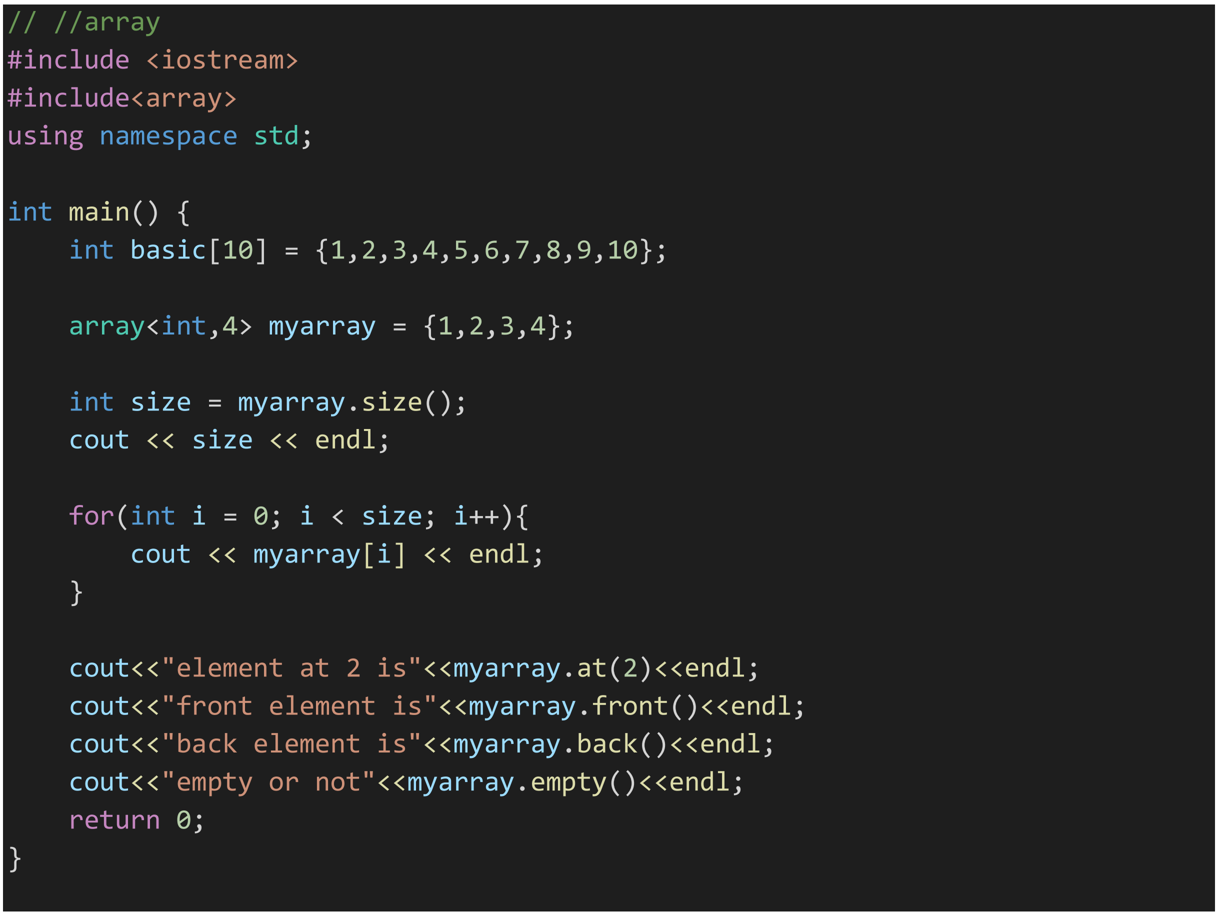
STL

1.Array

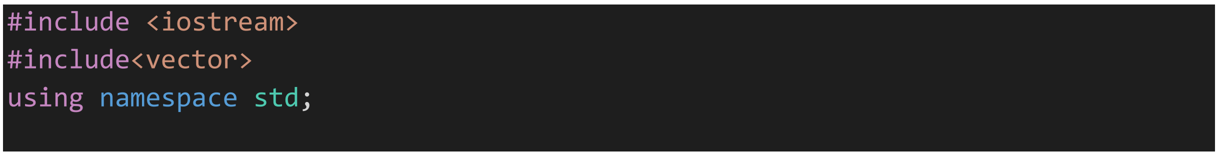


```cpp
// //array
#include <iostream>
#include<array>
using namespace std;

int main() {
    int basic[10] = {1,2,3,4,5,6,7,8,9,10};

    array<int,4> myarray = {1,2,3,4};

    int size = myarray.size();
    cout << size << endl;

    for(int i = 0; i < size; i++){
        cout << myarray[i] << endl;
    }

    cout<<"element at 2 is"<<myarray.at(2)<<endl;
    cout<<"front element is"<<myarray.front()<<endl;
    cout<<"back element is"<<myarray.back()<<endl;
    cout<<"empty or not"<<myarray.empty()<<endl;
    return 0;
}
```

OutPut:

```
4
1
2
3
4
element at 2 is3
front element is1
back element is4
empty or not0
```

2. Vector



```cpp
#include <iostream>
#include<vector>
using namespace std;
```

```
int main() {
    vector<int> v;
    vector<int> v1(5,10);
    cout<<"size of vector is"<<v.capacity()<<endl;
    cout<<"size of vector is"<<v.size()<<endl;
    for(int i: v1){
        cout<<i<<endl;
    }
    vector<int> v2(v1);
    for(int i: v2){
        cout<<i<<endl;
    }
    //front and back same as array
    return 0;
}
```

OutPut:

```
size of vector is0
size of vector is0
10
10
10
10
10
10
10
10
10
10
```

