# Time complexity + STL vectors

# Types of errors :

1. TLE: time limit exceeded
2. MLE: local array: 10^5 (Memory limit exceeded)
   Global array: 10^7
3. RE: runtime error

   -Out of bounds:

```
Int a[10]; //a[0...9]
cout<<a[12];
```

   -Division by 0/mod by 0:

```
Int a,b;
cin>>a>>b;
if(b!=0){
        cout<<a/b;
}else{
        cout<<”Div..”;
}
```

   -Int a[100000000];

```
Int a=1000000000;
Int b=a;
Int c = a;
Long long d = a+b+c;
cout<<d;
```

```
#include<bits/stdc++.h>
#define ll long long
Using namespace std;
Int main()
{
        ll a;
}


Int main()
{
        ios_base::sync_with_stdio(NULL);
        cin.tie(NULL);
        cout.tie(NULL);
```

Note-> never use (==) operator for comparison of floating number.

Time complexity:

```
Int n;
cin>>n;
Int sum=0;
for(int i=1;i<=n;i++){
        sum= sum+i;
}
```

Number of time i<=n is checked = n+1
Number of times  i++ is executed =n
Number of times sum=sum+i is executed=n
Total number of operations = n+1+n+n = 3*n+1
Time complexity of a code = O(n)

a1n^x+a2n^(x-1)+...
O(n^x)

Eg 1

```
for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){ // runs from 0 to n-1
                cout<<i+j<<" ";
        }
}
```

For i = 0, inner loop runs n times;

For i=1,  n times;
….
For i=n-1, n times;
Total  = n+n+n+...(n times)  = n*n = n^2

Eg 2

```
for(int i=0;i<n;i++){
        for(int j=0;j<=i;j++){
                cout<<i+j<<” “;
        }

}
```

For i=0, -> 1
For i=1, -> 2
For i=2, -> 3
….
For i=n-1, -> n
Total = 1+2+3+...n = n*(n+1)/2 = n^2/2+n/2
O(n^2)

```
cout<<10; // O(1)
```

Eg-3

```
for(int i=1;i<=n;i++){
    for(int j=1;j<=n;j+=i){
        cout<<i+j;
    }
}
```

For i=1, -> n times<br>
For i=2, -> n/2 times;<br>
For i=3, -> n/3 times<br>
…….<br>
Total  = n+n/2+n/3+n/4….<br>
          = nlogn<br>
O(nlogn)

Eg-4

```
for(int i=1;i<=n;i *=2){
    cout<<i;
}
```

i=1,2,4,8,.... Upto n<br>
Logn = number of terms<br>
O(logn)

E.g-5

```
for(int i=1;i<=n;i*=2){
    for(int j=1;j<=n;j*=2){
        cout<<i+j;
    }
}
```

O((logn)^2)

E.g-6

```
for(int i=1;i<=n;i*=3){
    cout<<i;
}
```

O(log_3(n)) approximately equal to O(log(n))

10^8 operations per second

O(N) -> linear time
N<=10^8
O(N^2)
N^2<=10^8
N<=10^4

Nlogn<=10^8
N<10^7

Space complexity:-

Int a[n];
Space complexity - > O(N)
Int a[n][n];
O(n^2);

Max limits of n….

1. O(n) -> 10^8
2. O(n^2) ->10^4
3. O(n^3) ->10^2.66
4. O(n^4) ->100
5. O(nlogn) ->10^7
6. O(2^n) ->30
7. O(n!) ->10

What is the time complexity of this code..
    sum=sum+2; - O(1)

Vector in C++ STL.

We have discussed about static Array..

Q. Write a syntax to define an array of 100 size of integer datatype.

Int Array[100]; -> Static array

vector<datatype> a; -> Dynamic array.
-> vector<string> vec;
Important functions on vector..

