Name-**Diya Goyal** Roll no-**102215255** Sub-group-**2NC11**

Lab Assignment 1

UCS 406 Data Structures and Algorithms

Note: Use C/C++ or JAVA programming language.

Q1. Write a program that

i. computes and returns the sum of all the integers between "first" and "last" inclusive.

```cpp
// i.   computes and returns the sum of all the integers between "first" and "last" inclusive.

#include<iostream>
using namespace std;

int main(){
    int first;
    int last;
    cin>>first;
    cin>>last;
    int sum=0;
    if(first>last){
        for(int i=last;i<=first;i++){
            sum=sum+i;
        }
    }
    else{
        for (int i = first; i <=last; i++)
        {
            sum=sum+i;
        }

    }
    cout<<"the sum is"<<sum<<endl;
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
3
9
the sum is42
```

ii. computes and returns the smallest positive integer n for which 1+2+3+...+n equals or exceeds the value of "goal".

```cpp
// ii.  computes and returns the smallest positive integer n for which 1+2+3+...+n equals or exceeds the value of "goal".

#include<iostream>
using namespace std;

int main(){
    int n=1,goal;
    int sum=0;
    cin>>goal;
    while(sum<goal){
        sum=sum+n;
        if(sum==goal){
            break;
        }
        n++;
    }
    cout<<n;
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
5
4
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>
```

iii. computes and returns the greatest common divisor (g.c.d.) of the arguments passed to it.

```cpp
//iii.  computes and returns the greatest common divisor (g.c.d.) of the arguments passed to it.

#include<iostream>
using namespace std;

int main(){
    int n1,n2,hcf;
    cout<<"enter 2 numbers: ";
    cin>>n1>>n2;
    if(n2>n1){
        int temp = n2;
        n2=n1;
        n1=temp;
    }
    for(int i=1;i<=n2;i++){
        if(n1%i==0 && n2%i==0){
            hcf=i;
        }
    }
    cout<<"hcf is: "<<hcf;
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
enter 2 numbers: 3 4
hcf is: 1
```

iv. Determines whether an integer is prime.

```cpp
//iv.   Determines whether an integer is prime.

#include<iostream>
using namespace std;

int main(){
    int n;
    cout<<"enter an integer:";
    cin>>n;
    bool prime;
    if(n==0 || n==1){
        prime= false;
    }
    for(int i=2; i<=n/2;i++){
        if(n%i==0){
            prime= false;
            break;
        }

    }
    if(prime){
        cout<<n<<" is not a prime number"<<endl;
    }
    else{
        cout<<n<<" is a prime number"<<endl;

    }
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
enter an integer:5
5 is a prime number
```

v. Prints the English name of an integer from 1 to 9.

```
#include<iostream>
using namespace std;

int main(){
    int n;
    cout<<"enter a number between 1 to 9: ";
    cin>>n;
    switch (n)
    {
    case 1:
        cout<<"one";
        break;
     case 2:
        cout<<"two";
        break;
     case 3:
        cout<<"three";
        break;
    case 4:
        cout<<"four";
        break;
    case 5:
        cout<<"five";
        break;
    case 6:
        cout<<"six";
        break;
    case 7:
        cout<<"seven";
        break;
    case 8:
        cout<<"eight";
        break;
    case 9:
        cout<<"nine";
        break;

        break;

    default:
        break;
    }
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
enter a number between 1 to 9: 5
five
```

vi. Reverses the order of the objects in an array.

```cpp
//vi.   Reverses the order of the objects in an array.

#include<iostream>
using namespace std;
    void reverse(int arr[],int n){
        int start=0;
        int end = n-1;
        while (start<=end)
        {
            swap(arr[start],arr[end]);
            start++;
            end--;
        }

    }
    void printArray(int arr[], int n){
        for (int i = 0; i < n; i++)
        {
            cout<<arr[i]<<" ";
        }
        cout<<endl;

    }
int main(){
    int arr[6]={1,2,3,4,5,6};
    reverse(arr,6);
    printArray(arr,6);

    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
6 5 4 3 2 1
```

