priority\_queue\_ex1

[code]

// constructing priority queues

#include <iostream> // std::cout

#include <queue> // std::priority\_queue

#include <vector> // std::vector

#include <functional> // std::greater

class mycomparison

{

bool reverse;

public:

mycomparison(const bool& revparam=false)

{reverse=revparam;}

bool operator() (const int& lhs, const int&rhs) const

{

if (reverse) return (lhs>rhs);

else return (lhs<rhs);

}

};

int main ()

{

int myints[]= {10,60,50,20};

std::priority\_queue<int> first;

std::priority\_queue<int> second (myints,myints+4);

std::priority\_queue<int, std::vector<int>, std::greater<int> >

third (myints,myints+4);

// using mycomparison:

typedef std::priority\_queue<int,std::vector<int>,mycomparison> mypq\_type;

mypq\_type fourth; // less-than comparison

mypq\_type fifth (mycomparison(true)); // greater-than comparison

return 0;

}

[result]

Don’t produce any output

But it constructs the priority queue using different ways.

More details, see the following link.

[ref]

<https://cplusplus.com/reference/queue/priority_queue/priority_queue/>