# STL set and map in C++

Some important C++ STL(Standard Template Library)

# **Containers**

## Vector

**Try to solve this problem:**

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

**Sol:**

| #include <bits/stdc++.h> #define int long long using namespace std; int32\_t main() {  int n;  cin>>n;  pair<int,int> a[n];  for(int i=0;i<n;i++){  cin>>a[i].first>>a[i].second;  }  int x=0,y=0; // x-> aoki's votes and y-> takahashi's votes  for(int i=0;i<n;i++){  x+=a[i].first;  }  for(int i=0;i<n;i++){  int temp = a[i].first;  a[i].first = 2\*a[i].first+a[i].second;  a[i].second = temp;//{2a+b,a}  }  sort(a,a+n);  int ans=0;  int i=n-1;  while(x>=y){  ans++;  y+=a[i].first-a[i].second;  x-=a[i].second;  i--;  }  cout<<ans;  /\*  a[i].first=a,a[i].second=b;  2\*a+b  \*/ } |
| --- |

**Note:** By default, the sort() function sorts the vector elements on basis of first element of pairs and compare the second element, only when first element of two pairs are same.

**Why we used #define int long long ?**

* To prevent overflows
* Since, it will replace int by long long everywhere in the program

**Why we used int32\_t main() instead of int main() ?**

* Because int is replaced by long long in our program, it will become long long main(), which gives error on compiling. So, we need to write int32\_t main(). int32\_t is same as int .

## Set

set is a special kind of STL container which stores **unique** elements in **sorted order**.

**Syntax of declaring a set**

set<data\_type> set\_name;

**Examples:**

| set<int> st;  set<float> st2;  set<double> st3; |
| --- |

| set<int> st; |
| --- |

**insert() function in set**

**Time complexity:** O( log n )

| st.insert(4); // {4} st.insert(3); // {3,4} st.insert(1); // {1,3,4} st.insert(3); // {1,3,4} |
| --- |

**size() function in set**

**Time complexity:** O(1)

Example:

| int n=st.size(); |
| --- |

**erase() function in set**

**Time complexity:** O(log n)

Example:

| st.erase(3); |
| --- |

1. If the number is present then it will remove it from the set.
2. If not then nothing happens.

**empty() function in set**

To know whether the set in empty or not

**Time complexity:**  O(1)

| bool isEmpty = st.empty(); // true/false |
| --- |

**Q. You will be given n numbers and after inserting each number you have print all the inserted numbers till now in sorted form;**

1 -> 1

3 -> 1,3

2 -> 1,2,3

**Method 1 - Use set**

| set<int> st;  for(int i=0;i<n;i++){  int x;  cin>>x;  st.insert(x); // O(logn);  print(st); // O(n);  } |
| --- |

**Total Time complexity:** O(n^2)

**Method 2 - Use array**

| vector<int> vec; for(int i=0;i<n;i++){  int x;  cin>>x;  vec.push\_back(x);  sort(vec.begin(),vec.end()); // O(nlogn)  int sz = vec.size();  for(int j=0;j<sz;j++) cout<<vec[j]<<" "; // O(n);  cout<<endl; } |
| --- |

**Time complexity**: O(n2 log(n))

[ Slower than method 1 ]

**Print elements in set**

| for(auto it=st.begin();it!=st.end();it++){  cout<<\*it<<" ";  } |
| --- |

**find() function in set**

set -> {1,3,5,6,7,8};

| auto it = st.find(6);   cout<< \*it << endl; // 6 |
| --- |

**If 6 is not present then it = st.end()**; NULL;

// \*it -> run time error.

**Q. Check whether a number x is present or not in the set.**

**Method - 1: ( Using .find() )**

| auto it = st.find(x);  if(it==st.end()) cout<<"NOT PRESENT";  else cout<<"PRESENT"; |
| --- |

**Method - 2: (Using .count() )**

count() returns the number of times an element occurs in the set

| int cnt = st.count(x);  if(cnt) {  cout<<"PRESENT"; } else {  cout<<"NOT PRESENT";  } |
| --- |

## Map

Map is a special kind of STL container which stores elements as **key-value pair**. No two mapped values can have same key. **All the keys are sorted in ascending order**.

