

Set

- In C++, sets are associative container which stores unique elements in some sorted order.
- By default, it is sorted ascending order of the keys

Creating a set

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

int main() {

    // Creating a set of integers
    set<int> s = {3, 5, 2, 1};

    for (auto x : s)
        cout << x << " ";
    return 0;
}
```

Output

1 2 3 5

Inserting elements

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

```
int main() {
    set<int> s = {1, 4, 2};

    // Insert elements into set
    s.insert(5);
    s.emplace(3);
    s.insert(5);

    for (auto x: s) cout << x << " ";
    return 0;
}
```

Output
1 2 3 4 5

Accessing elements

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

int main() {
    set<int> s = {1, 4, 2, 3, 5};

    // Accessing first element
    auto it1 = s.begin();

    // Accessing third element
    auto it2 = next(it1, 2);

    cout << *it1 << " " << *it2;
    return 0;
}
```

Output

1 3

Finding elements

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

int main() {
    set<int> s = {1, 4, 2, 3, 5};

    // Finding 3
    auto it = s.find(3);

    if (it != s.end()) cout << *it;
    else cout << "Element not Found!";
    return 0;
}
```

Output

3

Traversing elements

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

int main() {
    set<int> s = {5, 1, 4, 3, 2};

    // Traversing using traditional for loop
    for(auto it = s.begin(); it != s.end(); it++)
        cout << *it << " ";
    return 0;
}
```

Output

1 2 3 4 5

Deleting elements

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

int main() {
    set<int> s = {1, 4, 2, 3, 5};

    // Deleting elements by value
    s.erase(5);

    // Deleting first element by iterator
    s.erase(s.begin());

    for (auto x: s) cout << x << " ";
    return 0;
}
```

Output

2 3 4

#if an element in a given set is even, delete it

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      set<int> s = {1, 2, 3, 4, 5, 6, 7, 8};
6
7      for (auto it = s.begin(); it != s.end(); ) {
8          if (*it % 2 == 0)
9              it = s.erase(it);    // erase returns next iterator
10         else
11             it++;
12     }
13
14     for (int x : s)
15         cout << x << " ";
16
17     return 0;
18 }
```


Updating elements

- We cannot change the value of elements once they are stored in the set.