

Verifiche (verifiche)

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

Il professor Tarboto sta pensando come gestire le verifiche nel prossimo anno scolastico. Dopo un'accurata riflessione, ha deciso che opererà nel modo seguente: organizzerà N verifiche lungo l'anno e in ciascuna di queste chiederà esattamente K argomenti.

I suoi furbi studenti sono venuti in possesso della pianificazione e conoscono esattamente in quali giorni il professore ha pianificato le verifiche. Tali giorni sono numerati a partire da 0, che equivale al primo giorno di scuola.

Sfortunatamente, la memoria degli studenti è limitata e il loro carico di lavoro deve essere umano: questo significa che in un giorno uno studente può studiare un solo argomento, che conoscerà quindi a partire dal giorno seguente. Inoltre si ricorderà tale argomento, indipendentemente da quale sia, per T giorni. Ad esempio: se uno studente studia un argomento il giorno 2, lo conoscerà dal giorno 3 al giorno $2 + T$ (incluso).

Gli studenti possono studiare anche prima dell'inizio dell'anno scolastico, se necessario, con le stesse modalità esposte sopra.

Per ottimizzare la pianificazione aiuta gli studenti a capire qual è il numero minimo di giorni in cui devono studiare, affinché in occasione di *tutte* le verifiche lungo l'anno sappiano *almeno* K argomenti!

Dati di input

Il file `input.txt` è composto da 2 righe. La prima riga contiene tre interi N , T , K separati da spazio. Essi sono rispettivamente: il numero di verifiche, il numero di giorni per cui uno studente memorizza un argomento e il numero di argomenti richiesti ad ogni verifica. La seconda riga contiene N interi, i giorni (a partire da 0) in cui saranno svolte le verifiche.

Dati di output

Il file `output.txt` contiene un singolo intero: il numero minimo di giorni in cui uno studente deve studiare per arrivare preparato alle verifiche. Se non è possibile rispettare le condizioni del problema, stampare -1.

Assunzioni

- $1 \leq N, T, K \leq 1000$.
- Un anno scolastico ha al più 2000 giorni, quindi i giorni di svolgimento delle verifiche sono compresi tra 0 e 1999.
- I giorni di svolgimento delle verifiche sono forniti in ordine cronologico.

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 [40 punti]:** $N = 1$, ovvero c'è solo una verifica in tutto l'anno.
- **Subtask 3 [50 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
1 8 3 10	3
1 1 3 10	-1

Ferrovia ad alta velocità (altavelocita)

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

Negli ultimi anni in Italia sono stati costruiti diversi tratti di ferrovie ad alta velocità: ad esempio Milano - Roma è ora percorribile in meno di tre ore grazie a questo nuovo tipo di ferrovia, che deve essere realizzata appositamente seguendo particolari criteri.

Durante i lunghi viaggi in treno per andare in università, Luca si è appassionato alla materia osservando dal finestrino la realizzazione della nuova ferrovia ad alta velocità Milano - Brescia che sarà completata entro dicembre 2016. Ha notato che sono stati completati sinora N lotti (a singolo binario), ciascuno lungo esattamente dalla progressiva chilometrica L_i a quella L_f .

Essendo la ferrovia a più binari, alcune tratte sono riportate più volte: ciò significa che lì sono stati già completati più binari (che hanno chiaramente uguale progressiva chilometrica). Il capolinea, Milano, si considera a progressiva chilometrica 0.

Com'è noto, in Italia le opere pubbliche nascondono spesso malaffare: per questo il presidente dell'autorità nazionale anticorruzione ha deciso che effettuerà un sopralluogo a campione, su uno dei tratti già realizzati, per controllare che tutto sia in regola.

Il presidente ha scelto di fare un sopralluogo al K -esimo chilometro di binario che è stato posato a partire dal capolinea iniziale. Aiuta i tecnici della ferrovia a capire a quale progressiva chilometrica si trova tale K -esimo chilometro.

☞ Attenzione: K supera, in alcuni casi, 2^{32} . Se si lavora, ad esempio, in C/C++ è quindi richiesto l'uso del tipo `long long` al posto di `int`.

Dati di input

Il file `input.txt` è composto da $N + 1$ righe. La prima riga contiene due interi N e K , rispettivamente il numero di lotti già realizzati e il chilometro di binario posato su cui deve compiersi l'ispezione.

Le seguenti N righe contengono ciascuna due interi, la progressiva chilometrica a cui ha inizio il lotto, L_i , e quella in cui ha fine, L_f .

