Reg.No:

SNS College of Technology, Coimbatore-35.

(Autonomous)

B.E/B.Tech- Internal Assessment -II

Academic Year 2023-2024(EVEN)

Second Semester (Regulation R2023)

23ITT101 – PROGRAMMING IN C AND DATA STRUCTURES

Common to Aero, Auto, Agri, Mech, FT, MCT, Civil

Answer Key

A

Time: $1^{1/2}$ Hours **Maximum Marks: 50**

Answer All Questions

PART A — (5 x 2 = 10 Marks)

1. List out the four types of function declaration — CO2 — REM
   Without Arguments and without Return Value
   Without Arguments and with Return Value
   With Arguments and without Return Value
   WithArguments and with Return Value
2. Distinguish between Call by value and call by reference — CO2 — ANA
   In call by value, the values of the actual parameters copy into the function’s formal parameters. In call by reference the address of the the actual parameters copy into the function’s formal parameters
   In Call by Value, the actual arguments and passed parameters of a function refer to different memory locations. In Call by Reference, the actual arguments and passed parameters of a function refer to the same memory location
3. Write the logic for performing sorting of numbers in ascending order. — CO3 — APP

```
for (i = 0; i < n; ++i){
   for (j = i + 1; j < n; ++j){
      if (num[i] > num[j]){
         a = num[i];
         num[i] = num[j];
         num[j] = a;
      }
```

4. What is Pointer? How a variable is declared to the pointer? — CO3 — UND
   A pointer is a variable that stores the memory address of another variable as its value.
   The syntax of pointers is similar to the variable declaration in C, but we use the (*) dereferencing operator in the pointer declaration.

5. What do you mean by non-linear data structure? Give example CO3 REM
Data structures where data elements are not arranged sequentially or linearly are called **non-linear data structures**. In a non-linear data structure, single level is not involved.
Examples: Trees and Graphs

PART B — (2 x 13 = 26 Marks & 1 x 14 = 14 Marks)

6. (a) Define function and Write a C program for arithmetic operations by using with argument with return type method and without argument, without return type method CO2 APP 13

With arguments and with return value

```
void main()
{
        int a=6,b=7,c;
        c=add(a,b);
        printf("%d",c);
        getch();
}
int add(int i,int j)
{
        int k;
        k=i+j;
        return(k);
}
```

Without arguments and without return value

```
Void main()
{
        void add(void);
        add();
        getch();
}
void add()
{
        int a,b,c;
        printf("Enter a &b");
        scanf("%d %d",&a, &b);
        c=a+b;
        printf("%d",c);
}
```

(OR)

(b) Explain different classification of user defined functions based on parameter passing and return type with examples CO2 UND 13

FUNCTION PROTOTYPE(fn Name, args,return value)

Prototype helps to check the arguments & return value of the function

Types of Function

1.Without arguments and return value
2.Without arguments and with return value
3.With arguments and no return value
4.With arguments and with return value

```
Void main()
{
        void add(void);
        add();
        getch();
}
void add()
{
        int a,b,c;
        printf("Enter a &b");
        scanf("%d %d",&a, &b);
        c=a+b;
        printf("%d",c);
}
```

7. (a) Illustrate one dimensional array and its characteristics and Construct a C program to calculate the Sum of array elements.

The collection of data items can be stored under a one variable name using only one subscript such a variable is called one dimensional array.

Syntax:

datatype arrayname[Size];

Example:

int num[5];

One Dimensional array Initialization

After an array is declared its elements must be initialized .Otherwise it will contains an garbage values.

The array can be initialized by 2 stages

- ✓ At Compile time
- ✓ At Run time

➢ **At Compile time Initialization Syntax:**

datatype arrayname[Size]={list of values};

➢ **At Run time Initialization**

Arrays cane be explicitly initialized at run time. Usually applied for larger arrays

```
Example:
 for(i=0;i<100;i=i+1)
{

if(i<50)
{
sum[i]=0;

}
else
{
 sum[i]=1;
}
}
```

CO3 UND 13

1.Sum of Array Elements

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i,sum=0,arr[5];
printf("Enter Array Elements");
for(i=0;i<5;i++)
{
scanf("%d",&arr[i]);
}
for(i=0;i<5;i++)
{
sum=sum+arr[i];
}
printf("Sum of Array of Elements%d",sum);
}
```

(OR)

(b) Write a C program to perform matrix addition for the matrix size 3 X 3.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,a[3][3],b[3][3],c[3][3];
clrscr();
printf("\n Enter the elements of A Matrix");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("\n Enter the elements of B Matrix");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
scanf("%d",&b[i][j]);
}
}
printf("\n C Matrix \n");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
c[i][j]=a[i][j]+b[i][j];
printf("%d\n",c[i][j]);
}
}
}
```

CO3 APP 13

8. (a) Construct the c program for swapping 2 numbers with temporary variable using call by value method with sample output

Call by value

```
#include<stdio.h>
#include<conio.h>
void swap(int ,int );
void main()
{
int a=10 ,b=20;
clrscr();
printf("%d %d \n",a,b);
swap( a, b);
printf("after swap% d %d",a,b);
}
void swap(intx,int  y)
{
int temp=x;
x=y;
y=temp;
printf("%d %d \n",x,y);
}
```

CO2 APP 14

(OR)

(b) What is purpose of the ‘Structure’ in language C? Explain in detail with an example program. CO3 APP 14

Structure is a user-defined data in C language will allows us to combine data of different types together.

Structure helps to construct a complex data type which is more meaningful.

```
#include <stdio.h>
/* Created a structure here. The name of the structure is
 * StudentData.
 */
struct StudentData{
    char *stu_name;
    int stu_id;
    int stu_age;
};
int main()
{
     /* student is the variable of structure StudentData*/
     struct StudentData student;

     /*Assigning the values of each struct member here*/
     student.stu_name = "Steve";
     student.stu_id = 1234;
     student.stu_age = 30;

     /* Displaying the values of struct members */
     printf("Student Name is: %s", student.stu_name);
```

****

(Note: Und-Understand Rem-Remember Ana-Analyze App-Apply)

Prepared By **Verified By** **HoD**