# ARRAY

Linear search

#include <iostream>

using namespace std;

int linear\_search(int array[], int key, int n)

{

    for (int i = 0; i < n; i++)

    {

        if (array[i] == key)

        {

            return i;

        }

    }

    return -1;

}

int main()

{

    int n, key;

    cin >> n;

    int array[n];

    for (int i = 0; i < n; i++)

    {

        cin >> array[i];

    }

    cout<<endl;

    cin>>key;

    cout << linear\_search(array, key, n);

cout << “time complexity o(n)”;

}

Binary search

#include <iostream>

using namespace std;

int binery\_search(int array[], int n, int key)

{

    int s = 0, e = n, mid;

    while (s <= e)

    {

        mid = (s + e) / 2;

        if (array[mid] == key)

        {

            return mid;

        }

        else if (array[mid] > key)

        {

            e = mid - 1;

        }

        else

        {

            s = mid + 1;

        }

    }

     return -1;

}

int main()

{

    int n, key;

    cin >> n;

    int array[n];

    for (int i = 0; i < n; i++)

    {

        cin >> array[i];

    }

    cout << "Enter Key" << endl;

    cin >> key;

    cout << binery\_search(array, n, key);

    cou<<"Time complexity of binary search = O(log n base 2)";

}

Sum of Sub array

#include <iostream>

using namespace std;

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++)

    {

        int cur=0;

        for(int j=i;j<n;j++)

        {

            cur+=arr[j];

            cout<<cur<<endl;

        }

    }

}

Find max and min in the array

#include <iostream>

using namespace std;

int main()

{

    int arr[5] = {5, 7, 2, 0, 9};

    int max\_1 = arr[0], min\_1 = arr[0],h=0,x,y;

    for (int i = 0; i < 5; i++)

    {

        x=max(h,arr[i]);

        y=min(h,arr[i]);

        if (arr[i] > max\_1)

        {

            max\_1 = arr[i];

            cout << max\_1;

        }

        else if (arr[i] < min\_1)

        {

            min\_1 = arr[i];

            cout << min\_1;

        }

    }

    cout<<endl;

    cout<<"using inbilt function"<<endl;

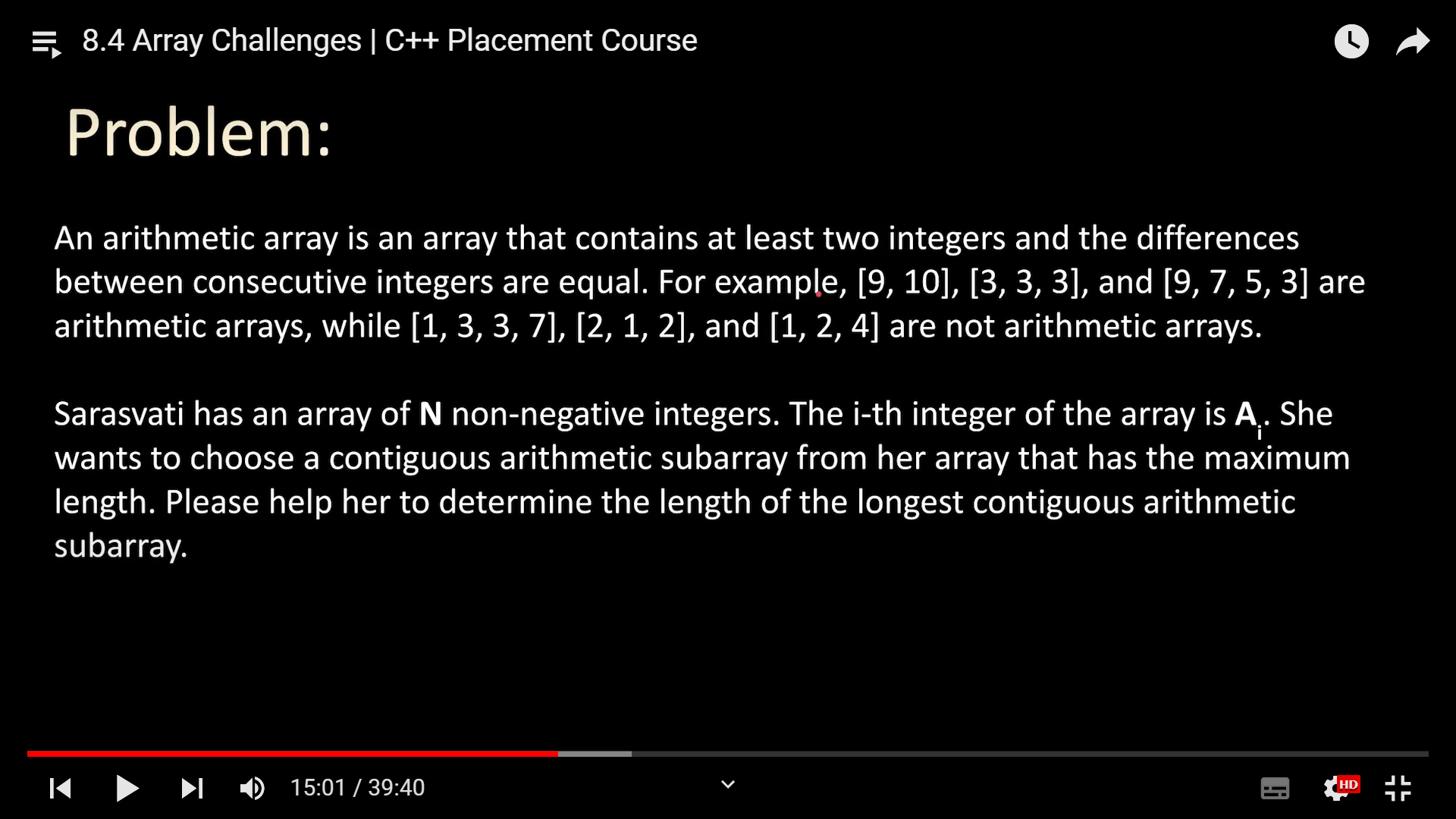
    cout<<x<<endl;

    cout<<y<< endl;

    cout << "maximum Element: " << max\_1 << endl;

    cout << "minimum Element: " << min\_1 << endl;

