M. Sc. (MATHEMATICS WITH APPLICATIONS IN COMPUTER SCIENCE) [M. Sc. (MACS)]

Term-End Examination

June, 2023

MMT-001 : PROGRAMMING AND DATA STRUCTURES

Time : $1\frac{1}{2}$ Hours Maximum Marks : 25

(Weightage : 20%)

Note : Question No. **1** is **compulsory**. Answer any **three** questions from Q. Nos. **2** to **5**. All programs should be written in 'C' language only. Use of calculator is **not** permitted.

P. T. O.

---

1. Write the output of the following segments of code in 'C'. Justify your answers with short explanations : 10

(i) int main()
    {int a=2, b=3;
    printf("%d + %d = %d", a,b,a+b);
    return 0;
    }

(ii) int main()
    { int i=10;
    { int i=20;
    printf("%d \n", i);
    }
    printf("%d", i);
    return 0;
    }

(iii) int main()
    { int i, j;
    for(i=0; i<=4; i++)
    {
    for(j=0; j<=i; j++)
    printf("%d", j);
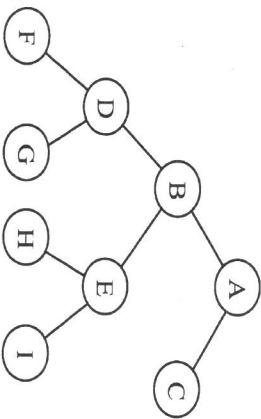   printf ("\n");
    }
    return 0;
    }

(iv) int main()

```
{ char s[10];
int i;
for (i=0; i<10; i++)
    s[i] = i + 65;
for (i = 0; i<10; i++)
    printf("%c", s[i]);
}
```

(v) main()

```
{ char string[] = "Hello world";
    display (string);
}
void display (char * string)
{
    printf("%s", string);
}
```

2. (a) Write a recursive function to compute the factorial of a number. 2

(b) Write preorder and postorder traversal of the binary tree, given below : 3

3. (a) What is a macro ? How does it differ from a function ? Explain with examples. 2

(b) Write the syntax for defining a node of a doubly linked list containing floating point data. Also write a function create-node() which creates a new node and returns the pointer to it. 3

4. (a) List and compare the static data structures with the dynamic data structures. 2

(b) How does Binary Tree differ from Binary Search Tree ? List the operations that can be performed on a Binary Search Tree. 3

5. (a) Write printf statements in C language for printing the number 289.342, using : 2

(i) 8 places, right justified;

(ii) 8 places, left justified upto two decimal digits

(b) Explain, with an example for each, the difference between the logical and (&&) and the bitwise and (&) operators. 3