A

AS
COMPUTER SCIENCE
Paper 1
7516/1

Tuesday 16 May 2023 Afternoon

Time allowed: 1 hour 45 minutes

[Turn over]

MATERIALS

For this paper you must have:

- **a computer**
- **a printer**
- **appropriate software**
- **the Electronic Answer Document**
- **an electronic version and a hard copy of the Skeleton Program**
- **an electronic version and a hard copy of the Preliminary Material**
- **an electronic version of the Data Files prog1.txt, prog2.txt and prog3.txt.**

You must NOT use a calculator.

INSTRUCTIONS

- **Type the information required on the front of your Electronic Answer Document.**
- **Before the start of the examination make sure your CENTRE NUMBER, CANDIDATE NAME and CANDIDATE NUMBER are shown clearly IN THE FOOTER of every page (not the front cover) of your Electronic Answer Document.**
- **Enter your answers into the Electronic Answer Document.**
- **Answer ALL questions.**
- **Save your work at regular intervals.**

[Turn over]

INFORMATION

- The marks for questions are shown in brackets.
- The maximum mark for this paper is 75.
- No extra time is allowed for printing and collating.
- The question paper is divided into THREE sections.

ADVICE

You are advised to allocate time to each section as follows:
SECTION A – 20 minutes;
SECTION B – 25 minutes;
SECTION C – 60 minutes.

AT THE END OF THE EXAMINATION

Tie together all your printed Electronic Answer Document pages and hand them to the Invigilator.

WARNING

It may not be possible to issue a result for this paper if your details are not on every page of your Electronic Answer Document.

DO NOT TURN OVER UNTIL TOLD TO DO SO

BLANK PAGE

SECTION A

You are advised to spend no more than 20 MINUTES on this section.

Enter your answers to SECTION A in your Electronic Answer Document. You MUST SAVE this document at regular intervals.

Question 05 in this section asks you to write program code STARTING FROM A NEW PROGRAM / PROJECT / FILE.

You are advised to save your program at regular intervals.

[Turn over]

0	1

A security system uses a motion sensor, a keypad and an alarm bell.

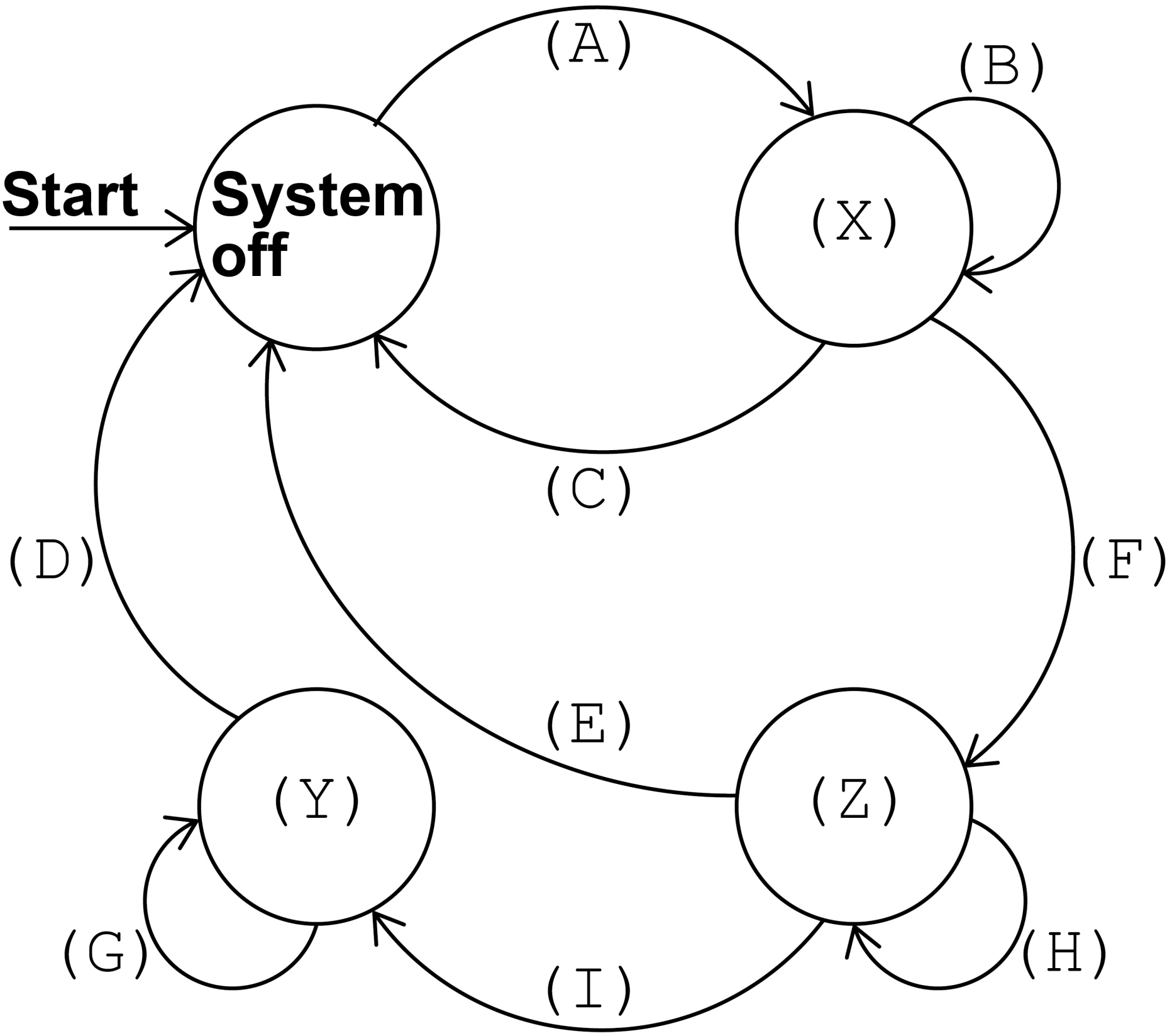
The system operates as follows:

- **the system is initially off**
- **when the system is switched on it goes into sensing mode**
- **the system can be switched off at any time by entering the correct code on the keypad**
- **if the system detects movement while in sensing mode it goes into alert mode**
- **after the system has been in alert mode for 10 seconds, it enters the alarm bell ringing mode**

- **if an incorrect code is entered on the keypad, once the system has been switched on, the system remains in its current mode.**

FIGURE 1, on page 10, shows a partially completed state transition diagram that represents the operation of the security system. Three of the states are labelled (`X`) to (`Z`) and events are labelled (`A`) to (`I`).

FIGURE 1



Complete TABLE 1, on page 13, by filling in the unshaded cells with the correct labels from FIGURE 1. You should write:

- **which labels, (`X`) to (`Z`), represent which state**
- **which labels (`A`) to (`I`) represent which event(s).**

Some of the events will be assigned more than one label.

Each label MUST only be used once.

Copy the contents of all the unshaded cells in TABLE 1, on page 13 into your Electronic Answer Document. [6 marks]

[Turn over]

BLANK PAGE

TABLE 1

EVENT / STATE	LABEL(S): (A) to (I), (X) to (Z)
Alarm bell ringing mode	
Alert mode	
Detect movement	
Enter correct code	
Enter incorrect code	
Sensing mode	
Switch on	
10 second delay elapsed	

[Turn over]

0	2

The algorithm, represented using pseudo-code in FIGURE 2, describes a method to rearrange four numbers in a data structure.

FIGURE 2

```
Numbers[0] ← 45
Numbers[1] ← 19
Numbers[2] ← 62
Numbers[3] ← 12
FOR X ← 1 TO 3
   Y ← X - 1
   N ← Numbers[X]
   WHILE Y > -1 AND N < Numbers[Y]
      Numbers[Y + 1] ← Numbers[Y]
      Y ← Y - 1
   ENDWHILE
   Numbers[Y + 1] ← N
ENDFOR
```

Complete TABLE 2, on page 17, by hand-tracing the algorithm in FIGURE 2.

You may not need to use all the rows in TABLE 2.

The first row of TABLE 2 has already been completed for you.