vii. Finds the index of the largest number in an array.

```cpp
//vii.  Finds the index of the largest number in an array.


#include<iostream>
using namespace std;
int largest(int arr[], int size){
    int max=arr[0];
    int i;
    int pos;
    for( i=0;i<size;i++){
        if(max<arr[i]){
            max=arr[i];
            pos=i;
        }

    }
    cout<<pos;
}

int main(){
    int n;
    cin>>n;
    int arr[n];
    for(int i=0;i<n;i++){
        cin>>arr[i];
    }
    for(int i=0;i<n;i++){
        cout<<arr[i];
    }
    cout<<endl;
    largest(arr, n);
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
6
1 2 3 4 5 6
123456
```

viii. Shifts the contents of array cells one cell to the right, with the last cell's contents moved to the left end.

```cpp
#include<iostream>
using namespace std;
void shift(int arr[], int size){
    int temp=arr[size-1];
    for( int i = size-1; i>0;--i){
        arr[i]=arr[i-1];
    }
    arr[0]=temp;
}
int printArray(int arr[], int size){
    for(int i=0;i<size;i++){
        cout<<arr[i]<<" ";
    }
}


int main(){
    int n;
    cin>>n;
    int arr[n];
    for(int i=0;i<n;i++){
        cin>>arr[i];
    }
    for(int i=0;i<n;i++){
        cout<<arr[i];
    }
    cout<<endl;
    shift(arr,n);
    printArray(arr,n);
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
5
1 2 3 4 5
12345
5 1 2 3 4
```

ix. Examines an array of integers and eliminates all duplication of values. The distinct integers are all moved to the left part of the array.

```cpp
#include<iostream>
using namespace std;

int removeDuplicates(int arr[], int n){
    if(n==0 || n==1){
        return n;
    }
    int temp[n];
    int j=0;
    for (int i = 0; i < n; i++)
    {
        if(arr[i]!= arr[i+1]){
            temp[j++]=arr[i];
        }
    }
    temp[j++]=arr[n-1];
    for(int i=0;i<j;i++){
        arr[i]= temp[i];
    }
    return j;


}

int main(){
    int arr[]= {1,2,2,3,4,4,5,6,7};
    int n= sizeof(arr)/sizeof(arr[0]);
    n = removeDuplicates(arr, n);
    for (int i = 0; i < n; i++)
    {
        cout<<arr[i]<<" ";
    }

    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
1 2 3 4 5 6 7 7
```

x. Copies numbers from two arrays into a third array. The numbers from the second array are placed to the right of the numbers copied from the first array.

```cpp

#include<iostream>
using namespace std;

int main(){
    int X[]= {1,2,3};
    int Y[]= {4,5,6};
    int m= sizeof(X)/sizeof(X[0]);
    int n= sizeof(Y)/sizeof(Y[0]);
     int arr[m+n];
     for (int i = 0; i < m+n; i++)
     {
         if(i<m){
             arr[i]= X[i];
         }
         else{
             arr[i]=Y[i-m];
         }
     }
     for (int i = 0; i < m+n; i++)
     {
         cout<<arr[i]<<" ";
     }


     return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
1 2 3 4 5 6
```

Q2. Write a program to find sum of all elements of an array; write a program to find maximum of elements of an array; write a program for linear search of an array.

```cpp
#include<iostream>
using namespace std;


int sumofelements(int arr[],int size){
    int sum=0;

    for(int i=0;i<size;i++){
        sum=sum+arr[i];
    }

    return sum;
}


int main(){
    int arr[]={1,2,3,4,5,6,7};
    int size=7;

    cout<<"sum of all elements ="<<(sumofelements(arr,size));
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
sum of all elements =28
```

Q3. Write a program for adding and subtracting Matrices using 2D Arrays.

```cpp
#include<iostream>
using namespace std;


void subtractingmatrix(){
    int n;
    cout<<"Enter the size of the matrix: ";
    cin>>n;
    int arr1[n][n], arr2[n][n], arr3[n][n];
    cout<<"Enter the elements of the first matrix: ";
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            cin>>arr1[i][j];
        }
    }
    cout<<"Enter the elements of the second matrix: ";
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            cin>>arr2[i][j];
        }
    }
    cout<<"The subtraction of the two matrices is: ";
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            arr3[i][j] = arr1[i][j] - arr2[i][j];
            cout<<arr3[i][j]<<" ";
        }
        cout<<endl;
    }
}

int main(){
    subtractingmatrix();
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
Enter the size of the matrix: 2
Enter the elements of the first matrix: 1 2 3 4
Enter the elements of the second matrix: 5 6 7 8
The subtraction of the two matrices is: -4 -4
-4 -4
```

