

Un `Disco` rigido (molto semplificato) memorizza file ed è composto da una sequenza di N settori, ciascuno capace di memorizzare 1024 byte. Qualora il numero di byte di un file sia maggiore di 1024, esso viene suddiviso in blocchi di dimensione 1024 (l'ultimo generalmente risulterà riempito solo parzialmente) e ciascun blocco viene memorizzato in un settore differente del disco, non necessariamente contigui. Un settore non può pertanto contenere porzioni di file differenti. Un file è identificato da un numero intero univoco maggiore o uguale a 1 (al primo file che verrà memorizzato verrà assegnato identificatore 1, al secondo identificatore l'identificatore 2, ecc.). Implementare il tipo di dato astratto `Disco`, dotandolo delle seguenti operazioni:

--- **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)*

✓ **`Disco d(num)`** ;

Costruttore che crea un disco con `num` settori. Inizialmente i settori sono tutti liberi. Al settore libero viene associato, per convenzione, l'identificatore zero.

✓ **`d.riserva(dim)`** ;

Operazione che sceglie quali settori riservare al nuovo file di dimensione `dim` byte da memorizzare. Se il numero di settori richiesto per memorizzare il file è minore o uguale al numero di settori liberi, la funzione riserva al file i primi settori liberi che trova a partire dall'inizio. L'identificatore utilizzato per identificare il file (un numero progressivo che inizia da 1) verrà restituito dalla funzione. Nel caso non ci siano settori a sufficienza a memorizzare tutti i blocchi del file, l'operazione lascia il disco inalterato e restituisce 0.

✓ **`cout << d`** ;

Operatore di uscita per il tipo `Disco`.

Nel caso di un disco di 5 settori che non memorizza file, l'uscita sarebbe : **00000**.

Nel caso di un disco di 8 settori in cui sia stata riservata la memorizzazione di un file che occupi 2 settori, poi di uno che ne occupi 3 ed infine di uno che ne occupi 1, l'uscita sarebbe: **11222300**

✓ **`d.cancella(id)`** ;

Operazione che sposta nel cestino il file con identificatore `id`, qualora questo sia presente (altrimenti non fa nulla). Cancellare un file consiste nella liberazione di tutti i settori ad esso riservati in precedenza (che tornano ad assumere l'identificatore 0).

Quando un file viene cancellato, il suo identificatore **non viene più riutilizzato** per i prossimi file. L'unica eccezione a questa regola è quando il disco viene formattato (si veda la prossima operazione), nel qual caso si ricomincia a numerare i file da 1.

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

✓ **`!d`** ;

Operazione che formatta il disco. Questa operazione cancella tutti i file precedentemente memorizzati e riporta ad uno l'identificatore del prossimo file da riservare.

✓ **`d.deframmenta()`** ;

Operazione che deframmenta il disco. Tutti i blocchi del file con identificatore maggiore debbono essere spostati all'inizio del disco, in settori contigui. Nei settori immediatamente successivi vanno spostati tutti i blocchi del file avente secondo identificatore maggiore, e così via. Alla fine del disco verranno spostati tutti gli eventuali settori liberi presenti.

Esempio: Se in un dato istante lo stato del disco è **11444340**, dopo la deframmenta sarà: **44443110**

✓ `d.getQuantiDischi()` ;

Operazione che restituisce il numero di dischi presenti in memoria in questo momento (ossia quelli creati ma non ancora distrutti).

✓ `~Disco()` ;

Distruttore.

Mediante il Linguaggio C++, realizzare il tipo di dato astratto **Disco**, 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 costr., della 'riserva' e dell'op. <<. Deve stampare: '11000000'
11000000

Altri test della 'riserva'. Deve stampare: '11222300'
11222300

Test della 'cancella' del file con id==2. Deve stampare: '11000300'
11000300

Altro test della 'riserva'. Deve stampare: '11444340'
11444340

Test della 'deframmenta'. Deve stampare '44443110'
44443110

Test dell'operatore ! di formattazione. Deve stampare '11111200'
11111200

Test distruttore e 'getQuantiDischi'. Deve stampare 2 (d2 viene distrutto)
2
```