1. push_back() :- used to add a new element.
   Time complexity - O(1)
   ```
   vector<int> Freshers; // initially the vector is empty
   Freshers={};
   1,2,3,4,5;
   Freshers.push_back(element);
   Freshers.push_back(1);
   Freshers.push_back(2);
   Freshers.push_back(3);
   Freshers.push_back(4);
   Freshers.push_back(5);
   ```

Freshers={1,2,3,4,5}

2. size()
   // Time complexity - O(1);
   //Int n = Freshers.size() -> returns the size of vector
   → How to access the elements presents in Vector.
   We access the elements presents in vector exactly in the same way as in case of array.

0 to 2^32-1

Freshers[0]-> represents the 1st element

```
Int n = Freshers.size();
for(int i=0;i<n;i++){
        Freshers.push_back(i+1); //O(1)
}
```

Eg

```
Int n;
cin>>n;
/*Int a[n];
for(int i=0;i<n;i++){
        cin>>a[i];
}
*/
vector<int> v;
for(int i=0;i<n;i++){
        Int temp;
        cin>>temp;
        v.push_back(temp);
```

```
    }
```

Eg

```
Int a[n],x=50;
```

Store elements present in the array a which are >x

```
vector<int> ans;
for(int i=0;i<n;i++){
    if(a[i]>x){
        ans.push_back(a[i]);
    }
}
```

Syntax to define a vector of size 100 at the time of declaration :-

```
vector<int> a;

 → vector<int> Vec(5);//0 to 4
    {0,0,0,0,0}
    Vec[2]=1;
    Vec.push_back(2);
    {0,0,1,0,0,2}
```

```
Int arr[] = {1,2,3,4,5};
```

# STL Pairs, sort() and structure in C++

## pop_back() in vector

- Removes the last element of a vector

```
vector<int> vec={1,2,3,4,5}; // (size=5)
vec.pop_back();
```

After pop_back(),

vec becomes {1,2,3,4}; (size=4)

**Time Complexity = O(1)**

## sort()

**Time Complexity = O(n log n )**

vec:={5,1,7,3,4,9}

**syntax-> sort(starting_iterator, ending_iterator)**

**starting_iterator->** vec.begin();

**ending_iterator->** vec.end();

**Example**

```
sort(vec.begin(),vec.end());
```

## swap()

**Syntax: swap(a, b);**

**swap is used to interchange the values of any 2 variables**

**Example for sort() :**

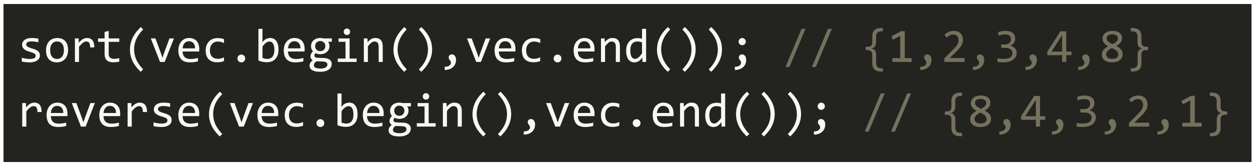
Let n = size of vector<br>
vec:= {1,2,3,4,5,6};<br>
Staring_itr -> vec.begin();<br>
Second_itr -> vec.begin()+1;<br>
Third_itr -> vec.begin()+2;<br>
..<br>
Last_itr -> vec.begin()+n-1;<br>
vec.begin()+n == vec.end();<br>
sort(vec.begin(), vec.begin()+n);<br>
sort(vec.begin(), vec.end());<br>
L---r  sort(l,r+1);

## reverse()

**Time complexity: O(n)**

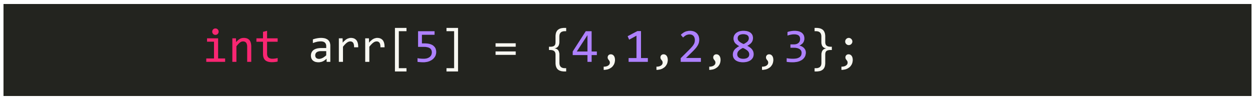
vec-> {4,1,2,8,3};

