#include <iostream>

using namespace std;

class ISorterStrategy

{

public:

virtual void sort() const = 0;

};

class CMergeSorter : public ISorterStrategy

{

public:

virtual void sort() const {

cout << "Merge sort()\n";

}

};

class CQuickSorter : public ISorterStrategy {

public:

virtual void sort() const {

cout << "Quick sort()\n";

}

};

class CHeapSorter : public ISorterStrategy

{

public:

virtual void sort() const {

cout << "Heap sort()\n";

}

};

// Context

class Collection

{

private:

ISorterStrategy\* m\_pSorter;

public:

Collection() { m\_pSorter = nullptr; }

void set\_sort(ISorterStrategy\* pSortStrategy) {

m\_pSorter = pSortStrategy;

}

void sort() const {

m\_pSorter->sort();

}

};

int main(int argc, char\* argv[])

{

CMergeSorter mergeSorter;

CQuickSorter quickSoter;

CHeapSorter heapSorter;

Collection colA;

colA.set\_sort(&mergeSorter);

colA.sort();

Collection colB;

colB.set\_sort(&quickSoter);

colB.sort();

return 0;

}