[Turn over]

BLANK PAGE

TABLE 2

X	Y	N	Numbers			
			[0]	[1]	[2]	[3]
			45	19	62	12

Copy the contents of all the unshaded cells in TABLE 2 into your Electronic Answer Document. [5 marks]

[Turn over]

0	3

Describe ONE difference between a global variable and a local variable. [2 marks]

0	4

Define the term algorithm. [2 marks]

BLANK PAGE

[Turn over]

0 5

The algorithm, represented using pseudo-code, in FIGURE 3 outputs a series of integers. The output depends upon the value entered by the user.

FIGURE 3

```
OUTPUT "Enter an integer greater than 1: "
INPUT Number
X ← 2
Count ← 0
WHILE Number > 1
   Multi ← FALSE
   WHILE (Number MOD X) = 0
      IF NOT Multi THEN
         OUTPUT X
```

```
      ENDIF
      Count ← Count + 1
      Multi ← TRUE
      Number ← Number DIV X
   ENDWHILE
   X ← X + 1
ENDWHILE
OUTPUT Count
```

TABLE 3, on page 23, lists the `MOD` and `DIV` operators for each of the available programming languages. You should refer to the row for your programming language.

[Turn over]

BLANK PAGE

TABLE 3

PROGRAMMING LANGUAGE	`MOD`	`DIV`
C#	`%`	`/`
Java	`%`	`/`
Pascal	`mod`	`div`
Python	`%`	`//`
VB.NET	`Mod`	`\`

[Turn over]

WHAT YOU NEED TO DO:

TASK 1

Write a program to implement the algorithm in FIGURE 3, on pages 20 and 21.

TASK 2

Test that your program works:

- **run your program, then enter the number** `23`
- **run your program, then enter the number** `25`
- **run your program, then enter the number** `1260`

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

0 5 . 1

Your PROGRAM SOURCE CODE for TASK 1. [9 marks]

0 5 . 2

SCREEN CAPTURE(S) showing the tests described in TASK 2. [1 mark]

[Turn over]

SECTION B

You are advised to spend no more than 25 MINUTES on this section.

Enter your answers to SECTION B in your Electronic Answer Document. You MUST SAVE this document at regular intervals.

These questions refer to the PRELIMINARY MATERIAL and the SKELETON PROGRAM, but do NOT require any additional programming.

Refer EITHER to the PRELIMINARY MATERIAL issued with this question paper OR your electronic copy.

0 6

State the identifier for:

0 6 . 1

a variable that is used to store a Boolean value. [1 mark]

0 6 . 2

a user-defined subroutine that returns only ONE value that is AN INTEGER. [1 mark]

0 6 . 3

a user-defined subroutine that returns only ONE value that is A SINGLE CHARACTER. [1 mark]

0 6 . 4

a user-defined subroutine that contains exception handling. [1 mark]

0 7

The SKELETON PROGRAM uses a number of data structures.

0 7 . 1

State the identifier of a data structure that stores values of MORE THAN ONE DATA TYPE. [1 mark]

0 7 . 2

State the identifier of a data structure that stores values of ONLY ONE DATA TYPE. [1 mark]

0 8

The SKELETON PROGRAM models a very simple processor. Many details of how a real processor works were removed because they did not need to be included for the purposes of the simulation.

State the computing term that best describes the concept of removing detail that is unnecessary from a problem when developing a solution. [1 mark]

0	9

When the SKELETON PROGRAM detects an error it outputs an error code which is a number between 1 and 11.

The error codes could be replaced by more helpful error messages.

Complete TABLE 4, on the opposite page, by writing the most appropriate error code from the SKELETON PROGRAM next to each proposed error message.

TABLE 4

PROPOSED ERROR MESSAGE	ERROR CODE
Duplicate label found	
File not found	
No assembled code to run	
No source code to display	
Unknown opcode	

Copy the contents of all the unshaded cells in TABLE 4 into your Electronic Answer Document. [5 marks]

1 0

This question refers to the subroutine `PassTwo`.

State TWO conditions that need to be met for error code 6 to be reported during the assembly process of the program. [2 marks]

1 1

This question refers to the subroutine `ExtractOperand`.