**How will you sort in decreasing order ?**



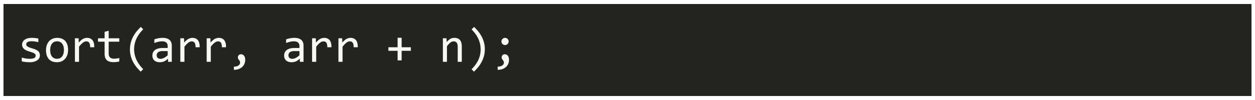
```
sort(vec.begin(),vec.end()); // {1,2,3,4,8}
reverse(vec.begin(),vec.end()); // {8,4,3,2,1}
```

**How do we sort in case of array?**



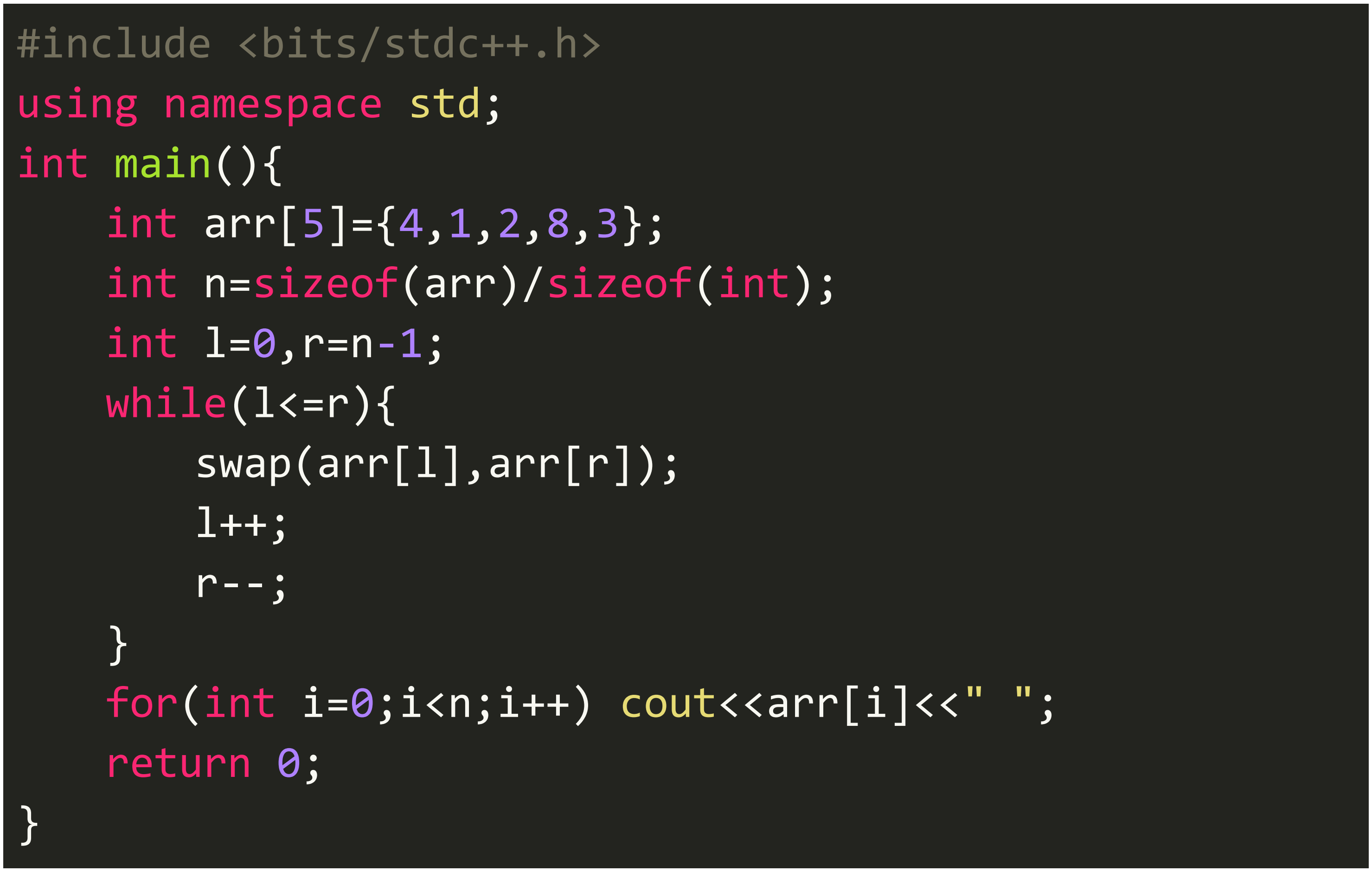
```
        int arr[5] = {4,1,2,8,3};
```

