

Oro, argento e bronzo (medaglie)

Limite di tempo: 1.0 secondi
 Limite di memoria: 256 MiB

In uno spiacevole incidente, Giorgio e Gabriele hanno mescolato tutte le loro innumerevoli medaglie vinte in varie competizioni, e non riescono più a ricostruire quali erano le loro in partenza. Decidono quindi di spartirselo secondo un gioco, detto *oro, argento e bronzo*.

Impilano quindi tutte le N medaglie, ciascuna delle quali può essere d'oro, d'argento o di bronzo. Giorgio, per maggiore anzianità, inizia il gioco e prende una, due o tre medaglie (a sua scelta) dall'inizio della pila. A seguire, Gabriele fa lo stesso e i turni si susseguono alternati finché tutte le medaglie sono state prese da uno dei due.

Giorgio, di recente in difficoltà economiche, ha il segreto piano di rivendere tutte le medaglie che riesce a raccogliere; e sa che potrà ottenere 500 euro per ogni medaglia d'oro, 300 euro per ogni medaglia d'argento e 100 euro per ogni medaglia di bronzo. Sapendo che sia Giorgio che Gabriele giocano nel modo ottimale¹, quanti euro al massimo potrà ottenere Giorgio?

Implementazione

Dovrai sottoporre esattamente un file con estensione `.c`, `.cpp` o `.pas`.

📖 Tra gli allegati a questo task troverai un template (`medaglie.c`, `medaglie.cpp`, `medaglie.pas`) con un esempio di implementazione da completare.

Se sceglierai di utilizzare il template, dovrai implementare la seguente funzione:

C/C++	<code>int gioca(int N, short M[]);</code>
Pascal	<code>function gioca(N: longint; var M: array of integer): longint;</code>

In cui:

- L'intero N rappresenta il numero di medaglie nella pila all'inizio.
- L'array M , indicizzato da 0 a $N - 1$, contiene la descrizione della pila di medaglie, in cui all'indice 0 è indicata la medaglia più alla base, mentre all'indice $N - 1$ è indicata la medaglia più in cima. Precisamente, $M[i]$ vale 0 per indicare una medaglia di bronzo, 1 per indicare una medaglia d'argento e 2 per indicare una medaglia d'oro.
- La funzione dovrà restituire il massimo numero di euro che Giorgio può ottenere dalla vendita delle medaglie guadagnate, che verrà stampato sul file di output. Si consideri che sia Giorgio che Gabriele giocano in maniera ottimale per ottenere a fine partita il massimo valore in euro possibile.

Dati di input

Il file `input.txt` è composto da due righe. La prima riga contiene l'unico intero N . La seconda riga contiene gli N interi M_i separati da uno spazio.

¹In un gioco a informazione perfetta (tutti i giocatori conoscono il complessivo stato del gioco) e deterministico (non ci sono elementi casuali), una strategia S è *ottimale* se in ogni configurazione del gioco ti fa scegliere la mossa con il *risultato garantito* più alto. Il risultato garantito di una mossa è il minimo risultato che si può ottenere a fine partita seguendo la strategia S , al variare della strategia T dell'avversario.

Dati di output

Il file `output.txt` è composto da un'unica riga contenente un unico intero, la risposta a questo problema.

Assunzioni

- $1 \leq N \leq 100\,000$.
- $0 \leq M_i \leq 2$ per ogni $i = 0 \dots N - 1$.

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 relativi ad esso.

- **Subtask 1 [10 punti]:** Casi d'esempio.
- **Subtask 2 [20 punti]:** $N \leq 10$.
- **Subtask 3 [40 punti]:** $N \leq 100$.
- **Subtask 4 [30 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
5 2 1 0 1 0	600
input.txt	output.txt
10 1 0 0 0 2 1 0 1 1 2	1600
input.txt	output.txt
10 1 0 0 0 2 1 0 1 0 2	1500

Spiegazione

Nel **primo caso di esempio**, a Giorgio conviene prendere solo la prima medaglia (del valore di 100 euro). A questo punto Gabriele non ha scelta che prendere le tre medaglie seguenti (del valore complessivo di 700 euro), perché qualunque mossa egli faccia sa che Giorgio prenderebbe tutto quello che resta.

Nel **secondo caso di esempio**, a Giorgio conviene prendere le prime due medaglie (del valore di 800 euro), a Gabriele dunque conviene prenderne tre (del valore di 700 euro), Giorgio quindi ne prende una (del valore di 500 euro), Gabriele tre (del valore di 300 euro) e Giorgio infine ancora una (del valore di 300 euro), per un totale di 1600 euro per Giorgio.

Nel **terzo caso di esempio**, a Giorgio conviene prendere la prima medaglia (del valore di 500 euro), a Gabriele dunque conviene prenderne una (del valore di 100 euro), e da qui la partita procede analogamente al precedente esempio ma a parti invertite, e quindi Giorgio raccoglie altri 1000 euro per un totale di 1500 euro per Giorgio.