Dati di output

Il file `output.txt` contiene un singolo intero: la progressiva chilometrica a cui si trova il K -esimo chilometro di binario che è stato posato, partendo dal capolinea.

Assunzioni

- $1 \leq N \leq 1000$.
- La massima progressiva chilometrica è 2^{31} .
- Per ogni lotto vale $L_i \leq L_f$, ad indicare che è stata completata la realizzazione di un binario dalla progressiva chilometrica L_i a L_f (estremi inclusi).
- Il valore K è un intero positivo che corrisponde a un chilometro di binario posato. In altre parole, K non supera mai il totale complessivo di chilometri di binari posati.

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.

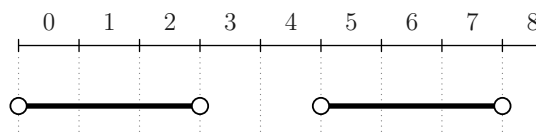
- **Subtask 1 [10 punti]:** Casi d'esempio.
- **Subtask 2 [25 punti]:** La massima progressiva chilometrica è 1 000.
- **Subtask 3 [15 punti]:** La massima progressiva chilometrica è 40 000.
- **Subtask 4 [15 punti]:** La massima progressiva chilometrica è 120 000.
- **Subtask 5 [35 punti]:** Nessuna limitazione specifica.

Esempi di input/output

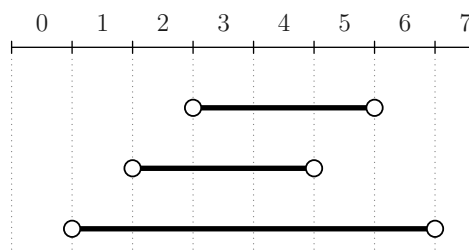
input.txt	output.txt
2 4 0 2 5 7	5
3 7 3 5 2 4 1 6	4

Spiegazione

Il **primo caso di esempio** è rappresentato dal seguente schema. C'è quindi un unico binario e il quarto chilometro posato si trova alla progressiva chilometrica 5.



Il **secondo caso di esempio** è rappresentato dal seguente schema:



Ci sono più binari: partendo da Milano (progressiva 0) incontriamo un chilometro di binari alla progressiva 1, due chilometri alla progressiva 2, tre chilometri alla progressiva 3. Il settimo chilometro di binario posato si trova quindi alla progressiva chilometrica 4.

Edifici universitari (edifici)

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

Luca ha iniziato a studiare in università. La struttura del grande campus universitario è particolarmente complessa. Esso è infatti costituito da N edifici dove vengono svolte diverse attività. In particolare K di questi N edifici contengono aule di didattica e aule studio, mentre i rimanenti $N - K$ contengono alloggi per gli studenti.

Gli edifici sono tra loro connessi grazie a M strade. Ciascuna ha una lunghezza, espressa in metri, e collega due edifici. Tutte le strade possono essere percorse in entrambi i sensi.

Dopo qualche mese in cui Luca si è spostato dalla sua città natale all'università in treno, ha deciso che è giunto il momento di affittare un alloggio in uno degli edifici dell'università. In tal modo sarà più vicino all'università e potrà alzarsi più tardi la mattina.

Lungo una giornata tipica, Luca va a trovare i suoi amici che sono dislocati a gruppetti in tutti i K edifici che contengono aule studio. La mattina Luca esce, fa il giro degli edifici in cui ci sono i suoi amici e al termine fa rientro nella sua stanza.

In quale edificio (adibito ad alloggi) conviene a Luca affittare una stanza per minimizzare la distanza percorsa ogni giorno, sapendo che può andare a visitare gli amici nell'ordine che vuole?

Dati di input

Il file `input.txt` è composto da $2 + M$ righe. La prima riga contiene tre interi N , M , K separati da spazio. Essi sono rispettivamente: il numero di edifici, il numero di strade e il numero di edifici contenenti aule studio. La seconda riga contiene K interi, i numeri identificativi degli edifici contenenti aule studio. Seguono M righe che descrivono le strade: ognuna contiene tre numeri A , B , L ad indicare che esiste una strada tra gli edifici A e B di lunghezza L .

Dati di output

Il file `output.txt` contiene un singolo intero: la distanza minima che Luca deve percorrere in un giorno, facendo la miglior scelta per l'affitto della stanza.