n = size of array (Here, it is 5)
Staring_itr -> arr;
Second_itr -> arr+1;
Third_itr -> arr+2;
..
Last_itr -> arr+n-1;
Ending_itr -> last_itr+1 == arr+n-1+1 == arr+n;



```
sort(arr, arr + n);
```

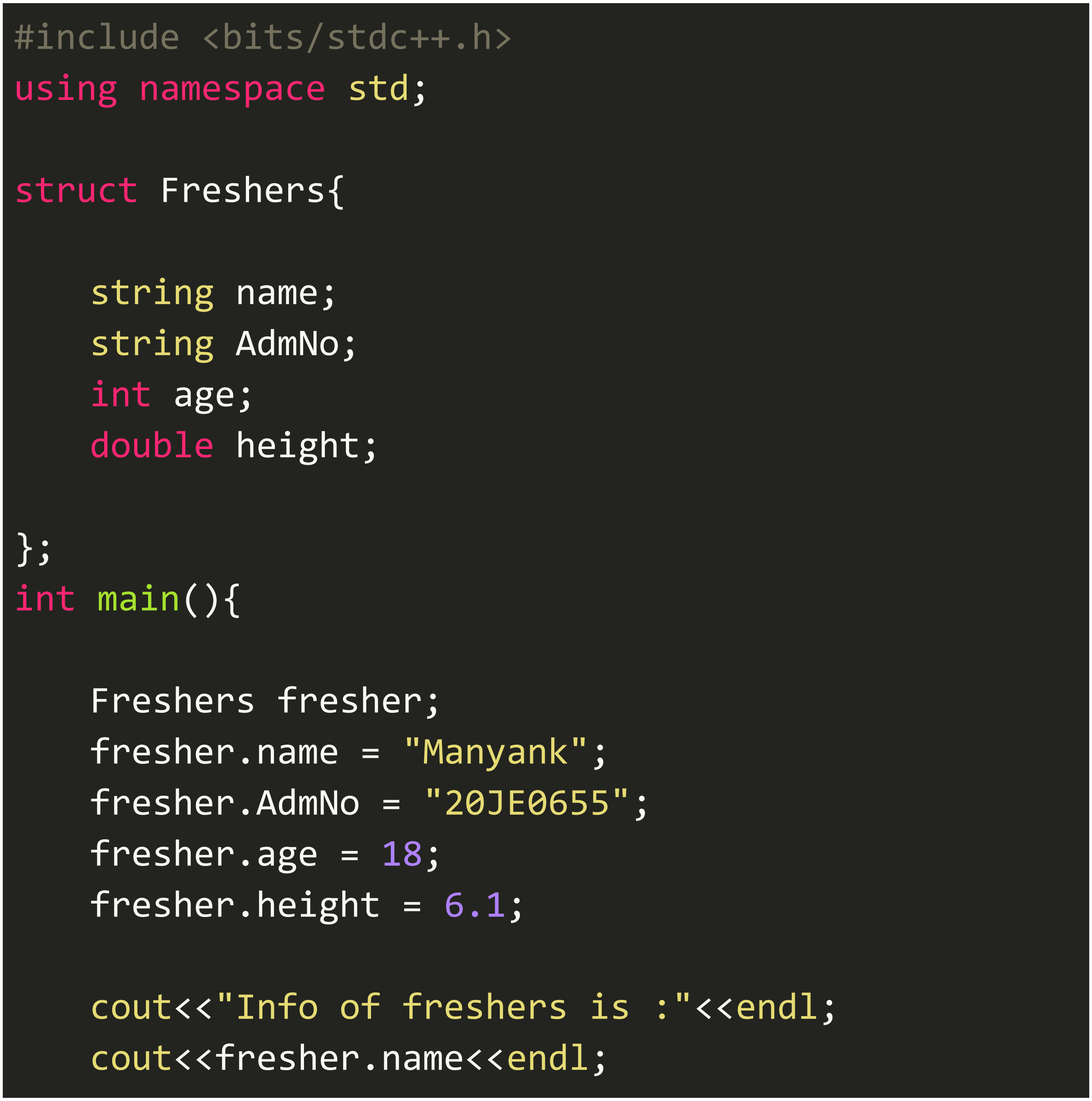
**Code-1 (To reverse an array without using reverse() )**



