Name: Tanzeela Asghar

Reg no: 2021 BSE 032

Section no: III-A

Course: Data Structures

Submitted to: Sir Rehan Ahmed Siddiqui

Lab 1

TASK#2

1. **Write the prototype of a function named DS() which accepts an array of integers, a pointer to double, a float by reference and returns a character.**

char DS(int arr[],double *ptr, float &r)

2. **Write statements as directed:**

int a[]={1,2,3,4,5};

int *p;

p = a;

//Print the elements of array using the mentioned notation:

for (int i = 0; i < 5; i++)

{

//Subscript notation with name of array

cout<<a[i]<<endl;

//Subscript notation with pointer 'p'

cout<<*p[i]<<endl;

//Offset notation using array name

cout<<*(a+i)<<endl;

//Offset notation using pointer 'p'

cout<<*(p+i)<<endl;

3. **What is the difference between function overloading and function overriding?**

Function Overloading is when multiple function with same name exist in a class. Function Overriding is when function have same prototype in base class as well as derived class.

Overloaded functions have different function signatures. Overridden functions have the same function signatures.

4. **Write C++ statement(s) to allocate space for 10 doubles (using dynamic memory allocation).**

Double *arr = new double[10];

5. **OUTPUT**

1

2

3

TASK#2

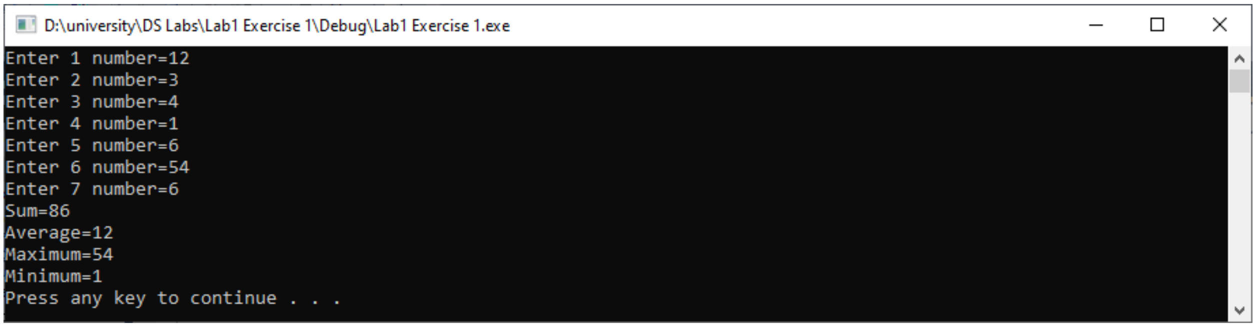
Exercise#1

Code:

```
#include "stdafx.h"
#include<iostream>
using namespace std;
int sum(int num[],int n)
{
	int sum=0;
	for(int i=0;i<n;i++)
	{
		sum=sum+num[i];
	}
	return sum;
}

int avg(int num[],int n)
{
	int sum=0,avg;
	for(int i=0;i<n;i++)
	{
		sum=sum+num[i];
	}
	avg=sum/7;
	return avg;
}
int min(int num[],int n)
{
	int min=num[0];
	for(int i=0;i<n;i++)
	{
		if(min>num[i])
			min=num[i];
	}
	return min;
}
int max(int num[],int n)
{
	int max=num[0];
	for(int i=0;i<n;i++)
	{
        if(max<num[i])
            max=num[i];
    }
    return max;
}
int _tmain(int argc, _TCHAR* argv[])
{
    int size=7;
    int *n=new int[size];
    for(int i=0;i<size;i++)
    {
        cout<<"Enter "<<i+1<<" number=";
        cin>>n[i];
    }
    cout<<"Sum="<<sum(n,size)<<endl;
    cout<<"Average="<<avg(n,size)<<endl;
    cout<<"Maximum="<<max(n,size)<<endl;
    cout<<"Minimum="<<min(n,size)<<endl;
    system("pause");
    return 0;
}
```

OUTPUT


D:\university\DS Labs\Lab1 Exercise 1\Debug\Lab1 Exercise 1.exe

```
Enter 1 number=12
Enter 2 number=3
Enter 3 number=4
Enter 4 number=1
Enter 5 number=6
Enter 6 number=54
Enter 7 number=6
Sum=86
Average=12
Maximum=54
Minimum=1
Press any key to continue . . .
```

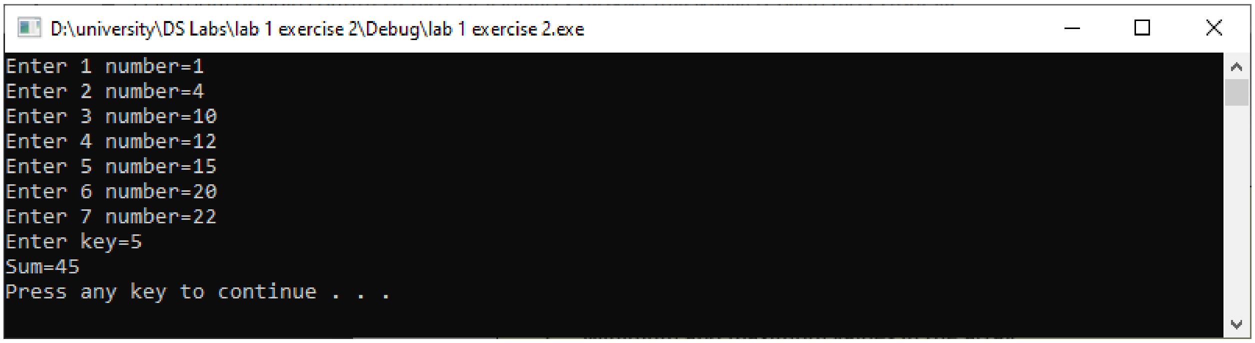
Exercise#2

Code:

```
#include "stdafx.h"
#include<iostream>
using namespace std;

int sum(int arr[7],int key)
{
    int sum=0;
    for(int i=0;i<7;i++)
    {
        if(arr[i]%key==0)
        {
            sum=sum+arr[i];
        }
    }
    return sum;
}
int _tmain(int argc, _TCHAR* argv[])
{
    int arr[7],key;
    for(int i=0;i<7;i++)
    {
        cout<<"Enter "<<i+1<<" number=";
        cin>>arr[i];
    }
    cout<<"Enter key=";
    cin>>key;
    cout<<"Sum="<<sum(arr,key)<<endl;
    system("pause");
    return 0;
}
```

