Practical 7

C programming

//a. Call by value program for swapping of two numbers

```
#include <stdio.h>
void swap (int a, int b)
{
        int temp;
        temp = a;
        a = b;
        b = temp;
        printf("After swapping first number is %d and second number is %d", a ,b);
}
int main(void)
{
        int first, second;
        printf("Enter two numbers : \n"); scanf("%d %d",&first,&second);
        swap(first,second);
        /* Check whether actual parameters is changed after swapping.*/
        printf(" \n After swap function called first number is %d and second number is %d",
         first ,second);
        return 0;
}
```

//b. Call by reference program for swapping of two numbers

```
#include <stdio.h>
/* Swap function declaration */
void swap(int * num1, int * num2);
void main()
{
        int num1, num2;
```

```
	/* Input numbers */
	printf("Enter two numbers: ");
	scanf("%d%d", &num1, &num2);

	/* Print original values of num1 and num2 */
	printf("Before swapping in main n");
	printf("Value of num1 = %d \n", num1);
	printf("Value of num2 = %d \n\n", num2);

	/* Pass the addresses of num1 and num2 */
	swap(&num1, &num2);
	/* Print the swapped values of num1 and num2 */
	printf("After swapping in main n");
	printf("Value of num1 = %d \n", num1); printf("Value of num2 = %d \n\n", num2);
	return 0;
}
/*Function to swap two numbers*/
void swap(int * num1, int * num2)
{
	int temp;
	// Copy the value of num1 to some temp variable temp = *num1;
	// Copy the value of num2 to num1
	*num1= *num2;
	// Copy the value of num1 stored in temp to num2
	*num2= temp;
	printf("After swapping in swap function n"); printf("Value of num1 = %d \n", *num1);
	printf("Value of num2 = %d \n\n", *num2);
}
```

Practical 8

//a. Write a program to read a matrix of size m*n

```
#include<stdio.h>
int main()
{
	int i,j,m,n;
	float a[10][10];

	printf("Enter row and column size:\n");
	scanf("%d%d", &m, &n);

	printf("Enter matrix elements:\n");
	for(i=0;i< m;i++)
	{
		for(j=0;j< n;j++)
		{
			printf("a[%d][%d]=",i,j);
			scanf("%d", &a[i][j]);
		}
	}
	printf("Matrix read is:\n");
	for(i=0;i< m;i++)
	{
		for(j=0;j< n;j++)
		{
			printf("%d\t",a[i][j]);
		}
		printf("\n");
	}
}
```

```
// C program to multiply two matrices
#include <stdio.h>
#include <stdlib.h>
// Edit MACROs here, according to your Matrix Dimensions for
// mat1[R1][C1] and mat2[R2][C2]
#define R1 2 // number of rows in Matrix-1
#define C1 2 // number of columns in Matrix-1
#define R2 2 // number of rows in Matrix-2
#define C2 2 // number of columns in Matrix-2

void mulMat(int mat1[][C1], int mat2[][C2])
{
   int rslt[R1][C2];

   printf("Multiplication of given two matrices is:\n");

   for (int i = 0; i < R1; i++) {
      for (int j = 0; j < C2; j++) {
         rslt[i][j] = 0;

         for (int k = 0; k < R2; k++) {
            rslt[i][j] += mat1[i][k] * mat2[k][j];
         }
         printf("%d\t", rslt[i][j]);
      }
      printf("\n");
   }
}
// Driver code
int main()
{
   // R1 = 4, C1 = 4 and R2 = 4, C2 = 4 (Update these
```

```
    // values in MACROs)
    int mat1[R1][C1] = { { 1, 1 },
                         { 2, 2 } };

    int mat2[R2][C2] = { { 1, 1 },
                         { 2, 2 } };

    if (C1 != R2) {
        printf("The number of columns in Matrix-1  must be "
               "equal to the number of rows in "
               "Matrix-2\n");
        printf("Please update MACROs value according to "
               "your array dimension in "
               "#define section\n");

        exit(EXIT_FAILURE);
    }
    // Function call
    mulMat(mat1, mat2);
    return 0;
}
```