

ASSOCIATION OF COMPUTER PROFESSIONALS

CERTIFICATE EXAMINATION

SPECIMEN PAPER

APPLIED PROGRAMMING

(TWO AND A HALF HOURS ALLOWED)

You have ten minutes to read through this paper before the start of the examination.

You should have the following for this examination:

*An Answer Book (or writing paper)
A Flowcharting Template
Coding Sheets
Print Layout Forms*

READ THE INSTRUCTIONS BELOW CAREFULLY:

Answer only **ONE** question.

Present your answer in the following parts:

1. A brief Introduction (but do not copy out the question)
2. A description of your Analysis and Method
3. A Flowchart (or pseudocode) of the logic
4. A Program written in a language of your choice
5. Test Data (usually an effects table or dry run)
6. A brief Summary or Conclusion

You should attempt to answer every part of your chosen question, including the testing, even if you have insufficient time to complete all the work.

Presentation	carries	10 Marks
Analysis and Method		25 Marks
Flowchart (or pseudocode)		25 Marks
Program coding		20 Marks
Testing		15 Marks
Summary or Conclusion		5 Marks

1. Change Calculation and Analysis.

At a supermarket check-out, the cash register adds the prices of goods and calculates the total due.

The customer pays an amount of money, and the cash register also calculates any change due; it then analyses the amount of change into denominations of notes and coins.

The denominations are: £20, £10, £5 in notes and £2, £1, 50p, 20p, 10p, 5p, 2p and 1p in coins (£1 = 100p).

For example, the cash register prints the following on its till receipt:

```
TOTAL DUE:      £26.01p
AMOUNT PAID:    £40.00p
CHANGE DUE:     £13.99p

ISSUE:  1      £10    note.
         1       £2     coin.
         1       £1     coin.
         1        50p  coin.
         2        20p  coins.
         1         5p  coin.
         2         2p  coins.

THANK YOU FOR YOUR CUSTOM.
-----
```

- a. Construct a flowchart and write a program to perform the change calculation and analysis as follows:
- (1) Enter the Total Due and the Amount Paid.
 - (2) Calculate the Change Due.
 - (3) Analyse the Change Due into the required denominations of notes and coins to be issued. (The analysis should yield the smallest number of notes or coins; thus 40p should be divided into 2 of 20p, and **not** 4 of 10p.)
 - (4) Print the till receipt in the format shown above; note that zero denominations should not be printed on the list.
 - (5) Include any error messages as necessary.
- b. Test the program, using a 'dry run' (effects table), and showing the print-out for each of the following items of data:

	TOTAL DUE	AMOUNT PAID
(1)	£72.75p	£80.00p
(2)	£14.40p	£20.00p
(3)	£8.77p	£10.00p
(4)	£9.00p	£9.00p
(5)	£16.50p	£6.50p

2. **Three Name Sort.**

A program is needed to sort names in alphabetical order. The first (prototype) version should work as shown below:

```

                                THREE NAME SORT

Enter Three Names:                Sorted Names are:

> Joe                            > Betty
> Mary                           > Joe
> Betty                          > Mary

                                Another Sort? (Y/N): _

```

The user is prompted to enter three names; each name must begin with a letter (A, B, C,...) and have between one and twelve characters. If all three names are valid, the program will sort the names in alphabetical order and display them alongside the original entries. If invalid names are detected, the program should display a suitable error message (such as 'Name too long') for each incorrect name. After each sort, the program should ask the user whether another sort is required.

a. Construct a flowchart (or pseudocode) and write a program to:

- (1) Create the screen layout shown above.
- (2) Enter three names.
- (3) Validate each name.
- (4) If all names are valid, sort them in alphabetical order and display the sorted names alongside the original entries; if errors are detected, suitable messages should be displayed (and no sorting should occur).
- (5) Ask the user whether 'Another Sort?' is required; if the response is 'Y', steps (1) to (4) should be repeated as many times as desired.

b. Test the logic of the program by constructing a dry run (effects table) for the following input data:

- | | |
|------------|---|
| (1) Zaheer | (4) Peter |
| Pravin | adam |
| Daniel | Michael |
| (2) Ali | (5) Dharmendra Shah |
| Nadir | 123 |
| Maria | Brian |
| (3) Tariq | (6) DANNY |
| Grace | BILL |
| Jitesh | Null string - zero length entry |
| | obtained by pressing the RETURN key only. |

Note that the QBASIC instructions:

```
LOCATE 5, 8
PRINT ">"
```

... would position the cursor at row 5, column 8 and print ">" at that position on the screen.

3. **Sequential File Search.**

A direct access file contains 16 records. The records are in sequence on a numeric key field. A record can be accessed using its logical record number (LRN):

LRN:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Key field:	11	15	16	20	23	24	28	31	34	37	42	48	59	72	81	86

A program is required which searches the file using a binary chop method. The user enters a search value (such as 37) at the keyboard, and the program reports whether a record with a matching key field has been found. The LRN of a matching record and the number of file read actions are also displayed. For example:

Search Value	Report	Number of File Reads
37	FOUND at LRN 10	3
22	NOT FOUND	4

- a. Construct a flowchart and write a program to search the data file using the binary chop method.
- b. Test the logic using an effects table on the Search Values below. (The table should report 'FOUND' or 'NOT FOUND', the LRN of a matched record and the number of file reads. It should also show program variables as they change.)
 - (1) 37) As
 - (2) 22) above.
 - (3) 86
 - (4) 8
 - (5) 11.