3.DeQue

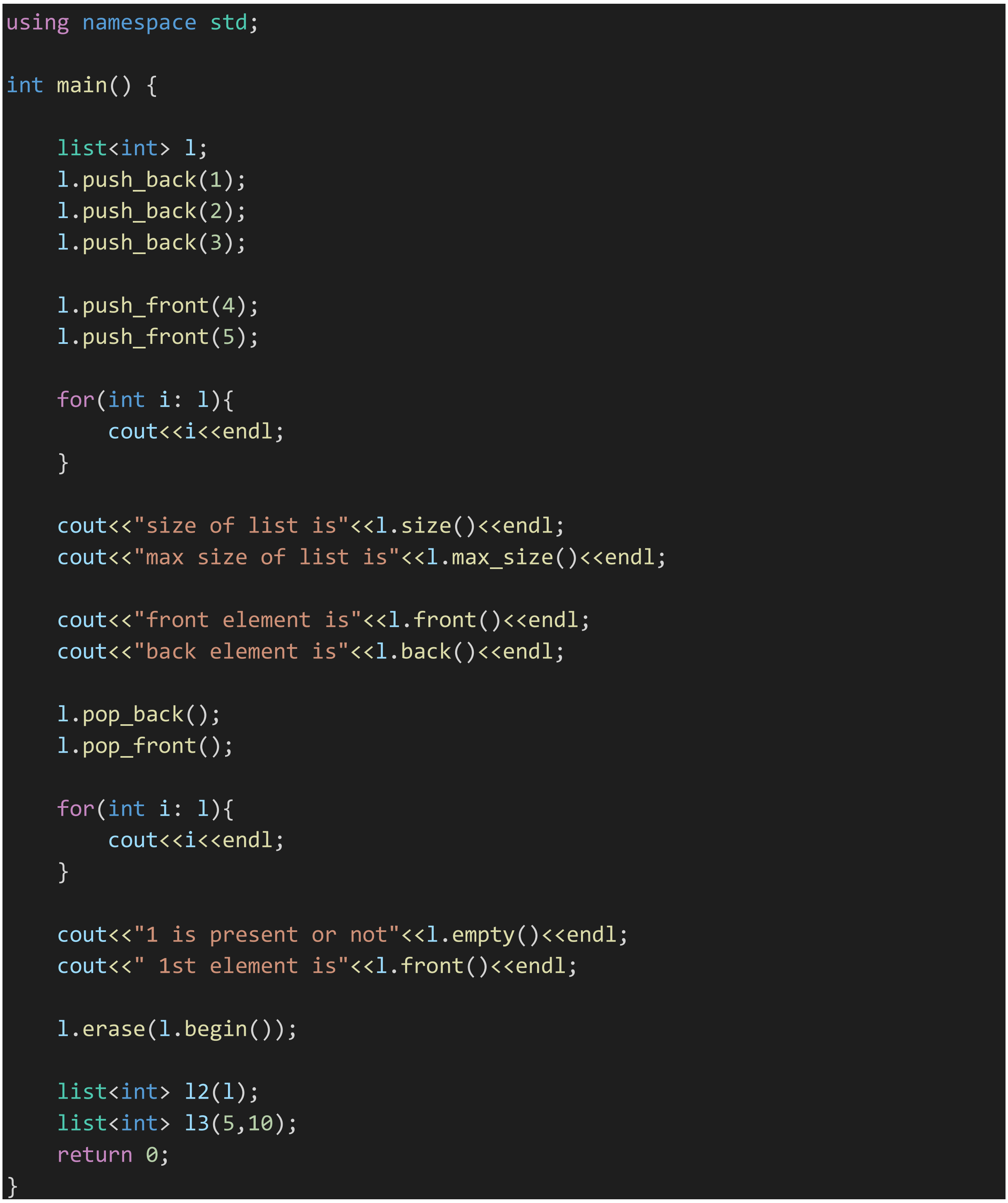
```
// Deque
#include <iostream>
#include<deque>
using namespace std;

int main() {

    deque<int> d;
    d.push_back(1);
    d.push_back(2);
    d.push_back(3);

    d.push_front(4);
    d.push_front(5);

    for(int i: d){
        cout<<i<<endl;
    }

    cout<<"size of deque is"<<d.size()<<endl;
    cout<<"max size of deque is"<<d.max_size()<<endl;

    cout<<"front element is"<<d.front()<<endl;
    cout<<"back element is"<<d.back()<<endl;

    d.pop_back();
    d.pop_front();

    for(int i: d){
        cout<<i<<endl;
    }

    cout<<"1 is present or not"<<d.empty()<<endl;
    cout<<" 1st element is"<<d.at(0)<<endl;

    d.erase(d.begin(),d.begin()+2);
    return 0;
}
```

OutPut:

```
5
4
1
2
3
size of deque is5
max size of deque is1073741823
front element is5
back element is3
4
1
2
1 is present or not0
 1st element is4
```

4.List

```
//  list
#include <iostream>
#include<list>
```



```
using namespace std;

int main() {

    list<int> l;
    l.push_back(1);
    l.push_back(2);
    l.push_back(3);

    l.push_front(4);
    l.push_front(5);

