#include <bits/stdc++.h>

using namespace std;

void print(vector<int>data){

cout<<"element number ="<<data.size()<<endl;

for(int x:data){

cout<<x<<" ";

cout<<endl <<endl;

}

}

int main()

{

// cout << find(arr,arr+5,4)-arr << endl;

// cout << find(arr,arr+5,2)-arr << endl;

// cout << find(arr,arr+5,9)-arr << endl;

// cout << binary\_search(arr,arr+5,2) << endl;

// cout << binary\_search(arr,arr+5,0) << endl;

vector <int>arr(5);

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL);

/// create

///vector <type> name (elements);

// vector<int> data (5); /// int data[5];

// print(data);

// vector<int> data;

// print(data);

// vector<int> data(5, 2);

// print(data);

// vector<int> data({1, 2, 3, 4, 5}); /// int data[] = {1, 2, 3, 4, 5}

// print(data);

// vector<int> data = {1, 2, 3, 4, 5}; /// int data[] = {1, 2, 3, 4, 5}

// print(data);

// vector<int> data = {1, 2, 3, 4, 5}; /// int data[] = {1, 2, 3, 4, 5}

// vector<int> data2 = data;

// print(data2);

/// add elements

// vector <int> data = {1,2,3};

// data.push\_back(5);

// data.push\_back(7);

// data.push\_back(9);

// print(data);

// vector <int> data = {1,2,3};

// /// data.begin()

// /// data.end()

//

//// data.insert( data.begin() , 7);

//// data.insert( data.begin() + 2, 7);

//

//// data.insert(data.end()-1, 7);

//

//// data.insert(data.begin(), 2, 7);

// data.insert(data.begin() + 1, 2, 7);

//

// print(data);

/// read

// vector <int> data = {3, 2, 1, 6, 4, 2};

// cout << data[2] << "\n";

// cout << data.front() << "\n";

// cout << data.back() << "\n";

// vector <int> data = {3, 2, 1, 6, 4, 2};

// for (int d:data){

// cout << d << " ";

// }

// vector <int> data = {3, 2, 1, 6, 4, 2};

// int ln = data.size();

// for (int i=0; i<ln; ++i){

// cout << data[i] << " ";

// }

// vector <int> data = {3, 2, 1, 6, 4, 2};

// int ln = data.size();

// for (int i=2; i<ln-1; ++i){

// cout << data[i] << " ";

// }

// vector <int> data = {3, 2, 1, 6, 4, 2};

//

// int \*d = data.data();

// cout << d[2];

/// update

// vector <int> data = {3, 2, 1, 6, 4, 2};

// data[2] = 7;

//// cout << data[2] << "\n";

// print(data);

/// swap

// vector <int> data = {3, 2, 1, 6, 4, 2};

// vector <int> data2 = {6, 9, 8};

//

// cout << "before swap\n";

// print(data);

// print(data2);

//

// data.swap(data2);

// cout << "after swap\n";

// print(data);

// print(data2);

/// delete

// vector <int> data = {3, 2, 1, 6, 4, 2};

//

// cout << "Before delete\n";

// print(data);

//

//// data.pop\_back();

//// data.clear();

//// data.erase(data.begin()+2, data.begin()+3);

// data.erase(data.begin()+2, data.end()-2);

//

// cout << "After delete\n";

// print(data);

return 0;

}

|  |
| --- |
| #include<bits/stdc++.h> |
|  |
|  | using namespace std; |
|  |  |
|  | int main() |
|  | { |
|  |  |
|  |  |
|  | int arr[5] = {5, 2, 7, 2, 1}; |
|  |  |
|  | cout << "before \n"; |
|  | for (int i=0; i<5; i++){ |
|  | cout << arr[i] << " "; |
|  | } |
|  | cout << "\n\n"; |
|  |  |
|  | sort(arr, arr+5); |
|  |  |
|  | cout << "after \n"; |
|  | for (int i=0; i<5; i++){ |
|  | cout << arr[i] << " "; |
|  | } |
|  | cout << "\n\n"; |
|  |  |
|  |  |
|  | return 0; |
|  | }  #include <bits/stdc++.h>  using namespace std;  bool myfunc(int a,int b){  return(a>b);  }  int main()  {  // map<string ,int> mp;  // mp["raj"]=100;  // mp["nahid"]=200;  // mp["raju"]=300;  // mp["apel"]=400;  // cout<<mp["raj"]<<endl;  // map<string ,int>::iterator it;  vector <int> vac={1,8,10,10,9,5,6};  vector <int> :: iterator it;  sort(vac.begin(),vac.end());  for(int i=0;i<vac.size();i++){  cout<< vac[i]<<endl;  }  puts("");  sort(vac.begin(),vac.end(), myfunc);  for(it=vac.begin();it!=vac.end();it++){  cout<<\*it<<endl;  }  puts("");  sort(vac.begin(),vac.begin()+5);  for(it=vac.begin();it!=vac.end();it++){  cout<<\*it<<endl;  }  return 0;  }  int main()  {  list<int >les;  list<int>::iterator it;  les.push\_back(10);  les.push\_front(5);  les.push\_back(15);  les.push\_front(2);  for(it=les.begin();it!=les.end();it++){  cout<<\*it<<endl;  }  puts("");  les.reverse();  for(it=les.begin();it!=les.end();it++){  cout<<\*it<<endl;  }  return 0;  }  #include <bits/stdc++.h>  using namespace std;  int main()  {  int ar[10]={1,2,3,4,5,8,6,7};  list<int> lu(ar,ar+8);  list<int>::iterator it;  cout<<lu.front()<<endl;  cout<<lu.back()<<endl;  lu.pop\_front();  lu.pop\_back();  // it=lu.begin();  // it++;  it=find(lu.begin(),lu.end(),6);  if(it==lu.end()){  cout<<"notfound"<<endl;  }  else{  lu.erase(it);  lu.insert(it,10);  for(it=lu.begin();it!=lu.end();it++){  cout<<\*it<<endl;  }  }  return 0;  } |

