

CEOI 2026



Giorno 2
(italiano)

Day 2
(Italian)

Taglio dei Fiori

Limite di tempo: 4 s Limite di memoria: 256 MiB

Nel giardino comunitario delle CEOI, stiamo coltivando una collezione di fiori speciali che intrecciano strettamente le loro radici. Se li tagliamo, ricrescono, a patto di non tagliare in modo troppo aggressivo causando danni irreparabili.

Se una coppia di fiori, a e b , intreccia le proprie radici, i fiori vengono definiti "connessi". In caso contrario, sono "disconnessi". Le radici crescono secondo la seguente regola: siano a e b due fiori disconnessi. Se esistono almeno altri 2 fiori c e d tali che sia a che b siano connessi sia con c che con d , allora le radici tra a e b cresceranno, ed essi diventeranno connessi.

Questi fiori sono ormai in crescita da un po' di tempo e tutte le radici che potevano crescere seguendo la regola sopra descritta sono cresciute. In altre parole, se due fiori a e b sono entrambi connessi con due fiori c e d , allora a e b sono garantiti essere connessi tra loro.

Dobbiamo sradicare questo giardino e trasferirlo nella prossima sede delle CEOI. Per semplificare la migrazione, vorremmo tagliare il maggior numero possibile di radici. Tuttavia, vogliamo che i fiori ricrescano fino a tornare al loro stato attuale. Qual è il numero massimo di coppie di fiori connessi che possiamo tagliare in modo che possano comunque ricrescere fino al loro stato attuale? Non importa quanti cicli di crescita saranno necessari.

Input

La prima riga dell'input contiene due interi separati da uno spazio, n e m , rispettivamente il numero di fiori e il numero di connessioni esistenti tra di essi. Seguono m righe, ciascuna contenente una coppia di interi a_i e b_i , che indicano che i fiori a_i e b_i sono connessi. I fiori sono contrassegnati con numeri interi da $1 \dots n$. L'input è garantito avere la proprietà descritta nella traccia del problema.

Assunzioni

- $1 \leq n \leq 1000$
- $1 \leq m \leq 10^5$

Subtask

- Subtask 1 (20 punti): $n \leq 10$ e $m \leq 20$
- Subtask 2 (14 punti): $m = \frac{n \cdot (n-1)}{2}$
- Subtask 3 (15 punti): Si garantisce che ogni fiore è connesso al massimo con altri 7 fiori.
- Subtask 4 (15 punti): $n \leq 50$ e $m \leq 1000$
- Subtask 5 (36 punti): Nessuna limitazione aggiuntiva.

Output

Stampare un singolo intero — il numero massimo di connessioni che possiamo tagliare.

Esempio

Input

```

9 14
1 2
1 4
1 5
2 4
2 5
3 4
4 5
3 6
4 6
6 7
6 9
7 9
8 9
5 8

```

Output

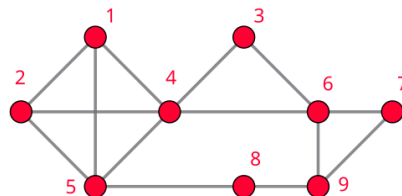
```

2

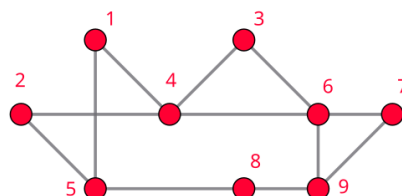
```

Commento

I fiori nell'esempio prima di qualsiasi taglio. È possibile verificare che non si possano formare ulteriori connessioni in base alla procedura di crescita descritta.



I fiori nell'esempio dopo il taglio hanno 2 connessioni in meno. La connessione tra 1 e 2 può ricrescere perché sia il fiore 1 che il fiore 2 sono connessi ai fiori 4 e 5. Analogamente, la connessione tra 4 e 5 può ricrescere perché sia il fiore 4 che il fiore 5 sono connessi ai fiori 1 e 2.



Torri

Limite di tempo: 1 s Limite di memoria: 256 MiB

Hai n computer e m torri, tutti situati in posizioni distinte lungo una linea. Devi accoppiare i computer utilizzando dei cavi in modo tale che ogni cavo parta da un computer, visiti alcune torri e termini in un altro computer. Il cavo può visitare qualunque torre (non solo quelle comprese tra i due computer) in un ordine arbitrario. Può saltare delle torri passandoci accanto senza visitarle. Può anche non visitare alcuna torre e connettere direttamente i due computer, ma non può connettere un computer a se stesso. Il numero di computer è pari.

Siano a e b le posizioni di due computer e siano $x_1, \dots, x_k$ le posizioni delle torri visitate dal cavo. La lunghezza del cavo è $|a - x_1| + |x_1 - x_2| + \dots + |x_{k-1} - x_k| + |x_k - b|$. Per un determinato cavo, definiamo il suo punteggio come $f \cdot u - l$, dove l è la sua lunghezza, f è una costante fissa e u è il numero di torri uniche che il cavo visita tra $x_1, \dots, x_k$. Più cavi possono visitare la stessa torre, e quella torre contribuirà al punteggio di ciascuno di quei cavi.

Devi calcolare la massima somma possibile dei punteggi dei cavi per accoppiare tutti i computer (cioè ogni computer deve appartenere a esattamente una coppia o, equivalentemente, deve essere collegato a esattamente un cavo).

