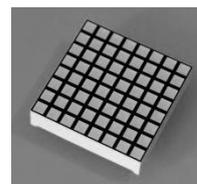


FONDAMENTI DI INFORMATICA I  
 FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI

Un Display digitale a led di dimensioni  $N \times N$  è un dispositivo contenente una griglia quadrata di led che possono essere accesi o spenti singolarmente (si veda figura a lato).



Si implementi il tipo di dato astratto Display utilizzato per astrarre una tale periferica hardware. Il tipo Display deve fornire 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)

✓ **Display d(N) ;**

Costruttore che crea un display di  $N \times N$  led. Inizialmente tutti i led sono spenti.  $N$  deve essere **positivo e dispari**, altrimenti  $N$  viene posto a 5.

✓ **d.set(ch) ;**

Operazione che accende alcuni led a seconda del valore del carattere  $ch$ . I led non coinvolti conservano il precedente stato. In particolare:

- se  $ch$  è un trattino (carattere '-'), accende tutti i led che si trovano sulla *riga centrale*.
- se  $ch$  è un'asta (carattere '|'), accende tutti i led della *colonna centrale*.
- se  $ch$  è uno slash (carattere '/'), accende solo la *diagonale ascendente* della matrice.

Se  $ch$  è un carattere diverso dai precedenti, la set lascia il Display immutato.

✓ **cout<<d;**

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

```
00000
00000
11111
00000
00000
```

dove viene mostrato un Display di dimensione 5, nel quale viene stampato il carattere '1' per indicare che il led in posizione (x,y) è acceso ('0' qualora fosse spento). L'operatore **non** aggiunge *new line* né prima né dopo la sequenza di zero e uno. Altri esempi sono:

00100	00001
00100	00010
00100	00100
00100	01000
00100	10000

[nel caso di set('|')]

[nel caso di set('/')]

**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)

✓ **d1 |=d2 ;**

Operazione di or bit a bit che modifica  $d1$  in modo che contenga il led acceso in posizione (x,y) se lo era già o se è acceso il led di  $d2$  che si trova nella medesima posizione. Nel caso i due Display abbiano dimensione diversa, **lascia d1 immutato**. Ecco un esempio:

00000	00100	00100
00000	00100	00100
11111	00100	11111
00000	00100	00100
00000	00100	00100

[precedente d1]

[d2]

[nuovo d1]

✓ **d.specchia()** ;

Operazione che “inverte” lo stato dei led lungo l’asse verticale di simmetria:

00001	10000
00010	01000
00100	00100
01000	00010
10000	00001
[prima]	[dopo]

✓ **d4=d2** ;

Operatore di assegnamento, che sostituisce il valore del Display risultato d4 con quello del Display d2.

✓ **~Display()** ;

Distruttore.

Mediante il Linguaggio C++, realizzare il tipo di dato astratto `Display`, 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 spostarsi sulla cartella ‘elaborato’ (`$ cd ~/elaborato`). Si utilizzi il comando `pwd` per verificare che ci si trovi nella cartella corretta `/home/studenti/elaborato`.
- 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, ...).