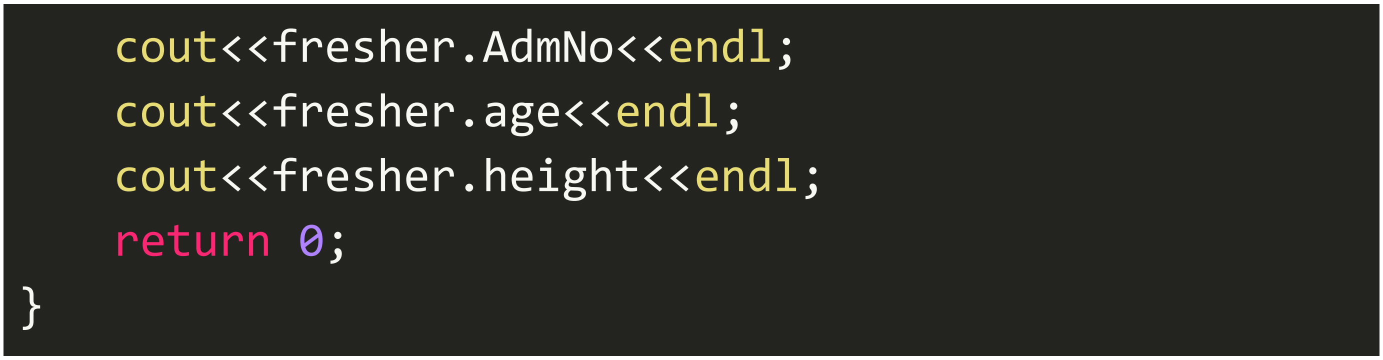
```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int arr[5]={4,1,2,8,3};
    int n=sizeof(arr)/sizeof(int);
    int l=0,r=n-1;
    while(l<=r){
        swap(arr[l],arr[r]);
        l++;
        r--;
    }
    for(int i=0;i<n;i++) cout<<arr[i]<<" ";
    return 0;
}
```