}



// kick start coding compitition

#include "bits/stdc++.h"

using namespace std;

int main()

{

    int n;

    cin >> n;

    int arr[n];

    for (int i = 0; i < n; i++)

    {

        cin >> arr[i];

    }

    int pd = arr[1] - arr[0];

    int j = 2;

    int current = 0, ans = 0;

    while (j < n)

    {

        if (pd == arr[j] - arr[j - 1])

        {

            current++;

        }

        else

        {

            pd = arr[j] - arr[j - 1];

            current = 2;

        }

        ans = max(ans, current);

        j++;

    }

    cout<< ans;

    return 0;

}

|  |  |
| --- | --- |
| array =[1,2,3,4,5,5,6,7]  sub array=[1,2,3,4]  NO OF SUB ARRAY Nc2+N=N\*(N+1)/2 | array =[1,2,3,4,5,5,6,7]  subsequence =[1,3,5,7]  continue without breaking order  2^n |

**Record bracker kick start**

**Problem:**

**Isyana is given the number of visitors at her local theme park on N consecutive**

**days. The number of visitors on the i-th day is Vi**

**. A day is record breaking if it**

**satisfies both of the following conditions:**

**● The number of visitors on the day is strictly larger than the number of**

**visitors on each of the previous days.**

**● Either it is the last day, or the number of visitors on the day is strictly larger**

**than the number of visitors on the following day.**

**Note that the very first day could be a record breaking day!**

**Please help Isyana find out the number of record breaking days**.

#include <iostream>

using namespace std;

int main()

{

    int n;

    cin>>n;

    int arr[n+1];

    arr[n]=-1;

    for(int i=0; i<n;i++)

    {

        cin>>arr[i];

    }

    if(n==1)

    {

        cout<<"1"<<endl;

        return 0;

    }

    int mx=-1,ans=0;

    for(int i=0;i<n;i++)

    {

        if(arr[i]>mx && arr[i]>arr[i+1])

        {

            ans++;

        }

        mx=max(mx,arr[i]);

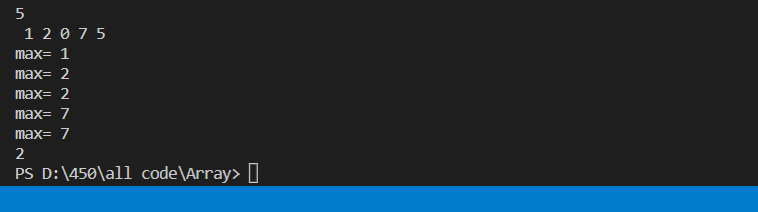
        cout<<"max= "<<mx<<endl;

    }

    cout<<ans<<endl;

    return 0;

}



2d array find element

#include <iostream>

using namespace std;

int main()

{

    int n, m;

    cin >> n >> m;

    int arr[n][m];

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < m; j++)

        {

            cin >> arr[i][j];

        }

    }

    for(int i=0; i<n;i++)

    {

        for(int j=0 ;j<m;j++)

        {

            cout<<arr[i][j]<<" ";

        }

        cout<<endl;

    }

    int x;

    bool flag= false;

    cout<<"Enter Element which you want to find";

    cin>>x;

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<m;j++)

        {

            if(arr[i][j]==x)

            {

                cout<<i<<j<<endl;

                flag=true;

            }

        }

    }

    if(flag==true)

    {

        cout<<"found element";

    }

    else{

        cout<<"not found";

    }

}

Spiral order matrix:

Char array & palindrome

#include <iostream>

using namespace std;

int main()

{

    int n;

    cin>>n;

              //char array must have at the end null char

    char arr[n+1];

    cin>>arr;

    cout<<arr<<endl;

    cout<<arr[3]<<endl;

    int flag=1;

    for(int i=0;i<n;i++)

    {

        if(arr[i] != arr[n-1-i])

        {

            flag=0;

            break;

        }

    }

    if(flag == 1)

    {

        cout<<"Yor word is palindrome";

    }

    else

    {

        cout<<"your word is not palindrome";

    }

}

Find maximum elemet in the array

#include <iostream>

using namespace std;

int main()

{

    int size;

    cin >> size;

    cin.ignore();

    char array[size + 1];

    cin.getline(array, size);

    cin.ignore();

    int curr\_length = 0, max\_length = 0;

    int max\_start = 0;

    int i = 0;

    while (1)

    {

        if (array[i] == ' ' || array[i] == '\0')

        {

            if (curr\_length > max\_length)

            {

                max\_length = curr\_length;

            }

            curr\_length = 0;

        }

        else

        {

            curr\_length++;

        }

        if (array[i] == '\0')

        {

            break;

        }

        i++;

    }

    cout << "Maximum Length : " << max\_length << endl;

    cout << "Word is : ";

    for (int i = 0; i < max\_length; i++)

    {

        cout << array[i + max\_start];

    }

    return 0;

}