    for(int i: l){
        cout<<i<<endl;
    }

    cout<<"size of list is"<<l.size()<<endl;
    cout<<"max size of list is"<<l.max_size()<<endl;

    cout<<"front element is"<<l.front()<<endl;
    cout<<"back element is"<<l.back()<<endl;

    l.pop_back();
    l.pop_front();

    for(int i: l){
        cout<<i<<endl;
    }

    cout<<"1 is present or not"<<l.empty()<<endl;
    cout<<" 1st element is"<<l.front()<<endl;

    l.erase(l.begin());

    list<int> l2(l);
    list<int> l3(5,10);
    return 0;
}
```

OutPut:

```
5
4
1
2
3
size of list is5
max size of list is357913941
```

```
front element is5
back element is3
4
1
2
1 is present or not0
 1st element is4
```

5.Stack

```cpp
// stack
#include <iostream>
#include<stack>
using namespace std;

int main() {

    stack<string> s;

    s.push("apple");
    s.push("mango");
    s.push("banana");

    cout<<"size of stack is"<<s.size()<<endl;
    cout<<"top element is"<<s.top()<<endl;

    s.pop();
    cout<<"size of stack is"<<s.size()<<endl;
    cout<<"top element is"<<s.top()<<endl;

    cout<<"empty or not"<<s.empty()<<endl;

    return 0;
}
```

