Name: **Enrolment No:**	

UPES

End Semester Examination, December 2023

Course: Programming in C — **Semester : 1**

Program: B.Tech (CSE)/B.Sc. (CSE)/BCA — **Time : 03 hrs.**

Course Code: CSEG 1025 — **Max. Marks: 100**

Instructions: Attempt all the questions.

SECTION A
(5Qx4M=20Marks)

S. No.		Marks	CO
Q 1	Write down at least two library functions from each of the header files mentioned below along with their respective functionality in just one line. ***{ stdlib.h, ctype.h, math.h, time.h, string.h}***	**4**	**CO5**
Q 2	Differentiate between preprocessor and debugger.	**4**	**CO1**
Q 3	State the difference between pointer and pointer-to-pointer variable.	**4**	**CO3**
Q 4	Perform the following conversion: a) $(108)_{10} = (?)_2$ b) $(FEA)_{16} = (?)_{10}$ c) $(7521)_8 = (?)_2$ d) $(A7E)_{16} = (?)_{10}$	**4**	**CO1**
Q 5	Discuss your understanding of a function-pointer using suitable code snippet.	**4**	**CO3**

SECTION B
(4Qx10M= 40 Marks)

Q 6	State the difference between recursion and iteration. Illustrate the use of recursion by writing a C program that multiplies two integers without using multiplication operator. (4 + 6 = 10 Marks)	**10**	**CO2**
Q 7	Define algorithm and discuss its significance in the development of efficient programs. Write down the algorithm that determines if the two user provided numbers are coprime to one another. (2 + 3 + 5 = 10 Marks)	**10**	**CO1**
Q 8	Discuss the various file opening modes in C. Explain the process of opening a file for read and write operations through an appropriate C program. (5 + 5 = 10 Marks)	**10**	**CO4**
Q 9	Write a C program that creates a structure with two fields referring to x and y coordinates in a 2-D Cartesian space. Create four variables denoting four points in the plane. Determine the Eucleadian distance between these points through a user-defined function which accepts the aforesaid structure variables. Based on the length of sides computed by this function, you should invoke another function from within the current function that determines if the quadrilateral is an equilateral one. OR Write a menu driven program that performs the following operations:	**10**	**CO3**

	(i) Addition of two complex numbers (ii) Subtraction of two complex numbers (iii) Multiplication of two complex numbers [Note: Use structures to represent a complex number.] [3 + 3 + 4 =10 Marks]		
SECTION-C (2Qx20M=40 Marks)			
Q 10	Differentiate between the following using suitable code snippets: a) Type conversion & type casting b) return and exit statements c) Local and global variables d) Static and shared libraries e) Static and dynamic memory allocation **OR** Write short notes on the following while explaining the related concepts using suitable code snippets: a) Structure member access operators- '.' & '->' b) Preprocessing directives c) Limitations of union d) Data processing in multidimensional array e) Dynamic memory allocation (4 x 5 = 20 Marks)	**20**	**CO2 + CO3+ CO4**

Q 11	a) Provide the blank spaces with suitable entries in the below-mentioned code snippets to achieve the prescribed outputs:
	(i) `#include<_ _ _ _ _ _.h>`
	`void main()`
	`{`
	`int i1=100;`
	`float f1=200.5;`
	`_ _ _ _ vptr;`
	`vptr=&i1;`
	`printf("i1 contains %d\n", *(_ _ _ _ _ vptr));`
	`shiva=&f1;`
	`printf("f1 contains %0.0f\n", *(_ _ _ _ _ vptr));`
	`}`
	
	Output: i1 contains 100
	f1contains 200
	
	(ii) `#include<_ _ _ _ _ _ _>`
	`#include< _ _ _ _ _ _ _ _>`
	`void main()`
	`{`
	`char str1[]="UPES", str2[20]="Dehradun";`
	`int i=_ _ _ , j= _ _ _ ;`
	`puts( _ _ _ _ );`
	`puts( _ _ _ _ );`
		20	**CO2 + CO3**

```
        while(str1[i]!= _ _ _ )
        {
                i++;
         }
        while(str2[j]!= _ _ _ )
         {
           str1[i]=str2[j];
           j++;
           i++;
         }
        str1[i]= _ _ _ _;
        printf("%s\n",str1);
    }
```

Output:	UPES Dehradun UPESDehradun

(5 + 5 = 10 Marks)

b) If required, rectify the code snippets provided below and predict the correct output. (Assume that all the necessary header files are already included.)

(i)
```
int main()
{
    int arr[] = {10, 20, 30, 40, 50};
    int *ptr = arr + 2;
    printf("%d\n", *ptr++);
    printf("%d\n", (*ptr)++);
     printf("%d\n", *ptr); // What is the output here?
    return 0;
}
```

(ii)
```
int multiply(int a, int b)
{
    return a * b;
}
int main()
{
    int (*operation)(int, int)=&multiply;
    int result = operation(4, 7);
    printf("Result: %d\n", result);
    return 0;
}
```

(iii)
```
void main()
    {
      int A;
     A=2*23,1+10,2*35;
     printf("\n\n A is\t%d\n",A);
    }
```

(iv)
```
void main()
    {
            char first[]="Mumbai";
            char sec[5]="Delhi";
            char third[7]="Compute";
            printf("%c\t%c\tEND1",first[5],sec[4]);
            printf("\n%c\t%c\tEND2\n%c\n",first[6],third[7],'\0');
  }
```

(2.5 x 4 = 10 Marks)