









Projeto Olímpico de Programação

[ C++ ] STL: Container Set

- Coleção ordenada de "key" (sem duplicidade);
- Permite acesso sequencial aos elementos armazenados;
- Complexidade das operações básicas:

Operação	Complexidade
Inserção e remoção no início, meio e final	O(1)
Busca	O(n)

- Iteradores são válidos após alteração na lista;
- Inserção no código:

```
#include <set>
```

Declaração:

```
set<int> numeros;
```

#### insert

```
#include <iostream>
#include <set>
#include <list>
using namespace std;
int main(){
    list<int> lista;
    set<int> dados;
                                                           10
    set<int>::iterator it;
                                                           20
                                                           30
    lista.push back(10);
    lista.push back(10);
    lista.push back(20);
    lista.push back(20);
    lista.push back(30);
    lista.push back(30);
    lista.push back(10);
    lista.push back(10);
    dados.insert(lista.begin(), lista.end());
    dados.insert(dados.begin(), 10);
    dados.insert(10);
    for (it = dados.begin(); it != dados.end(); ++it){
        cout << *it << endl;</pre>
    return 0;
```

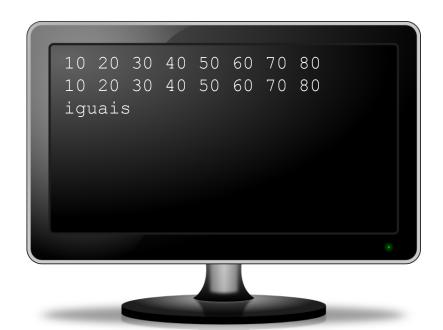
#### clear e erase

```
#include <iostream>
#include <set>
#include <list>
using namespace std;
void print_set(set<int> s);
int main(){
    set<int> dados;
    for (int i = 1; i \le 8; ++i) {
        dados.insert(i * 10);
    print set(dados);
    dados.erase(60);
    print set(dados);
    dados.erase(dados.begin(), ++++dados.begin());
    print_set(dados);
    dados.erase(dados.begin());
    print set(dados);
    return 0;
void print set(set<int> s){
    set<int>::iterator it;
    for (it = s.begin(); it != s.end(); ++it){
        cout << *it << " ";
    cout << endl;
```



#### **Operadores Relacionais**

```
#include <iostream>
#include <set>
#include <list>
using namespace std;
void print_set(set<int> s);
int main() {
    set<int> dados, copia;
    for (int i = 1; i \le 8; ++i) {
        dados.insert(i * 10);
        copia.insert(i * 10);
    print set(dados);
    print set(copia);
    if (dados == copia)
        cout << "iguais\n";</pre>
    else
        cout << "diferentes\n";
    return 0;
void print set(set<int> s){
    set<int>::iterator it:
    for (it = s.begin(); it != s.end(); ++it){
        cout << *it << " ";
    cout << endl;
```



#### find

```
#include <iostream>
#include <set>
#include <list>
using namespace std;
void print set(set<int> s);
int main(){
    set<int> dados;
    set<int>::iterator it;
    for (int i = 1; i <= 8; ++i) {
        dados.insert(i * 10);
    for (it = dados.begin(); it != dados.end(); ++it){
        cout << *it << " ";
    cout << endl;
    it = dados.find(60);
    if (it != dados.end()){
        cout << "Encontrei: " << *it << endl;</pre>
    }else{
        cout << "Nao encontrei" << endl;</pre>
    return 0:
```



#### Desafio!

Escrever um programa, em C++, seis valores ([1,60]) aleatórios e distintos.

Exibir os valores lidos.

Desafio! Resposta