# Struct

**Code-1**

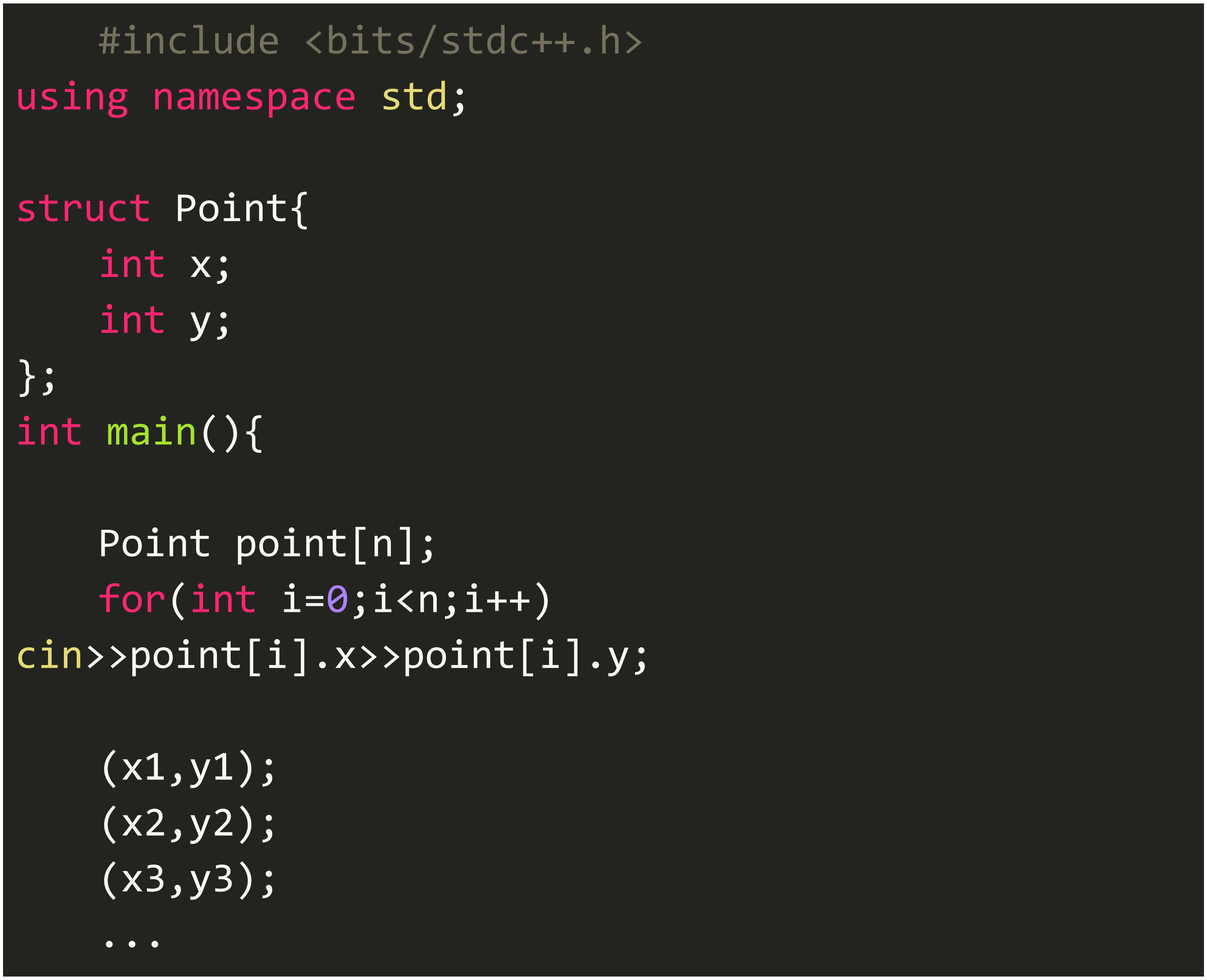


```
#include <bits/stdc++.h>
using namespace std;

struct Freshers{

    string name;
    string AdmNo;
    int age;
    double height;

};
int main(){

    Freshers fresher;
    fresher.name = "Manyank";
    fresher.AdmNo = "20JE0655";
    fresher.age = 18;
    fresher.height = 6.1;

    cout<<"Info of freshers is :"<<endl;
    cout<<fresher.name<<endl;
```

