

**FONDAMENTI DI INFORMATICA E
PROGRAMMAZIONE A OGGETTI**

Una *Lista di sfere* è in grado di contenere un numero illimitato di sfere di materiale diverso. I possibili materiali sono vetro e legno. Le sfere sono disposte nella *Lista di sfere* in modo che non vi siano mai due o più sfere consecutive dello stesso materiale. Le operazioni che possono essere effettuate su una *Lista di sfere* sono le seguenti:

- `ListaSfere p;`
Costruttore di default, che inizializza una lista di sfere `p`. Inizialmente la lista di sfere non contiene elementi.
- `p.inserisci(m)`
Funzione che inserisce una sfera di materiale `m` nella lista di sfere `p`. Il nuovo elemento viene inserito nella posizione più vicina all'inizio della lista tale che dopo l'inserzione non vi siano due o più sfere consecutive dello stesso materiale. Se non è possibile rispettare questa condizione, l'inserzione fallisce e la lista non viene modificata.
- `p.elimina(m)`
Funzione che estrae una sfera di materiale `m` dalla lista di sfere `p`. La sfera da estrarre è quella che occupa la posizione più vicina all'inizio della lista tale che dopo l'estrazione non vi siano due o più sfere consecutive dello stesso materiale. Se non è possibile rispettare questa condizione, l'estrazione fallisce e la lista non viene modificata.
- `cout << p`
Operatore di uscita per il tipo `ListaSfere`. L'uscita consiste nel materiale degli elementi che formano la lista di sfere `p`, separati da virgole e racchiusi tra parentesi angolate. Esempio:
`<LEGNO, VETRO, LEGNO, VETRO>`
In questo esempio, la lista è formata da quattro sfere. La prima sfera è di legno, l'ultima sfera è di vetro.
- `p % k`
Operatore di modulo che restituisce una `ListaSfere` che contiene i primi `k` elementi di `p`.
- `~p`
Operatore di complemento bit a bit, che modifica la lista di sfere `p` cambiando il materiale di ogni sfera della lista.
- `ListaSfere p1(p);`
Costruttore di copia, che inizializza una lista di sfere `p1` col valore della lista di sfere `p`.
- `p1 = p`
Operatore di assegnamento, che sostituisce il valore della lista di sfere `p1` con quello della lista di sfere `p`.
- `~ListaSfere()`
Distruttore.

Mediante il linguaggio C++, realizzare il tipo di dati astratti `ListaSfere`, definito dalle precedenti specifiche. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA:

- Effettuare il **login**
Nome: studenti
Password: studenti
- Aprire il *Dev-C++* (dal Menù *Avvio* (o *Start*) nella barra degli strumenti in fondo allo schermo, selezionare *Programmi* e quindi *Dev-C++*)
- **Prima di iniziare a svolgere l'elaborato**, selezionare la voce *Identifica studente* nel menù *Strumenti* all'interno dell'ambiente *Dev-C++* e inserire i dati richiesti
- Dal menu *File* del *Dev-C++*, aprire il progetto *esainf.dev* presente nella cartella *c:\esame\esainf*. Il progetto contiene tre file, denominati *compito.h*, *compito.cpp* e *main.cpp*
- Scrivere la dichiarazione della classe nel file *compito.h* e la definizione delle funzioni nel file *compito.cpp*. Il file *main.cpp* contiene la funzione principale *main()* che serve a verificare le funzioni scritte nel file *compito.cpp*. Il file *main.cpp* può essere modificato. **Si tenga presente, comunque, che in sede di valutazione dell'elaborato verrà considerato esclusivamente il contenuto dei file *compito.h* e *compito.cpp***
- Per una corretta stampa dell'elaborato **mantenersi entro i margini imposti dall'ambiente Dev-C++** cioè entro la linea verticale presente alla destra della pagina;
- **Per la Consegna:**
 - Selezionare la voce *Consegna* dal menù *Strumenti* (o *Tools*) all'interno dell'ambiente *Dev-C++* e premere il tasto INVIO fino a quando non viene chiusa la finestra che è stata attivata.
 - Aspettare al proprio posto di essere chiamati per verificare la stampa del proprio elaborato, firmarlo e **consegnarlo definitivamente**.

REGOLA PER LA CORREZIONE:

Per la correzione dell'elaborato è necessario che la definizione della classe con le funzioni

- costruttore (**ListaSfere p**)
- funzioni per l'inserimento e l'eliminazione di una sfera (**p.inserisci(m)**, **p.elimina(m)**)
- operatore di uscita (**operator<<**)

siano implementate correttamente.

Il codice di altre funzioni che impedisce la compilazione e l'esecuzione del progetto deve essere messo come commento e verrà comunque valutato in sede di correzione dell'elaborato.