Input

La prima riga contiene il numero di casi di test, T . I casi di test seguono uno dopo l'altro. Ogni caso di test è composto da tre righe. La prima riga contiene tre interi n , m e f — il numero di computer, il numero di torri e la costante f . La seconda riga contiene n interi $a_1, a_2, \dots, a_n$ — le posizioni dei computer. La terza riga contiene m interi $b_1, b_2, \dots, b_m$ — le posizioni delle torri.

Assunzioni

Siano N e M rispettivamente la somma di n e m su tutti i casi di test.

- $1 \leq T \leq 10^4$
- $1 \leq N, M \leq 2 \cdot 10^5$
- $0 \leq f \leq 10^9$
- n è pari
- $1 \leq a_i, b_i \leq 10^9$
- Tutte le posizioni dei computer e delle torri sono distinte (all'interno di un singolo caso di test).

Subtask

- Subtask 1 (5 punti): $N \leq 5000$, $m = 1$
- Subtask 2 (10 punti): $T \leq 20$, $n \leq 10$, $m \leq 100$

- Subtask 3 (27 punti): $N, M \leq 5000$
- Subtask 4 (21 punti): $N \leq 5000$
- Subtask 5 (37 punti): Nessuna limitazione aggiuntiva.

Output

Stampa T interi, ciascuno sulla propria riga — la massima somma possibile dei punteggi dei cavi per ogni caso di test.

Esempio

Input

```
4
2 1 100
1 10
11
4 1 10
2 4 6 8
20
4 1 10
2 4 6 8
5
6 3 10
2 13 4 8 6 10
5 1 9
```

Output

```
89
-4
12
51
```

Vim

Limite di tempo: 1 s Limite di memoria: 256 MiB

Sarebbe difficile trovare qualcosa di più *geek* del vecchio e classico Vim — un editor in cui possiamo fare cose che negli editor "normali" potremmo solo sognare. Questo, ovviamente, se siamo disposti a investire una settimana di apprendimento e un mese di pratica affinché i comandi insoliti ma potenti che offre diventino parte della nostra "memoria muscolare". Ebbene, proprio come in altri editor, in Vim abbiamo un *cursore*, che si trova sempre su uno dei caratteri. Inizialmente si trova sul primo carattere del testo e, proprio come in altri editor, in Vim c'è una *clipboard*, destinata a memorizzare (e indirettamente copiare e spostare) frammenti di testo. La clipboard è inizialmente vuota.

In questo problema, assumeremo che all'inizio ci sia esattamente un carattere '-' (meno) nell'editor, e il nostro obiettivo è produrre esattamente n caratteri meno consecutivi. Per ottenere questo risultato, useremo quattro comandi:

- **h**: Se il cursore si trova sul primo carattere, questo comando non fa nulla, altrimenti sposta il cursore di un carattere a sinistra.
- **l**: Se il cursore si trova sull'ultimo carattere, questo comando non fa nulla, altrimenti sposta il cursore di un carattere a destra.
- **Y**: La sequenza di caratteri che va dalla posizione del cursore fino alla fine del testo viene copiata nella clipboard degli appunti, "sovrascrivendo" qualsiasi contenuto precedente. (Se il cursore si trova sul primo carattere del testo, l'intero testo verrà copiato nella clipboard).
- **P**: Una copia del testo memorizzato nella clipboard viene inserita prima del carattere su cui si trova il cursore, e il cursore viene spostato sull'ultimo carattere inserito. Il contenuto della clipboard non viene modificato. Se la clipboard è vuota, non succede nulla.

Scrivi un programma che legga il numero n e stampi il numero minimo di comandi necessari per terminare con esattamente n caratteri '-' consecutivi in Vim nelle condizioni descritte. Il programma deve anche stampare una sequenza con il numero minimo di comandi.

Input

La prima riga contiene il numero di casi di test t . Le successive t righe contengono i casi di test con la lunghezza desiderata della stringa n .

Vincoli

- $1 \leq t \leq 100$
- $1 \leq n \leq 10^7$

Subtask

Se stampi solo il numero corretto di comandi, ma nessuna sequenza o una sequenza di mosse sbagliata, riceverai la metà dei punti per quel subtask.

- Subtask 1 (20 punti): $n \leq 100$
- Subtask 2 (8 punti): $n \leq 1000$
- Subtask 3 (18 punti): $n \leq 10^4$
- Subtask 4 (18 punti): $n \leq 10^5$
- Subtask 5 (18 punti): $n \leq 10^6$
- Subtask 6 (18 punti): Nessuna limitazione aggiuntiva.

Output

Per ciascun caso di test, stampa il numero minimo richiesto di comandi e un esempio di tale sequenza di comandi sulla propria riga.

Esempio

Input

2
21
2

Output

10 YPYPhPYPPP
2 YP

Commento

Come possiamo vedere nella tabella seguente, nel primo caso, la sequenza YPYPhPYPPP fornisce il risultato corretto. Il carattere '=' nella colonna Schermo rappresenta il carattere '-' su cui si trova il cursore.

Passo	Comando	Schermo	Clipboard
0		=	(vuota)
1	Y	=	-
2	P	=-	-
3	Y	=-	--
4	P	=---	--
5	h	=---	--
6	P	=-----	--
7	Y	=-----	-----
8	P	-----=-----	-----
9	P	-----=-----	-----
10	P	-----=-----	-----