//partea1

#include "pch.h"

#include <iostream>

using namespace System;

using namespace std;

template <class T> class vector

{

T\* pe;

int dim;

public:

vector(int);

~vector() { delete[] pe; }

T& operator[ ] (int i) { return pe[i]; }

void afis(); void sort();

};

template <class T> vector<T>::vector(int n) :dim(n)

{

pe = new T[n];

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

{

cout << "\n elem\_" << i << " "; cin >> pe[i];

}

}

template <class T> void vector<T>::afis() {

cout << endl;

for (int i = 0; i < dim; i++)cout << pe[i] << " ";

}

template <class T> void vector<T>::sort() {

int i, j; T aux;

for (i = 0; i < dim - 1; i++)

for (j = i + 1; j < dim; j++)

if (pe[i] > pe[j])

{

aux = pe[i]; pe[i] = pe[j]; pe[j] = aux;

}

}

void main() {

vector<int> vi(3);

vi.afis(); vi.sort(); vi.afis();

vector<float> vf(3);

vf.afis(); vf.sort(); vf.afis();

cout << "\n elementul minim este:" << vf[0];

cin >> vi[0];

}

//partea2

#include "pch.h"

#include <cliext\vector>

using namespace System;

using namespace cliext;

int main(array<System::String ^> ^args)

{

vector <double> v1;

for (int i = 0; i < 10; i++)

v1.push\_back(i \* 2.0);

vector<double>::iterator it = v1.begin();

for (; it != v1.end(); it++)

Console::WriteLine(\*it);

return 0;

}