// Marco RONDELLI – 13/07/20

**Programmazione B**

#include <iostream>

using namespace std;

const int cap = 100;

template <class T>

// vettore di elementi di tipo qualsiasi T

class Vettore{

private:

T\* A; // puntatore all'array di elementi

int dim; // dimensione vettore

public:

Vettore(n){

if(n<0) throw string("Grandezza invalida!");

A = new T[n]; // creo dinamicamente un array di vettori

dim = n;

}

Vettore(n, T j){

if(n<0) throw string("Grandezza invalida!");

A = new T[n];

dim = n;

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

A[i] = j;

}

// "rule of three" siccome allocazione dinamica

// costruttore di copia

Vettore(const Vettore<T>& V) {

this->A = new T[V.dim];

this->dim = V.dim;

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

this->A[i] = V.A[i];

}

// distruttore

~Vettore(){

delete[] A;

}

// funzione che ritorna l'elemento i-esimo

T& operator[](int i){

if(i<0 || i>dim)

throw string("Out of range!");

return A[i]; // reference return evita creazione di copia elemento

}

// copia profonda dei due vettori

Vettore<T>& operator=(const Vettore<T>& V) {

delete[] this->A;

this->A = new T[V.dim];

this->dim = V.dim;

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

this->A[i] = V.A[i];

return \*this; // ritorniamo l'oggetto copiato stesso

}

// determina l'uguaglianza degli elementi in posizione i-esima

// va dichiarato const, non vogliamo modifica accidentale del valore,

// inoltre Vettore<T>& è const quindi la funzione deve essere const

bool operator==(const Vettore<T>& V) const{

if(this->dim != V.dim) return false;

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

if(this->A[i] != V.A[i]) return false;

return true;

}

// stampa tutti gli elementi

void stampa(ostream& f\_out) const{

f\_out << '[';

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

f\_out << A[i];

f\_out << ' ';

}

f\_out << ']';

}

};

template <class T>

ostream& operator<<(ostream& f\_out, const Vettore<T> &V1) {

V1.stampa(f\_out);

return f\_out;

}

int main(){

try{

Vettore<int> A(cap-1);

Vettore<float> B(cap-1);

int i;

cin >> i;

A[i] = 3;

B[i] = 2.5;

cout << A[i] << endl;

cout << B[i] << endl;

Vettore<int> C(cap-1);

C = A;

if(A == C) cout<<"Vettori uguali!"<<endl;

else cout<<"Vettori diversi!"<<endl;

}

catch(string s){

cout << "Eccezione! "<< e << endl;

}

return 0;

}