

FONDAMENTI DI INFORMATICA I
FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI

Una SeqMat è una sequenza, potenzialmente illimitata, di matrici quadrate di interi di dimensioni arbitrarie e variabili l'una dall'altra. Tutte le matrici hanno la seguente caratteristica: sono composte da **elementi tutti uguali**. Utilizzando la rappresentazione che **minimizza il più possibile l'occupazione di memoria** delle singole matrici, implementare la classe SeqMat in modo tale da potervi effettuare le seguenti operazioni:

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

✓ **SeqMat s;**

Costruttore che crea una sequenza vuota, ossia senza alcuna matrice.

✓ **s.insMat(size, value);**

Operazione che aggiunge in coda alla sequenza una nuova matrice quadrata di dimensioni `size` x `size`, con tutti gli elementi pari a `value`.

✓ **cout<<s;**

Operatore di uscita per il tipo SeqMat. Un esempio di uscita è il seguente:

```
INIZIO_SEQUENZA
3333
3333
3333
3333
11
11
55
55
9
7777
7777
7777
7777
44
44
FINE_SEQUENZA
```

dove viene mostrata una sequenza di 6 matrici, di dimensioni 4, 2, 2, 1, 4 e 2, rispettivamente, inizializzate rispettivamente con i valori 3, 1, 5, 9, 7 e 4. Si noti come non vengono aggiunti spazi fra gli elementi di una stessa matrice. Aggiungere INIZIO_SEQUENZA e FINE_SEQUENZA prima e dopo la sequenza di matrici.

✓ **s-=size;**

Operatore che modifica `s` eliminando tutte le matrici di dimensioni `size`, dove `size` è un intero, lasciando immutato l'ordine delle altre matrici.

Esempio. Nell'esempio di sopra, se si eseguisse `s-=4` si otterrebbe:

```
INIZIO_SEQUENZA
11
11
55
55
9
44
44
FINE_SEQUENZA
```

SECONDA PARTE (*si invita a mettere sotto commento le operazioni di questa seconda parte che dovessero impedire la compilazione, il collegamento o la corretta esecuzione del codice*)

✓ **~SeqMat();**

Distruttore.

✓ `~s;`

Operazione che modifica la sequenza di matrici `s` fondendo tutte le eventuali matrici **consecutive di ugual dimensione** in un'unica matrice. La matrice risultato della fusione di due o più matrici ha come elemento (i,j) la somma degli elementi (i,j) di tutte le matrici coinvolte.

Esempio. Se `s` è la sequenza di matrici dell'esempio precedente (ossia dopo la rimozione delle matrici di dimensione 4), eseguendo `cout<<~s` si deve ottenere:

```
INIZIO_SEQUENZA
66
66
9
44
44
FINE_SEQUENZA
```

✓ `s2=s;`

Operatore di assegnamento, che sostituisce il valore della `SeqMat` risultato `s2` con quello della sequenza di matrici `s`.

✓ `++s;`

Operatore di incremento prefisso, che incrementa le **dimensioni** di ciascuna matrice di uno.

Esempio. Sia `s` la sequenza fornita al passo precedente, la `cout<<++s` deve stampare:

```
INIZIO_SEQUENZA
666
666
666
99
99
444
444
444
FINE_SEQUENZA
```

Mediante il Linguaggio C++, realizzare il tipo di dato astratto `SeqMat`, definito dalle precedenti specifiche.

Individuare le 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 avendo preso nota dell'identificativo della macchina (g34, s23, ...).