1 1 . 1

Describe the purpose of the FOR loop AND the IF statement INSIDE it.
[2 marks]

1 1 . 2

Describe the purpose of the IF statement AFTER the FOR loop. [2 marks]

1 1 . 3

Describe why a WHILE loop might have been a better choice than a FOR loop. [2 marks]

SECTION C

You are advised to spend no more than 60 MINUTES on this section.

Enter your answers to SECTION C in your Electronic Answer Document. You MUST SAVE this document at regular intervals.

These questions require you to load the SKELETON PROGRAM and to make programming changes to it.

1	2

This question extends the functionality of the SKELETON PROGRAM.

The functionality of the `SKP` opcode is to be changed so that it increments the value stored in the accumulator by 1

For example, if the accumulator contained 4, then executing the instruction

```
SKP
```

would change the value in the accumulator to 5

WHAT YOU NEED TO DO:

TASK 1

Amend the subroutine `ExecuteSKP` **so that it adds 1 to the accumulator and then updates the status register if required. The** `Registers` **data structure should be passed as a parameter.**

TASK 2

Amend the call to `ExecuteSKP` **in the** `Execute` **subroutine as required.**

[Turn over]

TASK 3

Test that the changes you have made work by conducting the following test:

- **run your amended SKELETON PROGRAM**
- **enter** `L`
- **load prog2**
- **enter** `A`
- **enter** `R`

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

1 2 . 1

Your PROGRAM SOURCE CODE for the entire subroutine `ExecuteSKP` **and the entire subroutine** `Execute`**. [4 marks]**

1 2 . 2

SCREEN CAPTURE(S) showing the requested test described in TASK 3.

The SCREEN CAPTURE(S) only need to show all of Frame 0 and all of the final frame. [1 mark]

[Turn over]

1	3

This question adds validation to the SKELETON PROGRAM. The subroutine `EditSourceCode` **asks the user to enter a line number. The line number must be an integer and the number of an existing line in the file containing the program.**

For example, the last line (line 11) of `prog2.txt` **is:**

```
    FINAL:          0
```

Therefore, a valid line number for `prog2.txt` **would be an integer between 1 and 11 inclusive.**

WHAT YOU NEED TO DO:

TASK 1

Amend the subroutine `EditSourceCode` **to check that the line number entered by the user is a valid**

existing line number. If an invalid value is entered the subroutine should output an appropriate error message. The program must not continue until a valid line number has been entered.

TASK 2

Test that the changes you have made work by conducting the following test:

- run your amended SKELETON PROGRAM
- enter `L`
- load prog2
- enter `E`
- enter `Q`
- enter `22`
- enter `0`
- enter `2`

[Turn over]

BLANK PAGE

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

1 3 . 1

Your PROGRAM SOURCE CODE for the entire subroutine `EditSourceCode`**.**

[5 marks]

1 3 . 2

SCREEN CAPTURE(S) showing the requested test described in TASK 2.

[1 mark]

1	4

This question adds a memory address check to the SKELETON PROGRAM.

Each time a `JSR` **opcode is executed the return address is stored in memory. The memory location used for this could already contain an instruction or data.**

If storing a return address will overwrite an instruction or data, then an appropriate error message should be output using the `ReportRunTimeError` **subroutine. The original code in the** `ExecuteJSR` **subroutine should only be carried out if there is no error.**

WHAT YOU NEED TO DO:

TASK 1

Amend the `ExecuteJSR` **subroutine. If storing a return address would overwrite an instruction or data, the** `ReportRunTimeError` **subroutine should be called with an appropriate error message.**

One method of completing this task would require the addition of extra parameter(s) to the `ExecuteJSR` **subroutine.**

[Turn over]

TASK 2

If your solution requires additional parameter(s) for the `ExecuteJSR` **subroutine amend the call to** `ExecuteJSR` **in the** `Execute` **subroutine.**

TASK 3

Test that the changes you have made work by conducting the following test:

- **run your amended SKELETON PROGRAM**
- **enter** `L`
- **load prog3**
- **enter** `A`
- **enter** `R`

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

1 4 . 1

Your PROGRAM SOURCE CODE for the entire subroutine `ExecuteJSR` **and the entire subroutine** `Execute` **if you have made changes in TASK 2. [4 marks]**

1 4 . 2

SCREEN CAPTURE(S) showing the requested test described in TASK 3.

