

Bustine di carte (carte)

Laura, Daria e Noemi sono grandi appassionate di carte da collezione. Finalmente sono riuscite a risparmiare abbastanza per comprare un po' di bustine e adesso devono decidere come distribuirle. Ognuna di loro ha infatti dei gusti particolari in tema di carte: Laura ama le carte laminate, Daria va invece pazza per i mostri d'aria e Noemi ha un debole per le carte con lo sfondo nero.



Le ragazze hanno comprato M carte, ogni carta può essere sia laminata, che contenere un mostro d'aria, che avere lo sfondo nero, oppure una qualsiasi combinazione di queste caratteristiche. Per non scontentare nessuno ciascuna ragazza prenderà quelle carte che possiedono solamente la qualità da loro prediletta. Vale a dire che Laura prenderà le carte laminate ma non d'aria né nere, Daria quelle d'aria ma non laminate né nere e Noemi quelle nere ma non laminate né d'aria.

Le ragazze vogliono capire quante carte avanzano dopo averle distribuite, in modo da poterle rivendere a Edoardo. Aiutale a calcolare il numero di carte avanzate!

Implementazione

Dovrai sottoporre un unico file, con estensione `.cpp`.

📖 Tra gli allegati a questo task troverai un template `carte.cpp` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

```
C++ | int scarta(int M, vector<bool> L, vector<bool> D, vector<bool> N);
```

- L'intero M rappresenta il numero di carte comprate.
- Il vettore L , indicizzato da 0 a $M - 1$, rappresenta quali carte sono laminate: $L[i]$ è `true` se la i -esima carta è laminata, `false` se non lo è.
- Il vettore D , indicizzato da 0 a $M - 1$, rappresenta quali carte sono d'aria: $D[i]$ è `true` se la i -esima carta è d'aria, `false` se non lo è.

- Il vettore N , indicizzato da 0 a $M - 1$, rappresenta quali carte sono nere: $N[i]$ è **true** se la i -esima carta è nera, **false** se non lo è.
- La funzione dovrà restituire il numero di carte avanzate dopo la distribuzione.

Il grader chiamerà la funzione `scarta` e ne stamperà il valore restituito sul file di output.

Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama le funzioni che dovete implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto da $M + 1$, contenenti:

- Riga 1: l'unico intero M .
- Riga $2 + i$: i tre valori $L[i]$, $D[i]$ e $N[i]$ separati da uno spazio.

Il file di output è composto da un'unica riga, contenente:

- Riga 1: il valore restituito dalla funzione `scarta`.

Assunzioni

- $1 \leq N \leq 100\,000$

Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test che lo compongono.

- **Subtask 1 [0 punti]**: Casi d'esempio.
- **Subtask 2 [20 punti]**: Non ci sono carte nere, vale a dire $N[i] = 0$ per ogni $i = 0, \dots, N - 1$.
- **Subtask 3 [80 punti]**: Nessuna limitazione aggiuntiva.

Esempi di input/output

stdin	stdout
4 1 1 1 1 0 0 0 0 0 0 1 0	2
8 0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 0 1 1 1	5

Spiegazione

Nel **primo caso di esempio** le ragazze hanno comprato 4 carte. Laura prende la seconda carta, che è solo laminata e Daria prende la quarta carta, che è solamente d'aria, avanzando così in totale 2 carte.

Nel **secondo caso di esempio** delle 8 carte totali Laura prende la quinta carta, Daria prende la terza e Noemi la seconda. Avanzano quindi 5 carte.