MCA-03

MCA Examination, June - 2015

Computer Programming Using C

MCA-03

Time : Three Hours] *[Max. Marks : 80*

Note: The Question paper is divided into three sections A, B and C. Write Answers as per given instructions.

Section-A

(Very Short Answer Questions)

(Marks: 2×8=16)

Note : Answer all questions. As per the nature of the question delimit answer in one word, one sentence or maximum upto 30 words. Each question carries 2 marks.

1. (i) What is flowchart?
   (ii) What are possible ways to put comment in C code?
   (iii) What is the use of left-shift and right-shift operator?
   (iv) Explain String Compare function?
   (v) What is the use of typedef keyword?
   (vi) Define Enumerated Data Types.
   (vii) Which function is used to move the file position to a desired location within the file?
   (viii) What is actual parameter?

Section-B

(Short Answer Questions)

(Marks: 8×4=32)

Note: Answer any four questions. Each Answer should not exceed 200 words. Each question carries 8 marks.

2. What do you mean by associativity? What is the associativity of the arithmetic operator?
3. What is pointer? Discuss advantages of pointer.
4. Compare modular and structured programming techniques with example.
5. What is macro? Summarize the similarities and differences between macros and functions.
6. What do you mean by life time of a variable, explain with example?
7. What is Structure? How structure is different from Union?
8. What is recursion? Explain it with example.
9. What is the main difference between getc() and fgets()?

Section-C

(Long Answer Questions)

(Marks: 16×2=32)

Note : Answer any two questions. You have to delimit your each answer maximum upto 500 words. Each question carries 16 marks.

10. What do you mean by Storage Classes? Explain with its classification with example.
11. Explain looping and conditional Structures in C programming in brief.
12. What is the use of flow chart? Draw a flow chart to input three numbers and print the largest number and explain it.
13. Write a C-program to obtain the following output :

```
*
* *
* * *
* * * *
```

—x—