

```

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

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

const B* p1 = new D(); B* p2 = new C(); B* p3 = new D(); C* p4 = new E(); B* p5 = new E();

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

class E: public C {
protected:
    void h() {cout <<" E::h ";}
public:
    void m() {cout <<" E::m "; g(); h();}
    C* n() {cout <<" E::n "; return this;}
};

```

Si considerino le precedenti definizioni che compilano correttamente. Per ognuno dei seguenti 16 statement scrivere nell'apposito riquadro:
UNA RISPOSTA PER OGNI LINEA CHE VA NUMERATA DA 01 A 16 ANCHE IN CASO DI NESSUNA RISPOSTA:

- **NC** se lo statement Non Compila correttamente;
- **ERT** se lo statement compila correttamente ma la sua esecuzione provoca un Errore a Run Time;
- se lo statement compila correttamente e non provoca errori a run-time allora si scriva **precisamente** la stampa che l'esecuzione produce in output su `cout`; se provoca Nessuna Stampa allora si scriva **NS**.
- **??** in caso di nessuna risposta.

```

01: p1->g();
02: (p1->n())->g();
03: p2->f();
04: p2->m();
05: (static_cast<D*>(p2))->k();
06: p3->k();
07: p3->f();
08: (p3->n())->m();
09: (p3->n())->n()->g();
10: (static_cast<C*>(p3->n()))->g();
11: p4->m();
12: p4->k();
13: (p4->n())->m();
14: p5->g();
15: p5->k();
16: (dynamic_cast<C*>(p5))->m();

```