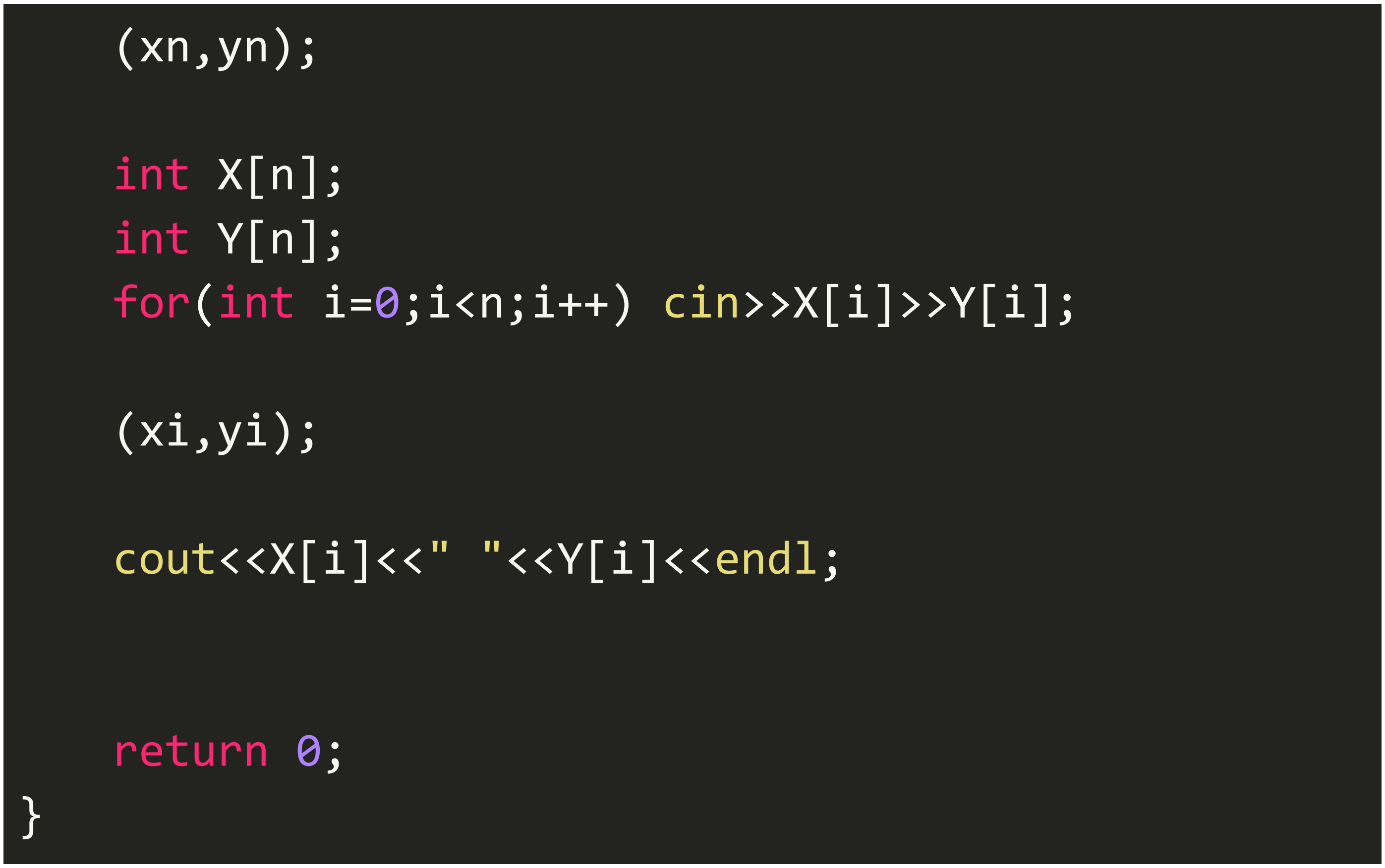


```
    cout<<fresher.AdmNo<<endl;
    cout<<fresher.age<<endl;
    cout<<fresher.height<<endl;
    return 0;
}
```

