



Forza Mediana! (ora in Versione Interattiva) (mediana_x)

Descrizione del problema

Ti viene fornito un set di n biglie numerate da 0 ad $n-1$, tutte di peso diverso. Devi scrivere una procedura che riordini le biglie o per peso strettamente crescente oppure per peso strettamente decrescente: la sola cosa importante è che, comunque prese 3 biglie consecutive nel tuo ordinamento, quella in posizione intermedia sia anche di peso intermedio. Il file sorgente da sottomettere deve avere la seguente struttura:

```
#include "ourLibToPlay.h"
```

```
void ordina(long int n) {  
    ...  
}
```

ma può contenere anche altre funzioni di servizio per la funzione `ordina` che sei chiamato a realizzare. Il parametro n che viene passato alla funzione `ordina` da te realizzata è il numero di biglie sotto esame. Le n biglie differiscono tutte per peso. Per riordinarle potrai invocare la seguente funzione:

```
long int bigliaIntermedia(long int bigliaA, long int bigliaB, long int bigliaC)
```

la quale, ricevute in input tre biglie qualsiasi, ti restituisce quella di peso intermedio tra le tre.

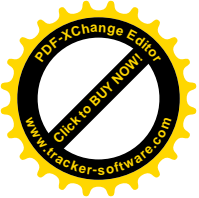
Una volta individuato un corretto ordinamento delle biglie, devi consegnarlo invocando la funzione:

```
void consegnaBiglieInOrdine( long int biglia_in_pos[] )
```

L'ordinamento consegnato viene considerato valido solo se esso è strettamente crescente oppure strettamente decrescente. Tuttavia, ogni chiamata alla funzione `bigliaIntermedia` ti costa una moneta e, oltre al tempo di calcolo, devi stare attento a non eccedere l'eventuale budget, ove previsto dal subtask.

Subtask

- **Subtask 1 [0 punti]:** le biglie che ti vengono affidate sono ordinate per peso crescente come nella loro numerazione da 0 a $n-1$.
- **Subtask 2 [0 punti]:** le biglie che ti vengono affidate sono ordinate per peso decrescente come nella loro numerazione da 0 a $n-1$.
- **Subtask 3 [0 punti]:** $n \leq 3$.
- **Subtask 4 [10 punti]:** $n \leq 30$.
- **Subtask 5 [20 punti]:** sono consentite al più $n(n-1)/2$ pesate, $n \leq 1000$.
- **Subtask 6 [35 punti]:** sono consentite al più $3n \log_2 n$ pesate, $n \leq 1000$.



- **Subtask 7 [35 punti]:** sono consentite al più $n + n(\lceil \log_2 n \rceil)$ pesate, $n \leq 100,000$.

Assunzioni

- $3 \leq n \leq 100\,000$.
- tempo limite: 1 secondo.
- Il programma termina dopo la prima chiamata alla funzione `consegnaBiglieInOrdine` oppure allo scadere del tempo limite.

Cosa deve contenere il File da Sottomettere, e come viene gestito dal server

L'estensione del file che sottometti, *.c, *.cpp oppure *.pas, determina se esso viene compilato col compilatore del C (gcc), del c++ (g++) oppure del PASCAL (fpc). Le opzioni di compilazione, specifiche al linguaggio, sono riportate nel dettaglio sulla pagina Descrizione del problema in modo che tu possa simulare esattamente in locale cosa succede sul server. I comandi di compilazione riportati assumono che il nome del file sottomesso (privato di estensione) coincida col nome del problema.

Nel caso di un problema interattivo come questo, dove il tuo programma interagisce con altri da noi predisposti (il grader ed anche un header da includere), i comandi di compilazione riportati compilano ed assemblano i vari pezzi (il tuo programma, il grader, e l'header). Se vuoi testare il comportamento del tutto in locale, devi allora scaricarti il nostro grader e l'header che rendiamo disponibile tra gli allegati alla pagina del problema.

Tra gli allegati potremmo inoltre fornire una soluzione di esempio che, pur essendo compilabile sia in locale che dal server (ove ad esso sottomessa), faccia pochi o nessun punto in quanto non significativa per il problema in questione.