Q4. Write a program to Multiply Matrices of 10X10.

```cpp
#include <iostream>
using namespace std;
int main()
{ int a[10][10], b[10][10], mult[10][10], r1, c1, r2, c2, i, j, k;
    cout << "Enter rows and columns for first matrix: ";
    cin >> r1 >> c1;
    cout << "Enter rows and columns for second matrix: ";
    cin >> r2 >> c2;
    while (c1!=r2) {
        cout << "Error! column of first matrix not equal to row of second.";
        cout << "Enter rows and columns for first matrix: ";
        cin >> r1 >> c1;
        cout << "Enter rows and columns for second matrix: ";
        cin >> r2 >> c2;
    }    cout << endl << "Enter elements of matrix 1:" << endl;
    for(i = 0; i < r1; ++i)
        for(j = 0; j < c1; ++j)  {
            cout << "Enter element a" << i + 1 << j + 1 << " : ";
            cin >> a[i][j];
        }    cout << endl << "Enter elements of matrix 2:" << endl;
    for(i = 0; i < r2; ++i)
        for(j = 0; j < c2; ++j) {
            cout << "Enter element b" << i + 1 << j + 1 << " : ";
            cin >> b[i][j];
        }    for(i = 0; i < r1; ++i)
        for(j = 0; j < c2; ++j) {
            mult[i][j]=0;
        }    for(i = 0; i < r1; ++i)
        for(j = 0; j < c2; ++j)
            for(k = 0; k < c1; ++k)  {
                mult[i][j] += a[i][k] * b[k][j];
            }    cout << endl << "Output Matrix: " << endl;
    for(i = 0; i < r1; ++i)
    for(j = 0; j < c2; ++j){
        cout << " " << mult[i][j];
        if(j == c2-1)
            cout << endl;
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
Enter rows and columns for first matrix: 2 2
Enter rows and columns for second matrix: 2 2

Enter elements of matrix 1:
Enter element a11 : 1 2 3 4
Enter element a12 : Enter element a21 : Enter element a22 :
Enter elements of matrix 2:
Enter element b11 : 5 6 7 8
Enter element b12 : Enter element b21 : Enter element b22 :
Output Matrix:
 19 22
 43 50
```

Q5. Write a program for Linear Search using Functions.

```cpp
#include<iostream>
using namespace std;

int search(int arr[], int n, int key){
    for(int i=0;i<n;i++){
        if(arr[i]==key){
            return i;
        }
    }
    return -1;
}
int main(){
int n;
cin>>n;
int arr[n];
for(int i=0;i<n;i++){
    cin>>arr[i];
}
for(int i=0;i<n;i++){
    cout<<arr[i];
}
cout<<endl;
cout<<"enter key:"<<endl;
int key;
cin>>key;
cout<<search(arr, n, key);
return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a.exe
6
1 2 3 4 5 6
123456
enter key:
5
4
```

Q6. Write a class for a rectangle. Create objects for this class and call the member functions to find the area and perimeter of the rectangle.

```cpp
#include <iostream>
using namespace std;
class Rectangle{
    private:
        int l, b;
    public:
        void input(int len, int bre){
            l = len;
            b = bre;
        }

        int area(){
            return l * b;
        }
        int perimeter(){
            return 2*(l+b);
        }
};
int main(){
    Rectangle r1, r2;
    r1.input(10, 9);
    r2.input(8, 6);
    cout << "Area of r1: " << r1.area() << endl;
    cout << "Area of r2: " << r2.area() << endl;
    cout << "Perimeter of r1: " << r1.perimeter() << endl;
    cout << "Perimeter of r2: " << r2.perimeter() << endl;
    return 0;
}
```

```
C:\Users\diyag\OneDrive\Desktop\DSA-practice\labAssignments\assignment1>a
Area of r1: 90
Area of r2: 48
Perimeter of r1: 38
Perimeter of r2: 28
```