Un `CampoMinato` rappresenta un'istanza dell'omonimo gioco (noto anche come "Mine" o "Prato fiorito"). Esso consiste in una griglia (chiamata tabellone) di `n×n` celle coperte, in corrispondenza delle quali sono state segretamente collocate delle mine. Compito del giocatore è quello di scoprire (o ripulire) tutte le celle lasciando coperte solo quelle che contengono le mine. Implementare le seguenti operazioni che possono essere effettuate su un `CampoMinato`:

--- Metodi invocati nella PRIMA PARTE di main.cpp: ---

✓ **`CampoMinato c(n);`**

Costruttore che inizializza un `CampoMinato` di dimensione `n×n`. Inizialmente non vi sono mine ed il gioco non può iniziare fintanto che non ne viene collocata almeno una. Se necessario, implementare anche il relativo distruttore.

✓ **`c.aggiungi_mina(r,c);`**

Funzione membro che aggiunge una mina nella cella individuata da indice di riga `r` e di colonna `c` (entrambe indicizzate da 0 a partire dall'elemento in alto a sinistra). L'aggiunta di mine è possibile solo a gioco non avviato. Se la mina viene collocata correttamente, il metodo restituisce `true`, altrimenti `false`.

✓ **`c.scopri(r,c);`**

Funzione membro che scopre la cella individuata da indice di riga `r` e di colonna `c` (secondo la medesima convenzione sugli indici fornita precedentemente). La prima (corretta) invocazione di questa funzione dà inizio al gioco, qualora possa iniziare. In caso di mancato inizio del gioco o scorretta invocazione della funzione, la struttura dati rimane inalterata. La presente funzione membro si comporta come segue:

1. Se la cella scoperta nascondeva una mina, viene rivelata sul tabellone la presenza della stessa (si veda l'operatore di uscita) ed il gioco termina immediatamente per sconfitta.
2. Altrimenti:
    2.1. Se almeno una delle celle adiacenti a quella scoperta contiene una mina, viene rivelato sul tabellone il numero di tali mine ed il gioco prosegue. Vengono considerate "adiacenti" le (eventuali) celle a destra, a sinistra, in basso, in alto, ma anche quelle in basso a destra, in basso a sinistra, ecc.
    2.2. Se la cella scoperta non è adiacente a nessuna mina, l'applicativo lascia la cella vuota.
3. Se la cella scoperta era l'ultima a non contenere una mina tra quelle non ancora scoperte, il gioco termina immediatamente per vittoria.

✓ **`cout << c;`**

Operatore di uscita per il tipo `CampoMinato`. Qualora il gioco debba ancora essere avviato ma non può essere fatto perché non è ancora stata collocata alcuna mina, l'operatore deve stampare a video "`Inserire una mina per avviare il gioco`", seguito da un accapo. Qualora il gioco sia terminato per sconfitta, l'operatore deve mostrare a video "`Game over`", seguito da un accapo. Qualora il gioco sia terminato per vittoria, l'operatore deve mostrare a video "`Vittoria!`", seguito da un accapo. In ogni altro caso, la stampa a video ha il seguente formato:

```
Campo Minato 3x3 - Mine da trovare: 3
X 2 X
X 2 1
X 1
```

Nell'esempio quello che viene mostrato, si ha un gioco 3x3 dove le "`X`" corrispondono a celle ancora da scoprire. Si noti che la cella in posizione (2,2) è "vuota" perché la sua scoperta è ricaduta nel caso 2.2 di cui sopra. Dallo stato del gioco si evince che le mine sono collocate in posizione (0,0), (0,2) e (2,0).

--- Metodi invocati nella SECONDA PARTE di main.cpp: ---

- ✓ **`c.scopri(r,c);`**

Modificare la funzione membro precedentemente introdotta in maniera tale che al punto 2.2 dell'elenco, oltre a lasciare la cella vuota, l'applicativo scopre a cascata anche tutte le celle adiacenti, incluse quelle diagonali, utilizzando la medesima logica descritta nella prima parte e qui.

- ✓ **`CampoMinato c2(c1);`**

Costruttore di copia che inizializza `c2` con un tabellone identico a quello di `c1`, fatto salvo che nessuna cella è scoperta e che il gioco non è da considerarsi avviato.

- ✓ **`c1 + c2;`**

Operatore di somma che restituisce un `CampoMinato` avente griglia di dimensione pari alla somma delle dimensioni ed inizializzato come segue:

1. Una copia del tabellone di `c1` è collocata in alto a sinistra.
2. Una copia tabellone di `c2` è collocata in basso a destra.
3. Le celle non ancora inizializzate sono vuote.

Il gioco risultato è da considerarsi non avviato.

Mediante il linguaggio C++, realizzare il tipo di dato astratto definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL come il tipo `string`, il tipo `vector`, il tipo `list`, ecc. **Gestire le eventuali situazioni di errore**.

USCITA CHE DEVE PRODURRE IL PROGRAMMA

```
--- PRIMA PARTE ---

Test del costruttore:
Inserire una mina per avviare il gioco

Test della aggiungi_mina:
Campo Minato 3x3 - Mine da trovare: 2
X X X
X X X
X X X

Test della scopri:
Campo Minato 3x3 - Mine da trovare: 2
X X X
X X X
X X 2

Campo Minato 3x3 - Mine da trovare: 2
X 1 X
X X X
X X 2

Game over


--- SECONDA PARTE ---

Test del costruttore di copia:
Campo Minato 3x3 - Mine da trovare: 2
X X X
X X X
X X X

Altri Test della scopri:
Campo Minato 3x3 - Mine da trovare: 2
  1 X
1 2 X
X X 2

Vittoria!

Test dell'operatore +:
Campo Minato 6x6 - Mine da trovare: 4
X X X X X X
X X X X X X
X X 2 X X X
X X X X X X
X X X X X X
X X X X X 2

Test del distruttore:
```

Note per la consegna:

Affinché l'elaborato venga considerato valido, il programma **deve** produrre almeno la prima parte dell'output atteso. In questo caso, i docenti procederanno alla valutazione dell'elaborato **solo se** lo studente avrà completato l'autocorrezione del proprio elaborato.

In **tutti** gli altri casi (per esempio, il programma non compila, non collega, non esegue o la prima parte dell'output non coincide con quella attesa), l'elaborato è considerato **insufficiente** e, pertanto, **non verrà corretto**.