

Esercizio 3

Si considerino le seguenti definizioni.

```
#include<iostream>
#include<string>
using namespace std;

class Z {
public:
    operator int() const {return 0;}
};

template<class T> class D; // dichiarazione incompleta

template<class T1, class T2 = Z, int k = 1>
class C {
    friend class D<T1>;
private:
    T1 t1;
    T2 t2;
    int a;
    C(int x =k): a(x) {}
};

template<class T>
class D {
public:
    void f() const {C<T,T> c(1); cout << c.t1 << c.t2 << c.a;}
    void g() const {C<int> c;}
    void h() const {C<T,int> c(3); cout << c.t2 << c.a;}
    void m() const {C<int,T,3> c; cout << c.t1;}
    void n() const {C<int,double> c; cout << c.t1 << c.t2 << c.a;}
    void o() const {C<char,double> c(6); cout << c.a;}
    void p() const {C<Z,T,7> c(7); cout << c.t2 << c.a;}
};
```

Determinare se i seguenti main() compilano correttamente o meno barrando la corrispondente scritta.

| | |
|---|-------------|
| int main() { D<char> d1; d1.f(); } | COMPILA |
| int main() { D<std::string> d2; d2.f(); } | COMPILA |
| int main() { D<char> d3; d3.g(); } | NON COMPILA |
| int main() { D<int> d4; d4.g(); } | COMPILA |
| int main() { D<char> d5; d5.h(); } | COMPILA |
| int main() { D<int> d6; d6.h(); } | COMPILA |
| int main() { D<char> d7; d7.m(); } | NON COMPILA |
| int main() { D<int> d8; d8.m(); } | COMPILA |
| int main() { D<char> d9; d9.n(); } | NON COMPILA |
| int main() { D<Z> d10; d10.n(); } | NON COMPILA |
| int main() { D<char> d11; d11.o(); } | COMPILA |
| int main() { D<Z> d12; d12.o(); } | NON COMPILA |
| int main() { D<char> d13; d13.p(); } | NON COMPILA |
| int main() { D<Z> d14; d14.p(); } | COMPILA |