**Code-2**

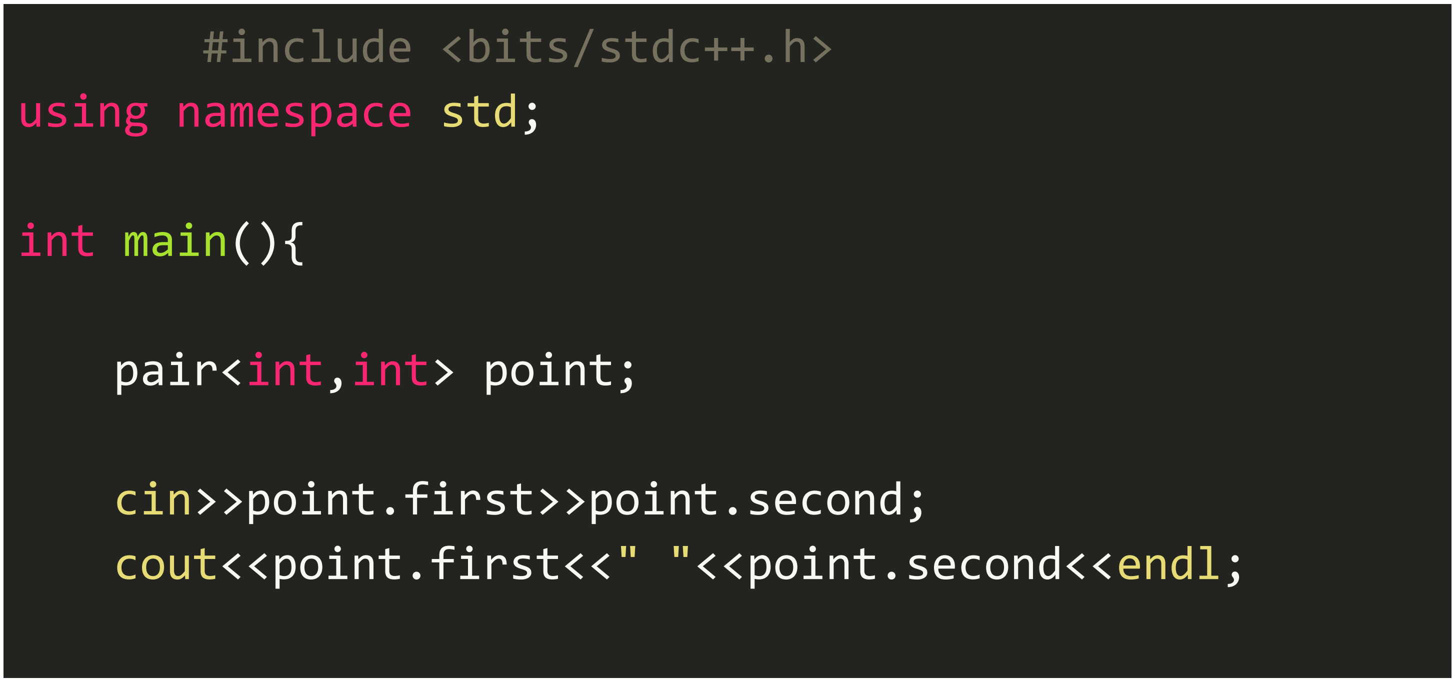


```
    #include <bits/stdc++.h>
using namespace std;

struct Point{
    int x;
    int y;
};
int main(){

    Point point[n];
    for(int i=0;i<n;i++)
cin>>point[i].x>>point[i].y;

    (x1,y1);
    (x2,y2);
    (x3,y3);
    ...
```



```cpp
    (xn,yn);

    int X[n];
    int Y[n];
    for(int i=0;i<n;i++) cin>>X[i]>>Y[i];

    (xi,yi);

    cout<<X[i]<<" "<<Y[i]<<endl;



    return 0;
}
```

### Pair in C++ STL



```cpp
       #include <bits/stdc++.h>
using namespace std;

int main(){

    pair<int,int> point;

    cin>>point.first>>point.second;
    cout<<point.first<<" "<<point.second<<endl;

    pair<string,double> Fresher;

    cin>>Fresher.first;
    cin>>Fresher.second;

    cout<<"The name of the student is: 
"<<Fresher.first<<endl;
    cout<<"The height of the student is: 
"<<Fresher.second<<endl;

    return 0;
}
```

**Point in 3D->**

```
#include <bits/stdc++.h>
using namespace std;
int main(){

    pair<int,pair<int,int>> point3D;
    // x-> point3D.first;
    // y-> point3D.second.first;
    // z-> point3D.second.second;

    return 0;
}
```

## Hackerrank Problem: Equalize the array

**Link: [https://www.hackerrank.com/challenges/equality-in-a-array/problem](https://www.hackerrank.com/challenges/equality-in-a-array/problem)**
**(First, try it by yourself, then only look at the solution below)**

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int a[n]; // 3 1 1 2 2 2 3 3 3
    for(int i=0;i<n;i++){
        cin>>a[i];
    }
    sort(a,a+n); // 1 1 2 2 2 3 3 3 3
    int ans = 10000;
    for(int i=0;i<n;i++){
        int freq=0;
        int com = a[i];
        while(i<n&&a[i]==com){
            freq++;
            i++;
        }
        i--;
        ans = min(ans,n-freq);
    }
    cout<<ans;
}
```

## Try this problem, it would be discussed in next class:

https://atcoder.jp/contests/abc187/tasks/abc187_d