

Una **Molecola** poliatomica è una descrizione di quali e quanti atomi compongono una data sostanza. Un atomo è identificato da una sigla costituita da due caratteri al massimo e da un intero strettamente positivo che indica la quantità di quell'atomo nella molecola.

Implementare le seguenti operazioni che possono essere effettuate su di una **Molecola**:

--- **PRIMA PARTE** --- (qualora siano presenti errori di compilazione, collegamento o esecuzione in questa parte, l'intera prova sarà considerata insufficiente e pertanto **non sarà corretta**)

✓ **Molecola m;**

Costruttore di default, che inizializza una molecola m. Inizialmente la molecola è vuota.

✓ **m.aggiungi(s, n);**

Operazione che, se necessario, aggiunge un nuovo tipo di atomo alla **Molecola** m. L'atomo ha sigla s e quantità n. Se la **Molecola** m contiene già l'atomo s, l'operazione aggiunge n alla quantità dell'atomo s in m.

✓ **cout << m;**

Operatore di uscita per il tipo **Molecola**. L'uscita ha il seguente formato:

<H2-Si5-C1-Fe3>

L'output mostrato corrisponde a una molecola che descrive una sostanza in cui sono stati aggiunti 3 atomi di ferro (Fe), 1 atomo di Carbonio (C), 5 atomi di silicio (Si) ed infine 2 atomi di Idrogeno (H). **Gli atomi aggiunti per ultimo debbono essere mostrati per primi.**

Nel caso di molecola senza atomi l'uscita deve essere <>.

✓ **m.elimina(s);**

Operazione che elimina l'atomo con sigla s dalla molecola m. Se l'atomo s non è presente nella molecola, l'operazione la lascia inalterata.

--- **SECONDA PARTE** ---

✓ **m += m1;**

Operatore di somma e assegnamento che, se necessario, aggiunge nuovi atomi a m in modo tale che la sigla di ciascun atomo di m1 sia presente anche in m. Per gli atomi presenti sia in m che in m1, la quantità è la somma delle quantità in m e m1. Per gli atomi presenti solo in m, la quantità rimane invariata. Per gli atomi presenti solo in m1, la quantità è quella in m1.

Esempio: Qualora m1 fosse **<C3-Li2>** ed m **<Si5-C1>**, dopo l'operazione m dovrebbe valere:

<Li2-Si5-C4>

✓ **m1 = m;**

Operazione che modifica m1, sostituendo al suo precedente contenuto quello di m.

~Molecola();

Distruttore.

Mediante il Linguaggio C++, realizzare il tipo di dato astratto **Molecola**, definito dalle precedenti specifiche. **Gestire le eventuali situazioni di errore.**

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA

AVVIO E IDENTIFICAZIONE

- Avviare la macchina in modalità diskless, scegliere “Fondamenti di Informatica I” ed effettuare il login: **nome:** studenti **password:** studenti
- Aprire un terminale e al prompt spostarsi sulla cartella ‘elaborato’ (`$ cd ~/elaborato`). Si utilizzi il comando `pwd` per verificare che ci si trovi nella cartella corretta `/home/studenti/elaborato`.
- Sempre al prompt dare il comando `ident`, sempre da dentro la cartella. Lo script richiede i propri dati (cognome, nome, numero di matricola e password (la password **non va dimenticata** in quanto è indispensabile per scaricare da internet il proprio elaborato a consegna avvenuta). Il comando `ident` crea il file `matricola.txt` nella cartella corrente. Lo script può essere lanciato più volte, in tal caso il file `matricola.txt` viene sovrascritto. Per verificare che il file sia stato creato e che il contenuto sia quello giusto dare il comando (la password è codificata):

```
$ cat /home/studenti/elaborato/matricola.txt
```

- A questo punto il docente verifica che tutti gli studenti abbiano effettuato l’identificazione, dopodiché provvede a inviare i seguenti file nella cartella `elaborato` del proprio PC: `compito.h`, `compito.cpp`, `main.cpp`. Controllare pertanto che questi file, insieme al file `matricola.txt`, siano presenti sul proprio elaboratore.

SVOLGIMENTO DELLA PROVA

- Definire ed implementare il tipo di dato astratto richiesto e le relative funzioni nei file `compito.h` e `compito.cpp`. Il file `main.cpp` contiene la funzione principale `main()` ed è utilizzato dallo studente per testare la sua implementazione della classe. Il file `main.cpp` può essere modificato a piacere. In sede di valutazione dell’elaborato verrà considerato **esclusivamente il contenuto dei file `compito.h` e `compito.cpp`** ed è pertanto **vietato cambiare nome a tali file**.

Per compilare e linkare dare il comando:

```
$ g++ main.cpp compito.cpp (eseguibile invocabile tramite $ ./a.out)
```

(utilizzare `g++ -g` per includere le informazioni di debug qualora si intenda debuggare con `ddd`).

PER CONSEGNARE O RITIRARSI

Recarsi dal docente **dopo aver preso nota dell’identificatore della macchina** (esempi: `g34`, `s23`, `c22`, ...).

USCITA CHE DEVE PRODURRE IL PROGRAMMA

Test del costruttore e della aggiungi. Deve stampare `<H2-Si5-C1-Fe3>`
<H2-Si5-C1-Fe3>

Altro test della aggiungi. Deve stampare `<H5-Si5-C1-Fe3>`
<H5-Si5-C1-Fe3>

Test della elimina. Deve stampare `<Si5-C1>`
<Si5-C1>

Test operatore +=. Deve stampare `<Li2-Si5-C4>`
<Li2-Si5-C4>

Test dell'op. di assegnamento. Deve stampare `<Li2-Si5-C4>`
<Li2-Si5-C4>

Test del distruttore: `m1` e' stato distrutto (non deve stampare nulla)