#include <iostream>

#include <conio.h>

#include <string>

#include <assert.h>

using namespace std;

template<class Type>

class pointerClass

{

Type\* list;

int length;

int maxSize;

public:

pointerClass()

{

list = NULL;

length = 0;

maxSize = 100;

}

pointerClass(int size)

{

if (size < 0)

{

cout << "Size cant be negative" << endl;

maxSize = 100;

}

else

maxSize = size;

length = 0;

list = new Type[maxSize];

assert(list != NULL);

}

bool isEmpty() const

{

return(length == 0);

}

bool isFull() const

{

return(length == maxSize);

}

int listSize() const

{

return length;

}

int maxListSize() const

{

return maxSize;

}

void print()

{

for (int i = 0; i < length; i++)

cout << list[i] << " ";

cout << endl;

}

void insert(int loc, const Type& item)

{

list[loc] = item;

length++;

}

pointerClass<Type> &operator =(const pointerClass<Type>& other)

{

if (this != &other)

{

delete[] list;

maxSize = other.maxSize;

length = other.length;

list = new Type[maxSize];

assert(list != NULL);

for (int i = 0; i < length; i++)

list[i] = other.list[i];

}

return \*this;

}

~pointerClass()

{

delete[]list;

}

};

void main()

{

pointerClass<int> first(5);

pointerClass<int> second;

int data;

cout << "Check whether list is empty " << endl;

cout << first.isEmpty() << endl;

cout << "Check whether list if full" << endl;

cout << first.isFull() << endl;

cout << "Input Values To First Array" << endl;

for (int i = 0; i < 5; i++)

{

cin >> data;

first.insert(i,data);

}

cout << "\n\n Now we assign values of first array \n to second through = operator " << endl;

second = first;

cout << "Following are List 1 and List 2" << endl;

cout << "List 1 : "; first.print();

cout << "List 2 : "; second.print();

\_getch();

}