

Sia B una classe polimorfa e sia C una sottoclasse di B. Definire una funzione `int Fun(const vector<B*>& v)` con il seguente comportamento: sia `v` non vuoto e sia `T*` il tipo dinamico di `v[0]`; allora `Fun(v)` ritorna il numero di elementi di `v` che hanno un tipo dinamico `T1*` tale che `T1` è un sottotipo di `C` diverso da `T`; se `v` è vuoto deve quindi ritornare 0. Ad esempio, il seguente programma deve compilare e provocare le stampe indicate.

```
#include<iostream>
#include<typeinfo>
#include<vector>
using namespace std;

class B {public: virtual ~B() {} };
class C: public B {};
class D: public B {};
class E: public C {};

int Fun(vector<B*> &v){...}
```

```
main() {
    vector<B*> u, v, w;
    cout << Fun(u); // stampa 0
    B b; C c; D d; E e; B *p = &e, *q = &c;
    v.push_back(&c); v.push_back(&b); v.push_back(&d); v.push_back(&c);
    v.push_back(&e); v.push_back(p);
    cout << Fun(v); // stampa 2
    w.push_back(p); w.push_back(&d); w.push_back(q); w.push_back(&e);
    cout << Fun(w); // stampa 1
}
```