OutPut:

```
size of stack is3
top element isbanana
size of stack is2
top element ismango
empty or not0
```

6.Queue

```cpp
// queue
```

```cpp
#include <iostream>
#include<queue>
using namespace std;

int main() {

    queue<string> q;

    q.push("apple");
    q.push("mango");
    q.push("banana");

    cout<<"size of queue is"<<q.size()<<endl;
    cout<<"front element is"<<q.front()<<endl;
    cout<<"back element is"<<q.back()<<endl;

    q.pop();
    cout<<"size of queue is"<<q.size()<<endl;
    cout<<"front element is"<<q.front()<<endl;

    cout<<"empty or not"<<q.empty()<<endl;


    return 0;
}
```

OutPut

```
size of queue is3
front element isapple
back element isbanana
size of queue is2
front element ismango
empty or not0
```

7. Priority Queue

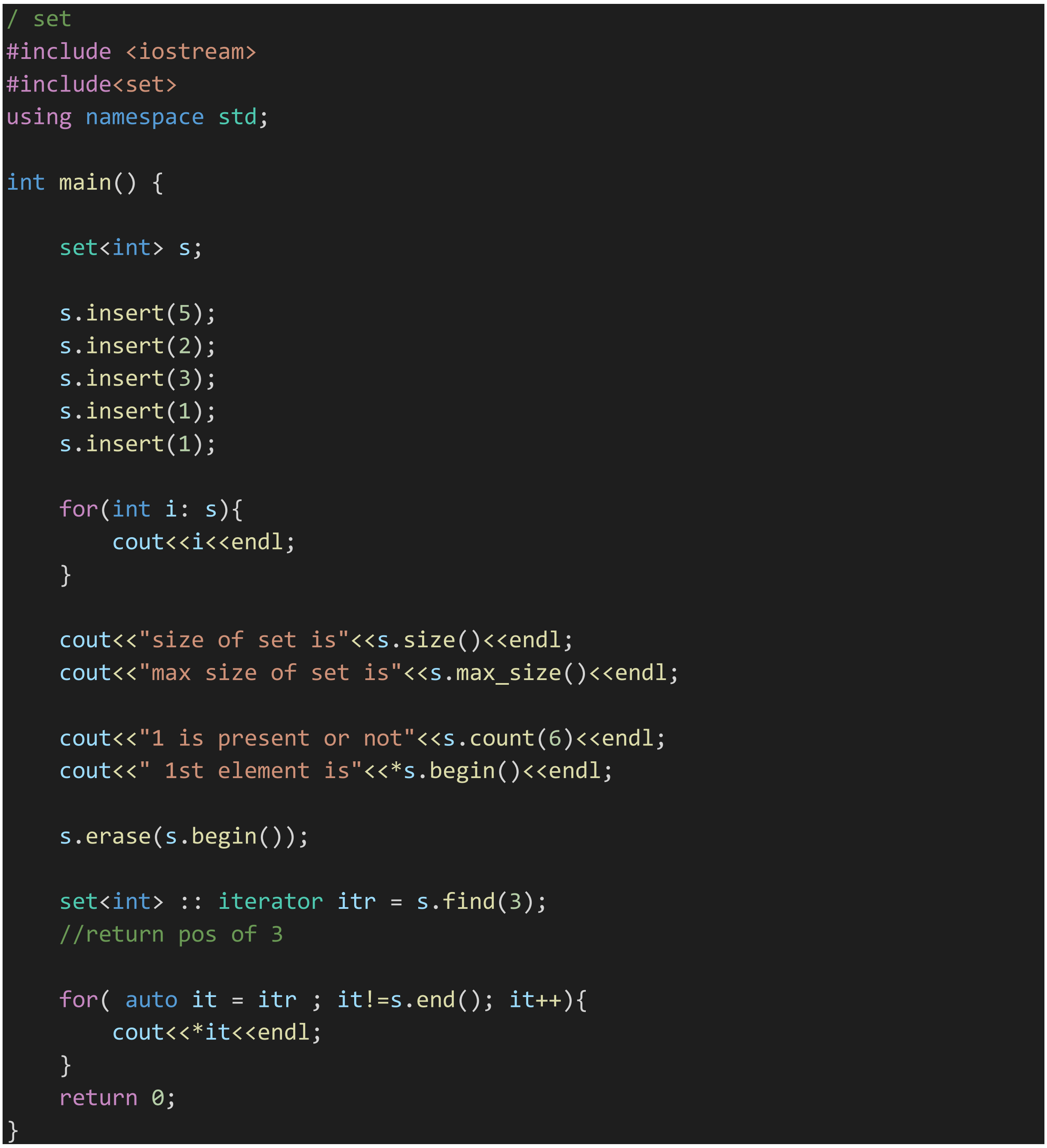
```cpp
// priority queue
#include <iostream>
#include<queue>
using namespace std;

int main() {
    //max heap
    priority_queue<int> pq;

    pq.push(1);
    pq.push(2);
    pq.push(3);
    pq.push(4);
    pq.push(5);

    cout<<"size of priority queue is"<<pq.size()<<endl;
    cout<<"top element is"<<pq.top()<<endl;

    int len1  = pq.size();
    for(int i= 0; i<len1; i++){
        cout<<pq.top()<<endl;
        pq.pop();
    }

    cout<<"empty or not"<<pq.empty()<<endl;

    //min heap
    priority_queue<int, vector<int>, greater<int>> pq2;

    pq2.push(1);
    pq2.push(2);
    pq2.push(3);
    pq2.push(4);
    pq2.push(5);


    cout<<"size of priority queue2 is"<<pq2.size()<<endl;
    cout<<"top element is"<<pq2.top()<<endl;
    int len  = pq2.size();
    for(int i= 0; i<len; i++){
        cout<<pq2.top()<<endl;
        pq2.pop();
    }



    cout<<"empty or not"<<pq2.empty()<<endl;


    return 0;
}
```

