

COMPUTER SCIENCE**Paper – 2
(PRACTICAL)***(Maximum Marks: 30)**(Time allowed: Three hours)*

*(Candidates are allowed additional 15 minutes for **only** reading the paper.
They must NOT start writing during this time.)*

The total time to be spent on the Planning session and Examination session is Three hours.

Planning session: 90 minutes

Examination session: 90 minutes

Note: Candidates are to be permitted to proceed to the Examination Session only after the 90 minutes of the Planning Session are over.

*This paper consists of **three** problems from which candidates are required to attempt **any one** problem.*

Candidates are expected to do the following:

1. Write an algorithm for the selected problem. [3]
(Algorithm should be expressed clearly using any standard scheme such as pseudo code or in steps which are simple enough to be obviously computable)
2. Write a program in **JAVA** language. The program should follow the algorithm and should be logically and syntactically correct. Document the program using mnemonic names / comments, identifying and clearly describing the choice of data types and meaning of variables. [7]
3. Code / Type the program on the computer and get a print out (Hard Copy). [2]
Typically, this should be a program that compiles and runs correctly.
4. Test run the program on the computer using the given sample data and get a print out of the output in the format specified in the problem. [3]

In addition to the above, the practical file of the candidate containing the practical work related to programming assignments done during the year is to be evaluated as follows:

- Programming assignments done throughout the year (by the Teacher) [10]
- Programming assignments done throughout the year (by the Visiting Examiner) [5]

Solve any **one** of the following Problems.

Question 1

An **Evil** number is a positive whole number which has even number of 1's in its binary equivalent.

Example: Binary equivalent of 9 is 1001, which contains even number of 1's.

Thus, 9 is an Evil Number.

A few Evil numbers are 3, 5, 6, 9....

Design a program to accept a positive whole number 'N' where $N > 2$ and $N < 100$. Find the binary equivalent of the number and count the number of 1s in it and display whether it is an Evil number or not with an appropriate message.

Test your program with the following data and some random data:

Example 1:

INPUT: N = 15

BINARY EQUIVALENT: 1111

NUMBER OF 1's: 4

OUTPUT: EVIL NUMBER

Example 2:

INPUT: N = 26

BINARY EQUIVALENT: 11010

NUMBER OF 1's: 3

OUTPUT: NOT AN EVIL NUMBER

Example 3:

INPUT: N = 145

OUTPUT: NUMBER OUT OF RANGE

Question 2

The potential of a word is found by adding the ASCII value of the alphabets.
(ASCII values of A to Z are 65 to 90).

Example: BALL
Potential = $66 + 65 + 76 + 76 = 283$

Write a program to accept a sentence which may be terminated by either “.”, “?” or “!” only.
The words of sentence are separated by single blank space and are in UPPER CASE. Decode the words according to their potential and arrange them in ascending order of their potential strength.

Test your program with the following data and some random data:

Example 1:

INPUT: HOW DO YOU DO?

OUTPUT: HOW = 238
DO = 147
YOU = 253
DO = 147

DO DO HOW YOU

Example 2:

INPUT: LOOK BEFORE YOU LEAP.

OUTPUT: LOOK = 309
BEFORE = 435
YOU = 253
LEAP = 290

YOU LEAP LOOK BEFORE

Example 3:

INPUT: HOW ARE YOU#

OUTPUT: INVALID INPUT

Question 3

Write a program to declare a square matrix M [] [] of order 'N' where 'N' must be greater than 3 and less than 10. Allow the user to accept three different characters from the keyboard and fill the array according to the instruction given below:

- (i) Fill the four corners of the square matrix by character 1.
- (ii) Fill the boundary elements of the matrix (except the four corners) by character 2.
- (iii) Fill the non-boundary elements of the matrix by character 3.

Test your program with the following data and some random data:

Example 1:

INPUT: N = 4
 FIRST CHARACTER: @
 SECOND CHARACTER: ?
 THIRD CHARACTER: #

OUTPUT:

@	?	?	@
?	#	#	?
?	#	#	?
@	?	?	@

Example 2:

INPUT: N = 5
 FIRST CHARACTER: A
 SECOND CHARACTER: C
 THIRD CHARACTER: X

OUTPUT:

A	C	C	C	A
C	X	X	X	C
C	X	X	X	C
C	X	X	X	C
A	C	C	C	A

Example 3:

INPUT: N = 12

OUTPUT: SIZE OUT OF RANGE