

Esercizio 3

```
class Z {
public: Z(int x) {}
};

class A {
public:
    void f(int){cout << "A::f(int) ";}
    virtual void f(bool){cout <<"A::f(bool) ";}
    virtual void f(Z){cout <<"A::f(Z) ";}
};

class B: virtual public A {
public:
    void f(const bool&){cout<< "B::f(const bool&) ";}
    void f(const int&){cout<< "B::f(const int&) ";}
};

class C: virtual public A {
public:
    virtual void f(Z){cout <<"C::f(Z) ";}
};

class D: public B, public C {
public:
    virtual void f(int*){cout<< "D::f(int*) ";}
    void f(int&){cout <<"D::f(int&) ";}
};

class E: public D {
public:
    void f(Z){cout <<"E::f(Z) ";}
};

B* pb=new B; C* pc = new C; D* pd = new D; E* pe = new E; A *pa1=pb, *pa2=pc, *pa3=pd, *pa4=pe; C *pc1=pe;
```

Le precedenti definizioni compilano correttamente. Per ognuna delle seguenti istruzioni scrivere nell'apposito spazio:

- **NON COMPILA** se la compilazione dell'istruzione provoca un errore;
- **ERRORE RUN-TIME** se l'istruzione compila correttamente ma la sua esecuzione provoca un errore a run-time;
- se l'istruzione compila correttamente e non provoca errori a run-time allora si scriva la stampa che l'esecuzione produce in output su `std::cout`; se non provoca alcuna stampa allora si scriva **NESSUNA STAMPA**.

<code>(dynamic_cast<B*>(pa1)) ->f(1);</code>
<code>(dynamic_cast<B*>(pa1)) ->f(true);</code>
<code>pa1->f(true);</code>
<code>pa2->f(1);</code>
<code>(dynamic_cast<C*>(pa2)) ->f(1);</code>
<code>(dynamic_cast<E*>(pa2)) ->f(1);</code>
<code>(dynamic_cast<C*>(pa3)) ->f(0);</code>
<code>(dynamic_cast<D*>(pa3)) ->f(0);</code>
<code>pa4->f(1);</code>
<code>(dynamic_cast<C*>(pa4)) ->f(1);</code>
<code>pc1->f(1);</code>
<code>(static_cast<E*>(pc1)) ->f(1);</code>
<code>(static_cast<A*>(pc1)) ->f(1);</code>