The SCREEN CAPTURE(S) only need to show the final frame and the contents of the stack before execution terminates. [1 mark]

1	5

This question extends the functionality of the SKELETON PROGRAM. The option to edit the source code is to be extended to allow lines to be deleted or inserted within a loaded source code program.

The options within the `EditSourceCode` **subroutine should now be:**

```
E - Edit this line
D - Delete the current line
I - Insert a new line above this line
C - Cancel edit
```

OPTION `D` should delete the current line without changing the size of the data structure. The lines after the deleted line need to be moved within the data structure so that there is no gap in the source code.

FIGURE 4, pages 48 and 49, shows an example where line 10 is being deleted. As a result lines 11 to 13 move. The size of the data structure does not change.

[Turn over]

FIGURE 4

```
1      NUM1:      2
2      NUM2:      5
3      NUM3:      -1
4      NUM4:      125
5     START: LDA  NUM1
6     WHILE: CMP# 12
7            BEQ  WEND
8            ADD  NUM2
9            JMP  WHILE
10           SKP
11     WEND: STA  NUM3
12           ADD  NUM4
13           HLT
14
```

```
NUM1:      2
NUM2:      5
NUM3:      -1
NUM4:      125
START: LDA NUM1
WHILE: CMP#12
       BEQ WEND
       ADD NUM2
       JMP WHILE
WEND:  STA NUM3
       ADD NUM4
       HLT
```

15
16
17
18
19

[Turn over]

You MUST WRITE YOUR OWN DELETE ROUTINE and not use any built-in delete function that might be available in the programming language you are using.

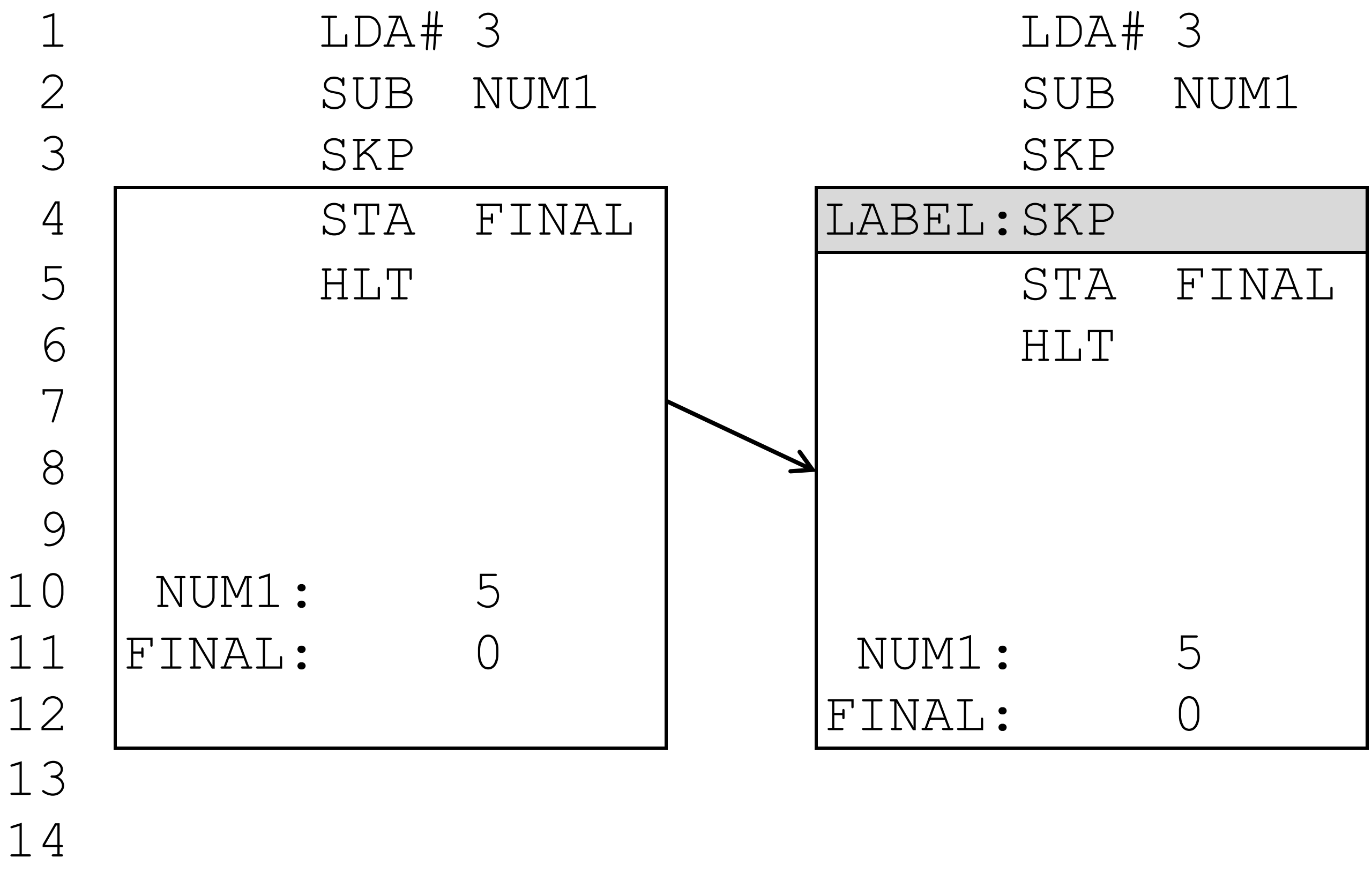
OPTION `I` should allow the user to enter a new line to be inserted above the chosen line without changing the size of the data structure. The lines after the inserted line need to be moved within the data structure so they are not overwritten.

