// Note that by default C++ creates a max-heap

// for priority queue

#include <iostream>

#include <queue>

using namespace std;

void showpq(priority\_queue <int> gq)

{

priority\_queue <int> g = gq;

while (!g.empty())

{

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

g.pop();

}

cout << '\n';

}

int main ()

{

priority\_queue <int> gquiz;

gquiz.push(10);

gquiz.push(30);

gquiz.push(20);

gquiz.push(5);

gquiz.push(1);

cout << "The priority queue gquiz is : ";

showpq(gquiz);

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

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

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

gquiz.pop();

showpq(gquiz);

return 0;

}

// C++ program to demonstrate min heap

#include <iostream>

#include <queue>

using namespace std;

void showpq(priority\_queue <int, vector<int>, greater<int>> gq)

{

priority\_queue <int, vector<int>, greater<int>> g = gq;

while (!g.empty())

{

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

g.pop();

}

cout << '\n';

}

int main ()

{

priority\_queue <int, vector<int>, greater<int>> gquiz;

gquiz.push(10);

gquiz.push(30);

gquiz.push(20);

gquiz.push(5);

gquiz.push(1);

cout << "The priority queue gquiz is : ";

showpq(gquiz);

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

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

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

gquiz.pop();

showpq(gquiz);

return 0;

}