

Una Concessionaria di auto sportive (*supercar*) gestisce un parco auto di dimensioni potenzialmente illimitate. Ogni supercar è caratterizzata da un nome. Il nome può essere una qualunque stringa alfanumerica con eventuali spazi, *purché lunga almeno un carattere*.



Implementare le seguenti operazioni che possono essere compiute su una Concessionaria:

--- **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)

✓ `Concessionaria c;`

Costruttore di default per una Concessionaria. All'inizio non sono presenti supercar.

✓ `c+=nome;`

Operazione che aggiunge la supercar `nome` alla concessionaria `c` se `nome` è una stringa valida. NB: in una concessionaria possono essere presenti anche più auto con lo stesso nome.

✓ `cout<<c;`

Operatore di uscita di una Concessionaria. L'uscita deve avere prima la stampa del numero di supercar, seguita dal carattere ':', e poi i nomi delle supercar, precedute da "=>". Le supercar vanno mostrate **in ordine alfabetico crescente**.

Esempio: Se la concessionaria contiene una "FERRARI F150", una "BUGATTI VEYRON" e una "MCLAREN F1", l'uscita deve essere la seguente:

```
3:=>BUGATTI VEYRON=>FERRARI F150=>MCLAREN F1
```

✓ `int(c)`

Operazione che converte una concessionaria `c` in un intero, corrispondente al numero di supercar attualmente presenti in `c` (restituisce 0 nel caso non ve ne siano).

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

✓ `c-=k;`

Funzione che modifica la concessionaria `c` eliminando le prime `k` supercar da `s` ("prime" secondo l'ordine alfabetico crescente).

Esempio: Se la concessionaria contiene una "FERRARI F150", una "BUGATTI VEYRON" e una "MCLAREN F1" e si chiama il `-=` con `k = 2`, dovranno essere eliminate "BUGATTI VEYRON" e "FERRARI F150". La `cout<<c;` a questo punto visualizzerà:

```
1:=>MCLAREN F1
```

✓ `c.cerca(str);`

Operazione che conta **quante supercar** (non quante occorrenze in assoluto) hanno nome contenente la sottostringa `str`. La funzione restituisce tale numero. `str` deve essere una stringa di *almeno un carattere*.

NB: E' vietato l'uso della funzione `strstr` della libreria `cstring`.

Esempio: Se `c` contiene "BUGATTI VEYRON, FERRARI F150 e MCLAREN F1" e se `str` è la stringa "AR", la funzione deve restituire **2**, per via di FERR(AR)I e MCL(AR)EN.

✓ `~Concessionaria();`

Distruzione della classe Concessionaria.

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

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 abbiamo 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’identificativo della macchina** (esempi: g34, s23, c22, ...).

USCITA CHE DEVE ESSERE PRODOTTA DAL PROGRAMMA

```
Test 1: costruttore, op. += ed op. << (deve stampare '1:=>FERRARI F150' )
1:=>FERRARI F150
```

```
Test 2: ulteriori test (deve st. '3:=>BUGATTI...=>FERRARI...=>MCLAREN...' )
3:=>BUGATTI VEYRON=>FERRARI F150=>MCLAREN F1
```

```
Test 3: conversione ad intero (deve stampare 3)
3
```

```
Test 4: funzione 'cerca' (deve st. '2', per via di FERR(AR)I e MCL(AR)EN )
2
```

```
Test 5: op. -= (deve stampare '1:=>MCLAREN F1' )
1:=>MCLAREN F1
```

```
Test 6: distruttore (non deve stampare nulla)
```