Spesa lampo (spesa)

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

Gabriele ha appena finito la sua ultima consulenza, e si prepara ad andare a casa della nonna per un meritato pranzo. Date le precedenti esperienze, decide però che è saggio prima passare da un supermercato a procurarsi un po' di bicarbonato, nel caso in cui i suoi propositi di mangiare il meno possibile falliscano miseramente dopo la dodicesima fetta di tiramisù.

Per non fare aspettare la nonna, questa deviazione deve costare il minor tempo possibile. Per questo Gabriele si è procurato la mappa dei supermercati della città, composta da N punti di interesse (incroci, piazze, edifici importanti) e M vie (a doppio senso di marcia) che collegano i punti di interesse. In particolare, K di questi punti rappresentano dei supermercati, mentre il punto di interesse indicato col numero 1 rappresenta il palazzo della *SteamPower S.P.A.*, dove si trova Gabriele in questo momento, e il punto numerato N è il condominio dove abita la nonna.

Data la mappa sopra descritta, Gabriele si chiede quanto tempo ci metterà (al minimo) per raggiungere la nonna, dovendo prima passare da un supermercato, e sapendo che ci vuole 1 minuto a percorrere ognuna delle M strade.

Implementazione

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

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

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

C/C++	`int compra(int N, int M, int K, int supermercati[], int da[], int a[]);`
Pascal	`function compra(N, M, K: longint; var supermercati, da, a: array of longint): longint;`

In cui:

- L'intero N rappresenta il numero di punti di interesse della città (numerati da 1 a N).
- L'intero M rappresenta il numero di strade (bidirezionali) che collegano i punti di interesse.
- L'intero K rappresenta il numero di supermercati presenti.
- L'array `supermercati`, indicizzato da 0 a $K-1$, contiene i numeri dei punti di interesse che corrispondono ai supermercati della città.
- Gli array `da` e `a`, indicizzati da 0 a $M-1$, contengono gli estremi delle strade, cioè i numeri dei punti di interesse che queste strade collegano.
- La funzione dovrà restituire il minimo numero di minuti necessari per raggiungere un supermercato e poi andare a casa della nonna, che verrà stampato sul file di output.

Dati di input

Il file `input.txt` è composto da $M+2$ righe. La prima riga contiene gli interi N, M, K. La seconda riga contiene i K interi `supermercati[`i`]` separati da uno spazio. Le successive M righe contengono le descrizioni delle strade: la i-esima di queste righe contiene i due interi `da[`i`]` e `a[`i`]`.

Dati di output

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

Assunzioni

- $3 \leq N \leq 10\,000$.
- $2 \leq M \leq 100\,000$.
- $1 \leq K \leq N - 2$.
- $1 < \texttt{supermercati}_i < N$ per ogni $i = 0 \ldots K - 1$.
- Nei punti 1 ed N non sono presenti supermercati, e i valori `supermercati[`i`]` non sono ripetuti.
- Tutte le strade sono bidirezionali e non sono duplicate nell'input.
- È possibile da ogni punto raggiungere ogni altro punto della città.
- Gabriele può percorrere la stessa strada più di una volta, o passare più volte da uno stesso punto.
- Gabriele impiega 1 minuto a percorrere ogni strada.
- Il tempo speso all'interno del supermercato è trascurabile.

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**]: $N, K \leq 100$.
- **Subtask 3** [**40 punti**]: $K \leq 10$.
- **Subtask 4** [**30 punti**]: Nessuna limitazione specifica.

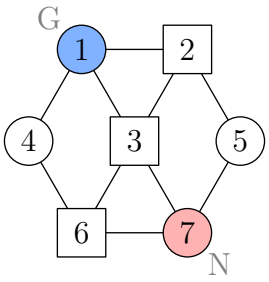
Esempi di input/output

input.txt	output.txt
7 10 3 3 6 2 1 2 2 5 7 3 4 6 3 6 6 7 7 5 1 4 3 1 2 3	2

input.txt	output.txt
9 13 3 3 6 5 2 5 9 2 7 1 6 3 9 3 1 8 2 6 6 5 7 9 2 8 4 7 4 5 8 9	4

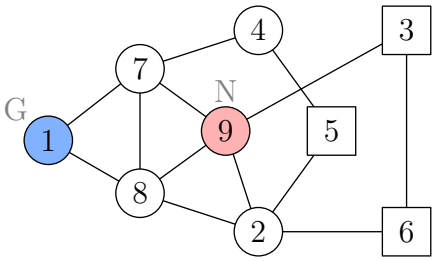
Spiegazione

Il **primo caso di esempio** corrisponde alla mappa



dove i nodi quadrati rappresentano i supermercati. A Gabriele conviene andare dal punto 1 al punto 3, comprare il bicarbonato, e poi ripartire, raggiungendo la casa della nonna al punto 7.

Il **secondo caso di esempio** corrisponde alla mappa



A Gabriele conviene andare dal punto 1 al punto 3 (in 3 minuti), e poi dal 3 tornare a casa della nonna, al nodo 9.