Assunzioni

- $2 \leq N \leq 5\,000$.
- $1 \leq M \leq 100\,000$.
- $1 \leq K \leq 6$.
- $K < N$.
- Una stessa strada viene indicata al massimo una volta nel file di input.
- Presi due edifici qualunque, esiste sempre almeno un percorso che li collega.
- $1 \leq A_i, B_i \leq N$ per ogni $i = 0 \dots N - 1$.
- $1 \leq L_i \leq 1\,000$ per ogni $i = 0 \dots N - 1$.
- Il percorso giornaliero può passare anche più di una volta lungo una stessa strada.

Assegnazione del punteggio

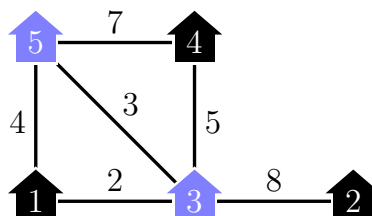
Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test relativi ad esso.

- **Subtask 1 [5 punti]:** Casi d'esempio.
- **Subtask 2 [30 punti]:** $K = 1$, ovvero Luca parte dalla sua stanza per raggiungere l'unico gruppo di amici e torna subito indietro.
- **Subtask 3 [15 punti]:** $K = 2$.
- **Subtask 4 [30 punti]:** $N \leq 500$.
- **Subtask 5 [20 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
<pre>4 5 1 3 2 4 5 2 1 2 1 4 3 4 3 1 3 1 2</pre>	2
<pre>5 6 2 3 5 3 1 2 1 5 4 3 5 3 2 3 8 5 4 7 4 3 5</pre>	9

Spiegazione



Nel **secondo caso di esempio**, rappresentato in figura, una scelta ottimale è affittare una stanza nell'edificio 1. A quel punto il percorso giornaliero di Luca consiste nel partire dall'edificio 1, percorrere una strada di lunghezza 2 per arrivare all'edificio 3, quindi una strada di lunghezza 3 per arrivare all'edificio 5, infine una strada di lunghezza 4 per tornare al proprio alloggio.

Esame di maturità (esame)

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

Anche quest'anno è giunta l'ora degli esami di maturità e il presidente della commissione della classe di Tommaso ha già stabilito regole severe per lo svolgimento della prova orale, alla presenza dei vari docenti.

Il presidente ha deciso che la disposizione dei tavoli, dietro a cui siederanno i docenti, sarà a forma di cerchio (con il candidato al centro). Ha comunicato inoltre che intende adottare la seguente prassi:

- Ciascun docente ha diritto a porre un'unica domanda, caratterizzata da una certa difficoltà D .
- Un docente non può porre la sua domanda se uno dei suoi due vicini è già intervenuto per la sua domanda. Chiaramente, nella disposizione a cerchio consideriamo vicini le due persone sedute (alla propria destra e alla propria sinistra).

Sapendo che la difficoltà complessiva di un esame è data dalla somma delle difficoltà delle domande poste, qual è la *massima* difficoltà ottenibile rispettando quanto chiesto dal presidente?

Dati di input

Il file `input.txt` è composto da 2 righe. La prima riga contiene un intero N , il numero dei docenti. La seconda riga contiene N interi, ciascuno rappresentante la difficoltà della domanda dell' i -esimo docente.

Dati di output

Il file `output.txt` contiene un singolo intero: la massima difficoltà ottenibile per un esame, rispettando le richieste del presidente.

Assunzioni

- $2 \leq N \leq 1\,000$.
- $1 \leq D_i \leq 100$ per ogni $i = 0 \dots N - 1$.
- Essendo la disposizione a cerchio, il docente numero 0 si considera seduto a fianco del docente $N - 1$ (e viceversa).

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 [50 punti]:** $N \leq 20$.
- **Subtask 3 [40 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
6 8 1 4 5 6 9	18
5 1 1 1 1 1	2

Spiegazione

Nel **primo caso di esempio** la massima difficoltà è ottenuta dalle domande dei docenti 0, 2 e 4 (per un totale di $8 + 4 + 6 = 18$). Si osservi che $8 + 5 + 9 = 22$ porta a un totale più alto ma non è una risposta lecita, perché le domande di difficoltà 8 e 9 arrivano da due docenti seduti vicini.

Nel **secondo caso di esempio** ci sono diverse combinazioni di domande che danno luogo alla massima difficoltà ma non c'è un modo di sceglierle, rispettando le regole, la cui somma superi 2.