

Esercizio 2

```
class A {
protected:
    virtual void h() {cout<<" A::h ";}
public:
    virtual void g() const {cout <<" A::g ";}
    virtual void f() {cout <<" A::f "; g(); h();}
    void m() {cout <<" A::m "; g(); h();}
    virtual void k() {cout <<" A::k "; h(); m(); }
    virtual A* n() {cout <<" A::n "; return this;}
};

class B: public A {
protected:
    virtual void h() {cout <<" B::h ";}
public:
    virtual void g() {cout <<" B::g ";}
    void m() {cout <<" B::m "; g(); h();}
    void k() {cout <<" B::k "; g(); h();}
    B* n() {cout <<" B::n "; return this;}
};

class C: public B {
protected:
    virtual void h() const {cout <<" C::h ";}
public:
    virtual void g() {cout <<" C::g ";}
    void m() {cout <<" C::m "; g(); k();}
    void k() const {cout <<" C::k "; h();}
};

A* p2 = new B(); A* p3 = new C(); B* p4 = new B(); B* p5 = new C(); const A* p6 = new C();
```

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 `cout`; se non provoca alcuna stampa allora si scriva **NESSUNA STAMPA**.

p2->f();	.....
p2->m();	.....
p3->k();	.....
p3->f();	.....
p4->m();	.....
p4->k();	.....
p4->g();	.....
p5->g();	.....
p6->k();	.....
p6->g();	.....
(p3->n())->m();	.....
(p3->n())->g();	.....
(p3->n())->n()->g();	.....
(p5->n())->g();	.....
(p5->n())->m();	.....
(dynamic_cast<B*>(p2))->m();	.....
(static_cast<C*>(p3))->k();	.....
( static_cast<B*>(p3->n()) )->g();	.....