#include <bits/stdc++.h>

using namespace std;

int main()

{

int ar[5]={1,5,8,9,4};

list<int> oi(ar,ar+5);

if(oi.empty()){

cout<<"empty"<<endl;

}

else{

cout<<"field "<<endl;

}

cout<<oi.front()<<endl;

cout<<oi.back()<<endl;

oi.pop\_back();

oi.pop\_front();

cout<<oi.front()<<endl;

cout<<oi.back()<<endl;

return 0;

}

#include <bits/stdc++.h>

using namespace std;

int main()

{

map<string ,int>m;

map<string, int>::iterator it;

m["Aeshad"]=25;

m["Arshed"]=20;

m["Alamin"]=21;

m["Ahasan"]=25;

m.insert(make\_pair("rakib",22));

for(it=m.begin();it!=m.end();it++){

cout<<it->first<< " "<<it->second<<endl;

}

return 0;

}

Use pair-------------------------------------------------------------------------------------------------------------------------------

#include <bits/stdc++.h>

using namespace std;

int main()

{

pair<int ,string>p(25,"Furkan");

cout<<p.first<< " "<<p.second<<endl;

return 0;

}

Set -------------------------------------------------------------------------------------------------------------------------------

#include <bits/stdc++.h>

using namespace std;

int main()

{

set <int> vac={1,8,10,10,9,5,6};

set <int> :: iterator it;

for(it=vac.begin();it!=vac.end();it++){

cout<< \*it<<endl;

}

return 0;

}

#include <bits/stdc++.h>

using namespace std;

int main()

{

set <string> p;

set <string>::iterator it;

p.insert("arnob");

p.insert("rakib");

p.insert("asik");

p.insert("roboin");

p.insert("nilu");

pair<set<string > ::iterator,bool>s;

if(s.second==false){

cout<<"not insert"<<endl;

}

else{

cout<<"insert"<<endl;

}

for(it=p.begin();it!=p.end();it++){

cout<<\*it<<endl;

}

return 0;

}

Strck------------------------------------------------------------------------------------------------------------------------------------

#include <bits/stdc++.h>

using namespace std;

int main()

{

stack <string> p;

p.push("arnob");

p.push("rakib");

p.push("asik");

p.push("roboin");

p.push("nilu");

while(!p.empty()){

string x;

x=p.top();

cout<<x<<endl;

p.pop();

}

return 0;

}

Priority\_queue---------------------------------------------------------------------------------------------------------------------------------

#include <bits/stdc++.h>

using namespace std;

int main()

{

priority\_queue <string> p;

p.push("arnob");

p.push("rakib");

p.push("asik");

p.push("roboin");

p.push("nilu");

while(!p.empty()){

string x;

x=p.top();

cout<<x<<endl;

p.pop();

}

return 0;}

multimap------------------------------------------------------------multiset-----------------------------------------------------------------

#include <bits/stdc++.h>

using namespace std;

int main()

{

multimap<string ,int>m;

multimap<string, int>::iterator it;

m.insert(make\_pair("rakib",22));

m.insert(make\_pair("rakib",22));

m.insert(make\_pair("rakib",22));

for(it=m.begin();it!=m.end();it++){

cout<<it->first<< " "<<it->second<<endl;

}

return 0;

}

#include <bits/stdc++.h>

using namespace std;

void showq(queue<int> gq)

{

queue<int> g = gq;

while (!g.empty()) {

cout << '\t' << g.front();

g.pop();

}

cout << '\n';

}

// Driver Code

int main()

{

queue<int> gquiz;

gquiz.push(10);

gquiz.push(20);

gquiz.push(30);

cout << "The queue gquiz is : ";

showq(gquiz);

cout << "\ngquiz.size() : " << gquiz.size();

cout << "\ngquiz.front() : " << gquiz.front();

cout << "\ngquiz.back() : " << gquiz.back();

cout << "\ngquiz.pop() : ";

gquiz.pop();

showq(gquiz);

return 0;

}