

Si considerino le seguenti definizioni di classe e funzione:

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

char F(A* p, C& r) {
    B* punt1 = dynamic_cast<B*> (p);  $A \leq B$ 
    try{
        E& s = dynamic_cast<E&> (r);  $C \leq D$ 
    }
    catch (bad_cast) {
        if (punt1) return 'O';  $\leftarrow A \leq B(V)$ 
        else return 'M';  $\leftarrow A \neq B$ 
    }
    if (punt1) return 'R';  $\leftarrow A \leq B(V)$ 
    return 'A';
}
```

$C \neq B$ —

A, C

Si consideri inoltre il seguente main() incompleto, dove ? è semplicemente un simbolo per una incognita:

```
main() {
    A a; B b; C c; D d; E e;  $(A, C)$ 
    cout << F(B, E) << F(B, C) << F(A, D) << F(A, C);
}
```

$\frac{B}{P} \quad \frac{B}{O} \quad \frac{A}{M} \quad \frac{A}{A}$

Definire opportunamente le chiamate in tale main() usando gli oggetti a, b, c, d, e locali al main() in modo tale che la sua esecuzione provochi la stampa ROMA.