B.Tech.
(SEMESTER-II) CARRY OVER EXAMINATION, 2012-13
COMPUTER PROGRAMMING

Time : 3 Hours /

( Total Marks : 100

SECTION – A

1. Attempt all question parts. 10 × 2 = 20

(a) Give the rules for writing pseudocodes.
(b) Define bit, byte and words.
(c) Write output for the following:
   (i) unsigned a = −5;
   printf("%u", a);
   (ii) char a = -129;
   printf("%d", a);
(d) Evaluate the following equation using operator precedence and associativity:
   int a = 5, b = 50, c;
   c = a + a - b/a + a*b + (a=a*a);
   printf("%d %d %d", a, b, c);
(e) Can the selection expression of a switch statement be a string? Justify.
(f) What is the difference between break and continue?
(g) What are the uses of pointers?
(h) Write a ‘C’ program to sum the N numbers using recursion.
(i) Is ‘main’ a keyword in C language? Justify your answer.
(j) What is mean by preprocessor? Where it is used?
SECTION – B

2. Attempt any three question parts: \(10 \times 3 = 30\)
   
   (a) Write short notes on the following:
       (i) System software and Application software.
       (ii) Machine language vs. higher level languages.
   
   (b) With relevant examples discuss the different operators in ‘C’ language.
   
   (c) Write a ‘C’ program to sum the elements above and below the main diagonal of a matrix using two dimensional arrays.
   
   (d) The annual examination is conducted for 50 students for three subjects. Write a ‘C’ program using structures to read the data and determine the following:
       (i) Total marks obtained by each student.
       (ii) The highest marks in each subject and the Roll No. of the student who secured it.
       (iii) The student who obtained the highest total marks.
   
   (e) What is a string? How it is represented in ‘C’ language? List some pre-defined string functions. Write a ‘C’ program for concatenation of two strings without using pre-defined functions.

SECTION – C

Attempt all questions. \(10 \times 5 = 50\)

3. Attempt any two parts: \(2 \times 5 = 10\)
   
   (a) Convert the following octal numbers into hexa-decimal numbers:
       (i) 234
       (ii) 456
       (iii) 111
       (iv) 25.33
       (v) 267.12
   
   (b) Draw the flowchart to sort a list of numbers in ascending order.
   
   (c) What is a Compiler? What is an Operating System? Also discuss the difference between compiler and linker.
4. Attempt any one part:  
(a) Define token, identifier and keyword. Explain about printf () and scanf () statement. Write a ‘C’ program to calculate the sum of positive integer numbers using scanf () and printf () statement.
(b) What is a function in ‘C’? Differentiate calling functions and called functions. With a neat example, discuss the concept of inter-function communication, passing arguments by value, scope rules and global variables.

5. Attempt any one part:  
(a) “C language provides three constructs for performing loop operations”. With relevant examples discuss the same.
(b) What is nested if..else statement? How it differs from if..else statement. Write a ‘C’ program using nested if-else statement to compute and print the income tax of each employee. For each employee, the employee number, name and his annual salary up to ₹ 30,000, there is no tax. If the annual salary is above ₹ 30,000 and up to ₹ 60,000, tax is computed as 20% of the salary. If the annual salary is above ₹ 60,000, for the amount up to ₹ 60,000 tax is computed as 20% of the amount and for the amount that exceeds ₹ 50,000 tax is computed as 30%.

6. Attempt any one part:  
(a) What is sequential search? Write a ‘C’ program to copy the content of one file to another file using FILE concept.
(b) Describe operations on pointers with example. Write a ‘C’ program to print the values before and after calling the functions using pointers as arguments.

7. Attempt any two parts:  
(a) Demonstrate the basic OS commands using LINUX for: creating, editing, saving and protecting access to files.
(b) How to edit and search text files in open-office based on LINUX OS environment?
(c) Write short on defining and calling macros, utilizing conditional compilation and passing values to the compiler.