#include <iostream>

#include<cstdlib>

#include<ctime>

#include<iomanip>

using namespace std;

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

template<class T>

class vector{

//friend vector<T> operator\*(T , const vector<T>&);

private:

int dim;

T\*elementos;

public:

vector();

vector(int);

~vector();

void imprimir();

T operator\*(const vector<T>&)const;

};

int main() {

srand(time(0));

//vector u;

vector<float> v(10);

//u.imprimir();

v.imprimir();

vector <float>z=2\*v;

z.imprimir();

//cout<<"v\*z="<<v\*z<<endl;

return 0;

}

template<class T>

vector<T>::vector(){

dim=3;

elementos=new T[dim];

elementos[0]=1;

elementos[1]=1;

elementos[2]=1;

}

template<class T>

vector<T>::vector(int n){

dim=n;

elementos=new T[dim];

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

elementos[i]=rand()%100;

}

template<class T>

vector<T>::~vector(){

delete[]elementos;

}

template<class T>

void vector<T>::imprimir(){

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"(";

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

cout<<setw(4)<<elementos[i];

cout<<")\n";

}

/\*template<class T>

vector<T> operator\*(T k, const vector<T>&u){

vector<T> v(u.dim);

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

v.elementos[i]=k\*u.elementos[i];

return v;

}\*/

template<class T>

T vector<T>:: operator\*(const vector<T>&v)const{

float s=0;

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

s+=elementos[i]\*v.elementos[i];

return s;

}