OutPut:

```
size of priority queue is5
top element is5
5
4
3
2
```

```
1
empty or not1
size of priority queue2 is5
top element is1
1
2
3
4
5
empty or not1
```

8.Set

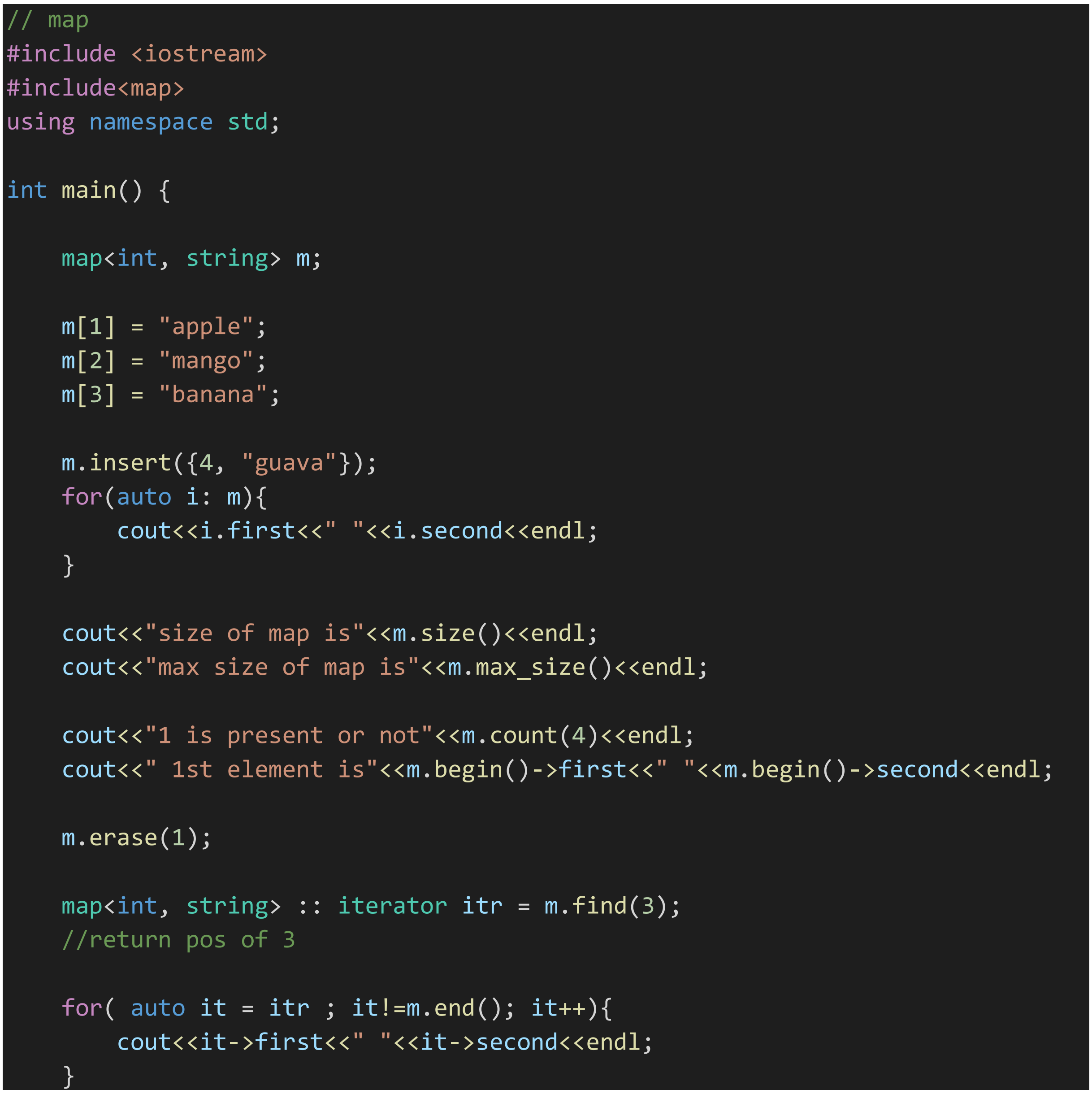


```
/ set
#include <iostream>
#include<set>
using namespace std;

int main() {

    set<int> s;

    s.insert(5);
    s.insert(2);
    s.insert(3);
    s.insert(1);
    s.insert(1);

    for(int i: s){
        cout<<i<<endl;
    }

    cout<<"size of set is"<<s.size()<<endl;
    cout<<"max size of set is"<<s.max_size()<<endl;

    cout<<"1 is present or not"<<s.count(6)<<endl;
    cout<<" 1st element is"<<*s.begin()<<endl;

    s.erase(s.begin());

    set<int> :: iterator itr = s.find(3);
    //return pos of 3

    for( auto it = itr ; it!=s.end(); it++){
        cout<<*it<<endl;
    }
    return 0;
}
```

OutPut:

```
1
2
3
5
size of set is4
max size of set is214748364
1 is present or not0
 1st element is1
3
5
```

8.Map



```cpp
// map
#include <iostream>
#include<map>
using namespace std;

int main() {

    map<int, string> m;

    m[1] = "apple";
    m[2] = "mango";
    m[3] = "banana";

    m.insert({4, "guava"});
    for(auto i: m){
        cout<<i.first<<" "<<i.second<<endl;
    }

    cout<<"size of map is"<<m.size()<<endl;
    cout<<"max size of map is"<<m.max_size()<<endl;

    cout<<"1 is present or not"<<m.count(4)<<endl;
    cout<<" 1st element is"<<m.begin()->first<<" "<<m.begin()->second<<endl;

    m.erase(1);

    map<int, string> :: iterator itr = m.find(3);
    //return pos of 3

    for( auto it = itr ; it!=m.end(); it++){
        cout<<it->first<<" "<<it->second<<endl;
    }
    return 0;
}
```

OutPut:

```
1 apple
2 mango
3 banana
4 guava
size of map is4
max size of map is97612893
1 is present or not1
 1st element is1 apple
3 banana
4 guava
```