

Esercizi di Programmazione ad Oggetti

Lista n. 2

Esercizio 1

```
class Z {
public:
    ...
};

template <class T1, class T2=Z>
class C {
public:
    T1 x;
    T2* p;
};

template<class T1, class T2>
void fun(C<T1, T2>* q) {
    ++(q->p);
    if(true == false) cout << ++(q->x);
    else cout << q->p;
    (q->x)++;
    if(*(q->p) == q->x) *(q->p) = q->x;
    T1* ptr = &(q->x);
    T2 t2 = q->x;
}

main() {
    C<Z> c1; fun(&c1); C<int> c2; fun(&c2);
}
```

Si considerino le precedenti definizioni. Fornire una dichiarazione (non è richiesta la definizione) dei membri pubblici della classe `Z` nel **minor numero possibile** in modo tale che la compilazione del precedente `main()` non produca errori.

Esercizio 2

Definire un template di classe `albero<T>` i cui oggetti rappresentano un **albero 3-ario** ove i nodi memorizzano dei valori di tipo `T` ed hanno 3 figli (invece dei 2 figli di un usuale albero binario). Il template `albero<T>` deve soddisfare i seguenti vincoli:

1. Deve essere disponibile un costruttore di default che costruisce l'albero vuoto.
2. Gestione della memoria senza condivisione.
3. Overloading dell'operatore di uguaglianza.
4. Overloading dell'operatore di output.