OUTPUT


D:\university\DS Labs\lab 1 exercise 2\Debug\lab 1 exercise 2.exe

```
Enter 1 number=1
Enter 2 number=4
Enter 3 number=10
Enter 4 number=12
Enter 5 number=15
Enter 6 number=20
Enter 7 number=22
Enter key=5
Sum=45
Press any key to continue . . .
```

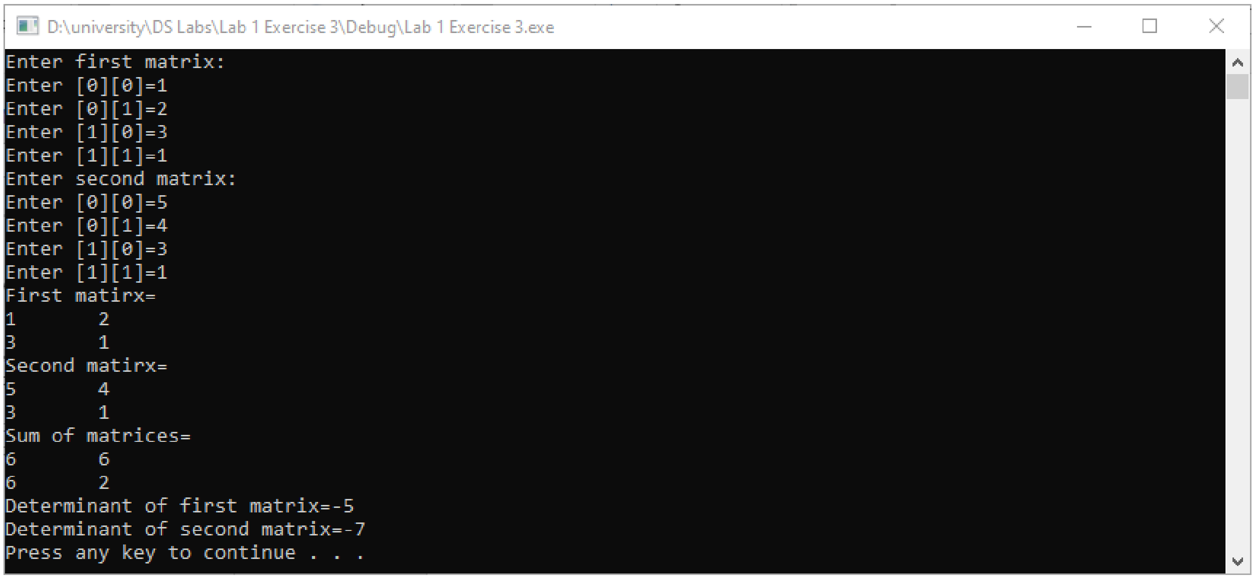
Exercise#3

Code:

```
#include "stdafx.h"
#include<iostream>
using namespace std;
class Matrix
{
    int m[2][2];
public:
    Matrix()
    {
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                m[i][j]=0;
            }
        }
    }
    void get_input()
    {
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                cout<<"Enter ["<<i<<"]["<<j<<"]=";
                cin>>m[i][j];
            }
        }
    }
    void display()
    {
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                cout<<m[i][j]<<"\t";
            }
            cout<<"\n";
        }
    }
    Matrix operator +(Matrix m2)
    {
        Matrix m3;
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                m3.m[i][j]=m[i][j]+m2.m[i][j];
            }
        }
        return m3;
    }
    int operator ~()
    {
        int determinant;
        determinant=(m[0][0]*m[1][1])-(m[0][1]*m[1][0]);
        return determinant;
    }
};
int _tmain(int argc, _TCHAR* argv[])
```

```
{
        Matrix m1,m2,m3;
        cout<<"Enter first matrix:"<<endl;
        m1.get_input();
        cout<<"Enter second matrix:"<<endl;
        m2.get_input();
        m3=m1+m2;
        cout<<"First matirx="<<endl;
        m1.display();
        cout<<"Second matirx="<<endl;
        m2.display();
        cout<<"Sum of matrices="<<endl;
        m3.display();
        cout<<"Determinant of first matrix="<<~m1<<endl;
        cout<<"Determinant of second matrix="<<~m2<<endl;
        system("pause");
        return 0;
}
```

OUTPUT


D:\university\DS Labs\Lab 1 Exercise 3\Debug\Lab 1 Exercise 3.exe

```
Enter first matrix:
Enter [0][0]=1
Enter [0][1]=2
Enter [1][0]=3
Enter [1][1]=1
Enter second matrix:
Enter [0][0]=5
Enter [0][1]=4
Enter [1][0]=3
Enter [1][1]=1
First matirx=
1       2
3       1
Second matirx=
5       4
3       1
Sum of matrices=
6       6
6       2
Determinant of first matrix=-5
Determinant of second matrix=-7
Press any key to continue . . .
```