

## Campi e alberi

### Alberi

Si consideri il gioco “Alberi”, costituito da una griglia quadrata, in cui in ogni elemento può esserci un albero o meno. Le dimensioni della griglia variano da gioco a gioco, ma sono di al più 12 elementi per lato. La griglia è occupata da aree di colore diverso, cui appartengono uno o più elementi, e le regole per la corretta presenza degli alberi sono le seguenti: i) ci devono essere 2 alberi per riga, ii) ci devono essere 2 alberi per colonna, iii) ci devono essere 2 alberi per colore, iv) negli elementi adiacenti ad un albero non ci devono essere alberi. Gli schemi più piccoli (con dimensione minore o uguale a 6) contengono 1 solo albero per riga, colonna e colore.

Si realizzi un programma che acquisisce in ingresso la specifica della disposizione delle aree e dei colori (nel formato indicato di seguito) e visualizza 1 se lo schema ricevuto in ingresso è corretto, 0 altrimenti, seguito dal carattere 'a capo'. Si organizzi il codice in sottoprogrammi.

Si suggerisce di sviluppare, tra gli altri, i seguenti sottoprogrammi:

**checkGriglia**

restituisce 1 se il numero di alberi e la loro disposizione è corretta

**checkRiga**

restituisce 1 se il numero di alberi nella riga della griglia è quello richiesto

**checkColonna**

restituisce 1 se il numero di alberi nella colonna della griglia è quello richiesto

**checkColore**

restituisce 1 se il numero di alberi in un'area di colore della griglia è quello richiesto

**checkDistanza**

restituisce 1 se quando si incontra un albero nella griglia esso non ha alberi negli elementi adiacenti

**Ingresso/Uscita:**

**input:** un insieme di numeri interi e caratteri

**output:** un intero

**Alcuni casi di test per il collaudo:**

5 % dimensione del campo di gioco

AABBB % un carattere per ogni colore

AAAAA

AAAAC

DDDEE

DDDEE

0 2 % coordinate degli alberi

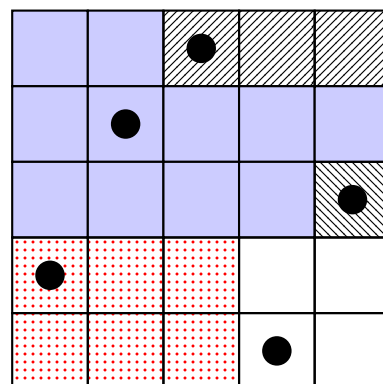
1 1

2 4

3 0

4 3

0



5 % dimensione del campo di gioco  
AABBB % un carattere per ogni colore  
AAAAA  
AAAAC  
DDDEE  
DDDEE  
0 2 % coordinate degli alberi  
1 0  
2 4  
3 1  
4 3  
1