You should check that there is sufficient space in the data structure to accommodate a new line. If there is not sufficient space, an error message should be displayed.

FIGURE 5, on pages 52 and 53, shows an example where lines 4 to 11 move and a new line 4 is inserted. The size of the data structure does not change.

[Turn over]

FIGURE 5


1 LDA# 3
2 SUB NUM1
3 SKP
4 STA FINAL
5 HLT
6
7
8
9
10 NUM1: 5
11 FINAL: 0
12
13
14
LDA# 3
SUB NUM1
SKP
LABEL:SKP
STA FINAL
HLT
NUM1: 5
FINAL: 0

15
16
17
18
19

[Turn over]

You MUST WRITE YOUR OWN INSERT ROUTINE and not use any built-in insert function that might be available in the programming language you are using.

WHAT YOU NEED TO DO:

TASK 1

Amend the `EditSourceCode` **subroutine to include the delete line option.**

TASK 2

Test that the changes you have made work by conducting the following test:

- **run your amended SKELETON PROGRAM**
- **enter** `L`
- **load prog1**

- **enter** `E`
- **enter** `10`
- **enter** `D`

TASK 3

Amend the `EditSourceCode` **subroutine to include the insert line option.**

[Turn over]

TASK 4

Test that the changes you have made work by conducting the following test:

- **run your amended SKELETON PROGRAM**
- **enter** `L`
- **load prog2**
- **enter** `E`
- **enter** `4`
- **enter** `I`
- **enter** `LABEL: SKP`

EVIDENCE THAT YOU NEED TO PROVIDE

Include the following evidence in your Electronic Answer Document.

1 5 . 1

Your PROGRAM SOURCE CODE for the entire subroutine `EditSourceCode`**. [12 marks]**

1 5 . 2

SCREEN CAPTURE(S) showing the requested test described in TASK 2.

The SCREEN CAPTURE(S) only need to show the test from entering option `E` **until after the edited source code has been displayed. [1 mark]**

1	5	.	3

SCREEN CAPTURE(S) showing the requested test described in TASK 4.

The SCREEN CAPTURE(S) only need to show the test from entering option E until after the edited source code has been displayed. [1 mark]

END OF QUESTIONS

BLANK PAGE

BLANK PAGE