

Il gioco elettronico `wumpus` consiste in una creatura all'interno di un mondo virtuale. Il mondo virtuale è costituito da $n \times m$ caselle con n e m numeri interi strettamente positivi. Le caselle sono numerate a partire da 0. La prima casella in alto a sinistra ha coordinate (0,0). La creatura si sposta nelle caselle del mondo virtuale. Le caselle possono contenere un numero maggiore o uguale a 0 di gettoni, che incrementano il punteggio della creatura quando si sposta sulla casella. Implementare le seguenti operazioni per il gioco `wumpus`:

- `wumpus p(n, m, i, j);`
Costruttore che crea un gioco `p`. Il mondo virtuale ha dimensione $n \times m$. La casella di coordinate (x,y) contiene $(x+y)$ gettoni. La creatura si posiziona sulla casella (i, j) e colleziona il numero di gettoni della casella. La casella (i,j) assume un numero di gettoni uguale a 0.
- `p.move(x, y);`
Operazione che muove la creatura nella casella di coordinate (x,y) . Sono ammessi solo spostamenti di una casella in orizzontale, o di una casella in verticale. La creatura non può uscire fuori dal mondo virtuale. Se la creatura cambia posizione, colleziona il numero di gettoni della casella di destinazione. Dopo questa operazione, la casella di destinazione avrà un numero di gettoni uguale a 0.
- `cout << p;`
Operatore di uscita del gioco.
L'uscita mostra i punti acquisiti della creatura. Inoltre l'uscita visualizza il mondo virtuale. In particolare il carattere 'C' indica che la casella è occupata dalla creatura (in questo caso il numero di gettoni è sempre 0), il carattere '-' indica una casella senza gettoni, il carattere '*' indica una casella che contiene gettoni. In questo esempio il mondo virtuale è 3×5 , la creatura è sulla casella (1, 0) ed ha acquisito 8 punti.

Creatura: 8

```
* * - * *
C - - * *
* * * * *
```

- `wumpus p1(p);`
Costruttore di copia che crea un gioco `p1` uguale a al gioco `p`.
- `p1 = p;`
Operatore di assegnamento, che sostituisce il valore del gioco risultato `p1` con quello del gioco `p`.
- `p % k;`
Operazione che restituisce un gioco con mondo virtuale di dimensione $k \times k$. Le righe (colonne) del mondo virtuale sono le prime k righe (colonne) di `p`. I gettoni delle caselle rimangono invariati rispetto a quelli del gioco `p`. La creatura è posizionata nella casella (0,0) ed assume un numero di gettoni uguale a 0.
- `~wumpus();`
Distruttore.

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

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA:

- Effettuare il **login**
Nome: studenti
Password: studenti
- Prima di iniziare a svolgere l'elaborato, eseguire **ident.bat** nella cartella **C:\esame** e inserire i dati richiesti
- Aprire il *Dev-C++* (dal Menù *Avvio* (o *Start*) nella barra degli strumenti in fondo allo schermo, selezionare *Programmi* e quindi *Dev-C++*)
- Dal menu *File* del *Dev-C++*, aprire il progetto *esainf.dev* presente nella cartella *c:\esame\esainf*. Il progetto contiene tre file, denominati *compito.h*, *compito.cpp* e *main.cpp*
- Scrivere la dichiarazione della classe nel file *compito.h* e la definizione delle funzioni nel file *compito.cpp*. Il file *main.cpp* contiene la funzione principale *main()* che serve a verificare le funzioni scritte nel file *compito.cpp*. Il file *main.cpp* può essere modificato. **Si tenga presente, comunque, che in sede di valutazione dell'elaborato verrà considerato esclusivamente il contenuto dei file *compito.h* e *compito.cpp***
- Per una corretta stampa dell'elaborato **mantenersi entro i margini imposti dall'ambiente Dev-C++** cioè entro la linea verticale presente alla destra della pagina;
- **Per la Consegna:**
 - 1) Eseguire lo script **esegui.bat** dalla cartella **C:\esame\esainf** e premere il tasto INVIO fino a quando non viene chiusa la finestra che è stata attivata.
 - 2) Aspettare al proprio posto di essere chiamati per verificare la stampa del proprio elaborato, firmarlo e **consegnarlo definitivamente**.

REGOLA PER LA CORREZIONE:

Per la correzione dell'elaborato è necessario che siano implementate correttamente sia la definizione della classe che le seguenti funzioni:

- - costruttore (**wumpus p(n, m, i, j)**)
- - move (**p.move(x, y)**)
- - stampa a video del gioco (**cout << p**)
siano implementate correttamente.

Il codice di altre funzioni che impedisce la compilazione e l'esecuzione del progetto deve essere messo come commento e verrà comunque valutato in sede di correzione dell'elaborato.