**All the keys are unique.**

**Example:** All the Freshers will be having a unique admission number.

“20JE0666” -> Sakshi ;

“20JE0648” -> Saksham ;

“20JE0654” -> Shivali ;

**Key -> value;**

**Syntax of declaration ->**

| map<firstDatatype, secondDatatype> mp; |
| --- |

**Example:**

| map<string, string> mp;  mp["20JE0666"]="Sakshi";  mp["20JE0648"]="Saksham";  mp["20JE0654"]="Shivali"; |
| --- |

**Note->** Two different keys may have same values.

Like admission number “20JE0888” and “20JE0898” both can have name as “Yash”

map->[{Sakhi: 20JE0666}, {Saksham: 20JE0648},......};

**Another example:**

| map<int,char> mp1; mp1[1]='A'; mp1[2]='B'; mp1[3]='C';  map<char,int> mp2; mp2['A']=1; mp2['B']=2;  mp2['C']=3; |
| --- |

1. **size() function in map:**

| int n = mp.size(); |
| --- |

**Time complexity:** O(1)

1. **erase() function in map:**

| mp.erase(key); |
| --- |

**Time complexity:** O(logn);

1. **count() function in map:**

| mp.count(key); |
| --- |

**Time complexity:**O(logn);

-> mp[key] = value;

If not present that key -> random value.

If case of integer -> default 0;

1. **Printing all elements of a map:**

The elements of map are a **pair** of key and value. So, you can use .first to access key and .second to access value and iterate through all values, similar to that in a set.

**Since, iterators are pointers, use arrow operator to access their members.**

The code to print everything inside a map named mp would be like:

| for (it = mp.begin(); it != mp.end(); it++) {  cout<<( it->first )<< ' ' <<( it->second )<< '\n';  } |
| --- |

**For more functions of map, refer:**

<https://www.cplusplus.com/reference/map/map/>

## Question1->

**Print this pattern using for loop:**

1

1,2,

1,2,3,

1,2,3,4

1,2,3,4,5

**Approach:** (Use a variable i for row)

i=1 1

i=2 1,2,

i=3 1,2,3,

i=4 1,2,3,4

i=5 1,2,3,4,5

| for(int i=1;i<=n;i++){  for(int j=1;j<=i;j++) cout<<j<<" ";  cout<<endl;  } |
| --- |

## Question2

<https://www.hackerrank.com/challenges/marcs-cakewalk/problem>

**Approach:-**

1, 2, 4, 8, 16..... [constant value]

A1 A2 A3 A4 A5..... [any particular arrangement]

minimise this expression( A1+2\*A2+4\*A3+8\*A4....)

[1,2,3] ascending order ----> 17 (maximum)

[2,1,3] any random order ----> 16 (in between)

[3,2,1] descending order ----> 11 (minimum)

**int** n=calorie.size();

**long** ans=0;

sort(calorie.begin(),calorie.end());

reverse(calorie.begin(),calorie.end());

**for**(**int** i=0;i<n;i++){

**long** currValue = (**long**)pow(2,i)\*calorie[i];

ans+=currValue;

}

**return** ans;

**Note:**

When I was using **“int currValue”** then there was a problem of **overflow** so always look at the **worst case** value and here when I changed it to **“long currValue”** then **all the test cases got passed**.

## ASCII Vaue using “typecasting”

When you convert a character of string into int, it will get converted to its ASCII code.

**Each character has a unique ASCII code.**

Like '0' has ASCII code 48

'1' => 49

'2' => 50

...

...

'9' => 57

'a' => 97

'b' => 98

...

...

'z' => 122

'A' => 65

'B' => 66

...

...

'Z' => 90

| #include <bits/stdc++.h>  using namespace std;   int main(){  char c='a';  cout<<c<<endl;   int value = (int)c;  cout<<value<<endl;  return 0;  } |
| --- |