

## Inviti alla festa (festa)

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

Gabriele e Giorgio hanno deciso di organizzare una festa. Per questo hanno raccolto una lista di tutti i loro amici, e ora vogliono invitarli. Affinché gli amici di Gabriele e Giorgio non si annoino alla festa, è necessario che ogni invitato conosca almeno altri due degli amici invitati.

Conoscendo il grafo delle conoscenze reciproche tra gli amici di Giorgio e Gabriele, stabilisci qual è il massimo numero di persone che Gabriele e Giorgio possono invitare alla festa.

## Implementazione

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

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

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

C/C++	<code>int invita(int N, int M, int conoscenzeA[], int conoscenzeB[]);</code>
Pascal	<code>function invita(N, M: longint; var conoscenzeA, conoscenzeB: array of longint): longint;</code>

In cui:

- L'intero  $N$  rappresenta il numero di amici di Gabriele e Giorgio.
- L'intero  $M$  rappresenta il numero di archi nel grafo delle conoscenze (coppie di amici che si conoscono).
- Gli array `conoscenzeA` e `conoscenzeB`, indicizzati da  $0$  a  $M - 1$ , contengono le informazioni sugli archi del grafo: per ogni  $0 \leq i < M$ , gli amici `conoscenzeA[i]` e `conoscenzeB[i]` si conoscono. Gli amici sono indicizzati con valori  $0, 1, \dots, N - 1$ .
- La funzione dovrà restituire il massimo numero di persone che è possibile invitare rispettando la condizione che ogni invitato conosca almeno altri due invitati. Tale valore verrà stampato sul file di output.

## Dati di input

Il file `input.txt` è composto da  $M + 1$  righe. La prima riga contiene gli interi  $N$  e  $M$  separati da uno spazio. Le righe successive contengono la descrizione degli archi del grafo delle conoscenze: sulla  $i$ -esima di queste righe sono presenti gli interi `conoscenzeA[i]` e `conoscenzeB[i]`, separati da uno spazio.

## Dati di output

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

## Assunzioni

- $1 \leq N \leq 10\,000$ .
- $0 \leq M \leq 100\,000$ .
- Il grafo delle conoscenze è un grafo non diretto (cioè la conoscenza è sempre reciproca). Tutti gli archi sono validi ( $0 \leq \text{conoscenzeA}[i], \text{conoscenzeB}[i] \leq N-1$  e  $\text{conoscenzeA}[i] \neq \text{conoscenzeB}[i]$ ), e non vengono ripetuti.
- Gabriele e Giorgio non vanno contati nel numero di amici conosciuti dagli invitati.
- Nel caso in cui non sia possibile invitare alcun amico, stampare il valore 0.

## 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.

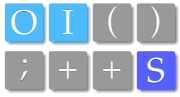
- **Subtask 1 [10 punti]:** Casi d'esempio.
- **Subtask 2 [20 punti]:**  $N \leq 10$ .
- **Subtask 3 [40 punti]:**  $N \leq 1000$ .
- **Subtask 4 [30 punti]:** Nessuna limitazione specifica.

## Esempi di input/output

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

input.txt	output.txt
3 2 0 2 1 2	0

input.txt	output.txt
9 10 0 1 2 0 5 2 4 5 3 4 2 3 7 2 6 5 0 5 1 8	5



## Spiegazione

Nel **primo caso di esempio** il massimo numero di invitati si raggiunge invitando alla festa gli amici 3, 4 e 5.

Nel **secondo caso di esempio** non è possibile invitare alcun amico alla festa.

Nel **terzo caso di esempio** il massimo numero di invitati si raggiunge invitando alla festa gli amici 3, 4, 2, 0, 5.