U.S.N. [ ][ ][ ][ ][ ][ ][ ][ ][ ]

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

August 2023 Semester End Make-Up Examinations

Programme: B.E.
Branch: Common to All Branches
Course Code: 22CS1ESICP
Course: Introduction to C-Programming

Semester: I
Duration: 3 hrs.
Max Marks: 100
Date: 14.08.2023

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

UNIT - I

1 a) Discuss the rules to be followed for naming an identifier, such as a variable, in C. **4**

b) Convert the following equations into equivalent C-programming statements. (Use of minimum number of characters is preferred). **8**

(i) $p = a + \dfrac{b}{cx+d} - \left(\dfrac{f^2}{gx+\frac{h}{j}}\right)\dfrac{k}{l}$

(ii) $x = \dfrac{-b+\sqrt{b^2-4ac}}{2a}$

(iii) $y = e^{-xt}\left[\text{cosine of } \left(\dfrac{\pi x}{a}\right)\right](1+x)^n$

(iv) $result = x > y \,?\, 1 : 0$ (into equivalent if-else statement)

c) Write a program to read the coefficients of the quadratic equation $ax^2 + bx + c = 0$, compute and print the roots of the equation. The program must check for nature of roots (real/imaginary) and output the roots with suitable messages. **8**

UNIT - II

2 a) Differentiate *while* and *do-while* constructs writing their syntax. **6**

b) A prime number is a positive integer that is divisible by 1 and itself. Write a program to generate first 100 prime numbers. **8**

c) Write the output of the following program. **6**

```c
#include<stdio.h>
void main()
{
    int i, j, x =0;
    for (i=0; i<5; ++i)
        for(j=0; j<i; j++)
        {
            x + = (i + j -1);
            printf("%d\n", x);
            break;
        }
    printf("x = %d", x);
}
```

UNIT - III

3 a) A set of *n* integer numbers in descending order is given as input to a program. Write the source code (C-program) to read these numbers and search for a key element using binary search. The program must output suitable messages and identify the position of the element, if found.   **10**

    b) Two matrices *A* and *B* of order *m* × *n* are to be read as input. Write a program to read these two matrices.
Compute and output the following: matrix *A*, matrix *B*, *A* + *B*, *A* − *B*, transpose of (*A* + *B*).   **10**

UNIT - IV

4 a) Write a program that would satisfy following conditions: **10**
      (i)   Define two global integer variables *x* and *y*; two local integer variables *a* and *b*. (Use of additional variables is permitted).
      (ii)   Read values of *x*, *y*, *a*, and *b* in the main program; output their values before swapping.
      (iii) Swap values of *x* and *y* (using a function with call by value).
      (iv) Swap values of *a* and *b* (using another function with call by reference).
      (v)   Output the final values of four variables in the main function.

    b) Write a program to read two strings, concatenate the strings (second string appended to first string without the use of built-in function) and output the two strings.   **10**

OR

5 a) Write a user-defined function to compute factorial of a given integer. Use this function in the main program that reads *n* and *r* as integer input, computes ^nP_r and ^nC_r using the function and output the computed values in suitable format.   **10**

$$\text{Note: } ^nP_r = \frac{n!}{(n-r)!} \; ; \; ^nC_r = \frac{n!}{(n-r)! \, r!}$$

    b) Write a program to read a string, reverse the given string without use of built-in function, and check whether it is a palindrome.   **10**

UNIT - V

6 a) Write the syntax of structure definition and structure declaration and explain them with an example to store employee details by having structure members: emp-id, emp-name , salary and display the same.   **10**

    b) Write the output of the following program. You are required to justify the output through proper description for each output statement.   **10**

```
#include<stdio.h>
void function1(int u, int v);
void function2(int *pu, int *pv);
void main()
{
    int u=1, v=2;
    printf("Before calling function1: u = %d, v = %d",u,v);
    function1(u,v);
    printf("\nAfter calling function1: u = %d, v = %d",u,v);

    printf("\nBefore calling function2: u = %d, v = %d",u,v);
    function2(&u,&v);
    printf("\nAfter calling function2: u = %d, v = %d",u,v);
}
void function1(int u, int v)
{
    u = 0; v = 0;
    printf("\nWithin function1: u = %d, v = %d",u,v);
    return;
}
void function2(int *pu, int *pv)
{
    *pu = 3; *pv = 4;
    printf("\nWithin function2: *pu = %d, *pv = %d",*pu,*pv);
    return;
}
```

OR

7   a)   Define a structure called *cricket* that will describe the following information: **10**
     player_name, team_name, batting_average.
     Using *cricket*, declare an array *player* with 50 elements and write a program to
     read the information about all the 50 players and print a list containing names
     of players with their batting average.

    b)   Write a function using pointers to exchange the values stored in two locations   **10**
     in the memory. Use this function in a main program that reads two variables
     and exchanges their values. The program must output the original values and
     final values with suitable messages.

******