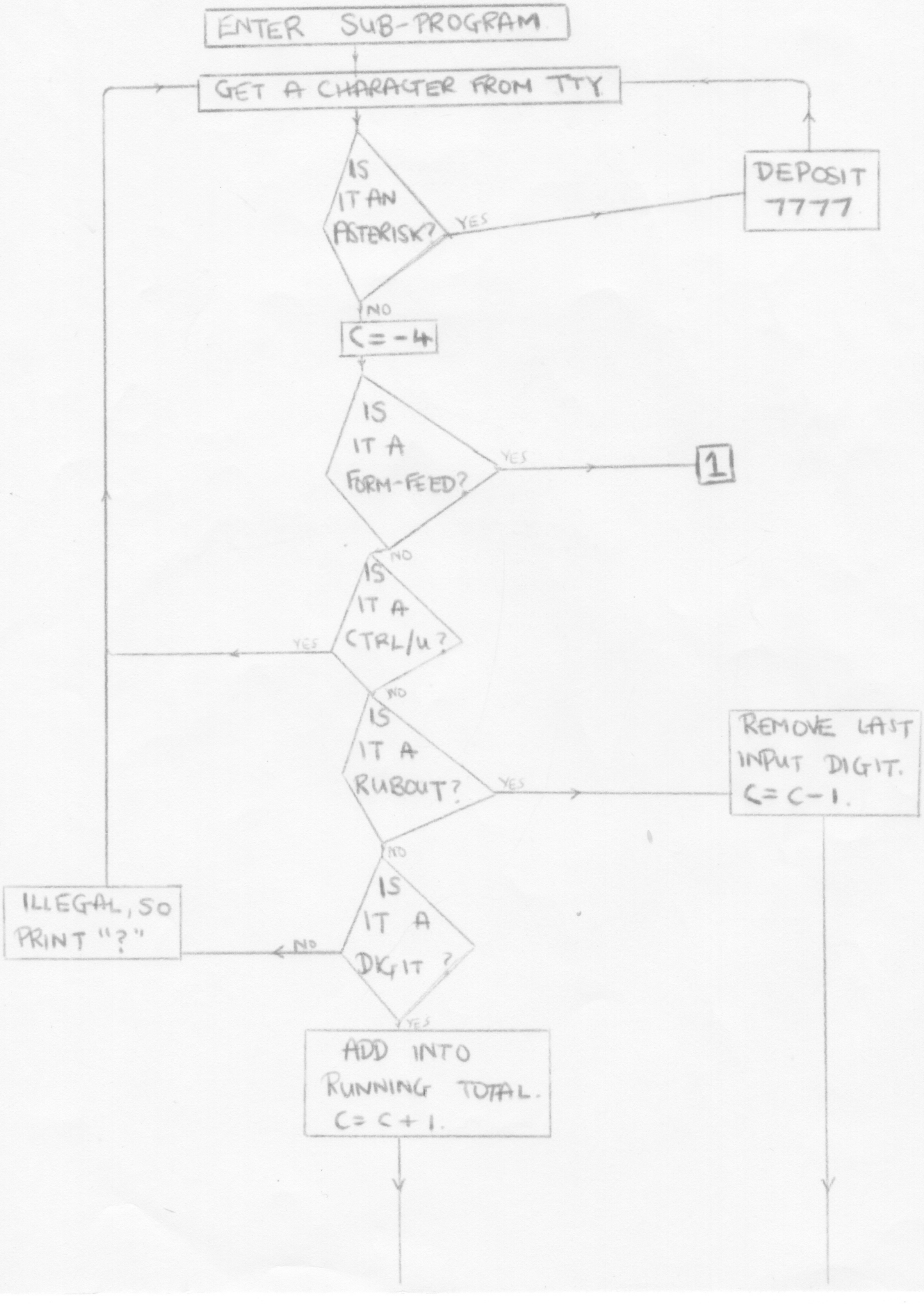
FLOW DIAGRAM TWO: “COMMP” SUB-PROGRAM

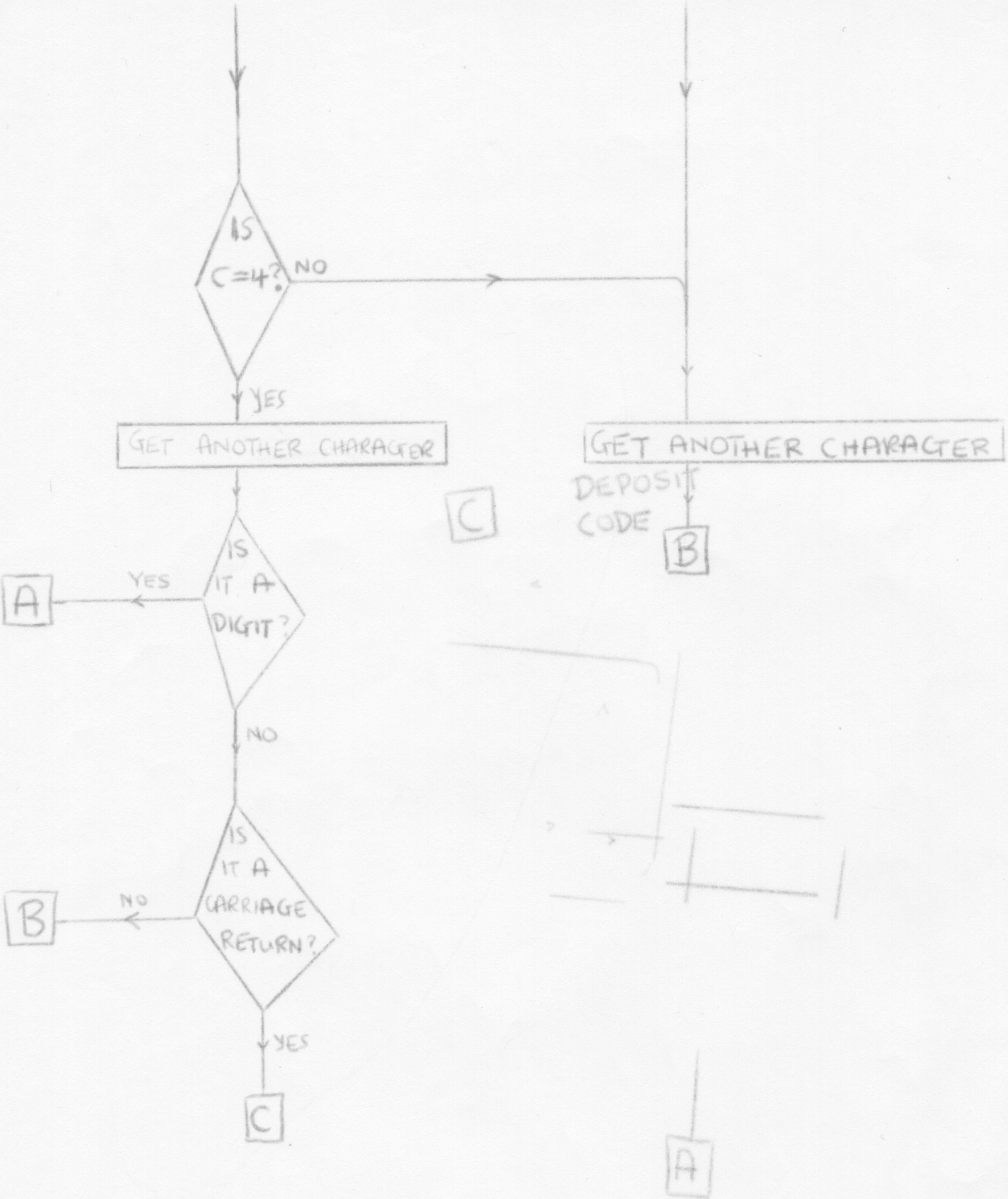

ENTER SUB-PROGRAM
SET STORE COUNTER TO CURRENT BUFFER VALUE
IS THERE ANY TEXT?
NO
1
YES
IS USER READY?
NO
YES
PUNCH AND ADD INTO CHECKSUM INITIAL CHARACTERS (2ϕϕ, 1ϕ2, ϕϕϕ)
IS FIELD SETTING REQUIRED?
YES
READ FIELD SETTING FROM S/R AND PUNCH
NO


DOES CURRENT STORE CONTAIN STARTING LOCATION CODE (7777) ?
YES
SET ORIGIN/CONTENTS SWITCH TO Ø1ØØ
NO
GET FIRST BYTE OF LOCATION, ADD IN O/C SWITCH, AND PUNCH, ADDING INTO CHECKSUM
GET SECOND BYTE, PUNCH, AND ADD INTO CHECKSUM
HAVE ENOUGH WORDS BEEN PUNCHED?
INCREMENT STORE COUNTER
YES
PUNCH CHECKSUM
1

FLOW DIAGRAM THREE:

"COMMR" SUB-PROGRAM




IS C=4?
NO
YES
GET ANOTHER CHARACTER
GET ANOTHER CHARACTER
DEPOSIT CODE
C
B
IS IT A DIGIT?
YES
A
NO
IS IT A CARRIAGE RETURN?
NO
B
YES
C
A