



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema A - Abejas

Límite de tiempo: 1 segundo

### Problema

Era un hermoso viernes al medio día en la FES Acatlán, los jóvenes estudiantes descansaban tranquilamente en los pastos y las bancas de las áreas comunes de la FES, cuando de pronto, un cúmulo de abejas comenzó a atacar a las personas, las abejas salían de todos lados y picaban a cualquier persona que se pusiera en su camino, todos comenzaron a correr tratando de huir de las abejas pero para algunos estudiantes les fue imposible escapar, extraño ¿no?

Después de varios días de investigación, los integrantes del Grupo de Algoritmia resolvieron el gran misterio acerca del evento de las abejas, un estudiante de MAC, llamado Peter, fue el culpable del acontecimiento. Peter era un chico muy extraño y callado, descubrieron que tenía un odio irracional hacia las personas y por eso aventó un panal de abejas aquel día.

A pesar de todo, Peter era un chico muy inteligente y no hacía las cosas al azar, descubrieron que, basándose en sus conocimientos de matemáticas y computación, logró desarrollar un sistema perfectamente elaborado para molestar a la gente. Peter veía a la FES como una gran cuadrícula (similar al plano cartesiano) donde podía identificar la posición de cualquier objeto con sus coordenadas en  $X$  y  $Y$ , así es como logró identificar las posiciones de todos los panales de abejas que había en la FES. Ya que tenía identificados los panales sólo esperaba a que hubiera una hora pico donde cierto número  $N$  de personas se encontraran cerca de un panal, en ese momento comenzaba su proceso para molestar. Peter sabía que si golpeaba el panal con cierta fuerza  $F$  las abejas saldrían muy enojadas y picarían a todo aquel que se encontrara en un radio  $R$  alrededor del panal, muchas veces era imposible poder afectar las  $N$  personas cerca del panal así que Peter elegía un número  $K$ , ( $K \leq N$ ), de personas a las que al menos quería afectar. Jamás se logró descubrir como es que Peter calculaba el nivel de fuerza  $F$  con el que era necesario golpear un panal.

La historia de Peter es muy interesante, tan interesante que tú estás obsesionado con poder emular el proceso que tenía Peter para molestar a la gente, así que desarrollarás un programa al que dadas las coordenadas de un panal, el número de personas cerca del panal y el número de personas a las que Peter quería molestar, calcule el área mínima de la zona afectada por las abejas para poder molestar al menos al número de personas que Peter eligió.

### Entrada

La primer línea de entrada será un entero  $C$ , ( $1 < C \leq 500$ ) que indica el número de casos de prueba. Cada caso de prueba comienza con una línea que contiene dos números flotantes  $PX$  y  $PY$ , ( $-500 \leq PX, PY \leq 500$ ) que indican las coordenadas del panal, la siguiente línea contiene dos enteros  $N$  y  $K$ , ( $1 < K \leq N < 200$ ) que indican el número de personas cerca del panal y el número de personas que Peter quiere molestar respectivamente, de ahí siguen  $N$  líneas donde cada una contiene dos números flotantes  $Xi$ ,  $Yi$ , ( $-500 \leq Xi, Yi \leq 500$ ) que indican las coordenadas de la persona  $i$ , cada caso de prueba termina con un \*.

## **Salida**

Para cada caso de prueba la salida será un número flotante, redondeado a dos cifras después del punto decimal, que indica el área mínima de la zona afectada por las abejas para molestar al menos a las  $K$  personas. El valor de  $\pi$  que se debe utilizar es 3,14159265359. Para hacer los cálculos del problema será necesario utilizar números flotantes de doble precisión.

## **Entrada ejemplo**

```
2
0.0 0.0
4 2
0.0 1.0
0.0 2.0
0.0 3.0
4.0 0.0
*
0.0 0.0
4 4
0.0 1.0
0.0 2.0
0.0 3.0
4.0 0.0
*
```

## **Salida Ejemplo**

```
12.57
50.27
```

---

Mauricio Eduardo Montalvo Guzmán - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema B - Buscando Oro

Límite de tiempo: 3 segundos

### Problema

Has descubierto la Ciudad de Oro. Como te encanta el oro, has empezado a recolectarlo. Pero hay tanto oro que te estás cansando mientras lo recolectas, así que quieres saber cuál es el mínimo esfuerzo necesario para obtener todo el oro.

La ciudad está descrita con una cuadrícula  $2D$ , donde tu posición inicial está marcada con una '**x**'. Un espacio vacío se denotará con un '.'. Y los lugares que contienen oro se denotarán con una '**g**'. En cada movimiento puedes desplazarte a todos los 8 lugares adyacentes dentro de la ciudad.

### Entrada

La primera línea será un entero  $T$  ( $1 \leq T \leq 100$ ), que denota el número de casos. Cada caso empezará con dos enteros,  $m$  y  $n$  ( $0 < m, n \leq 20$ ) denotando los renglones y columnas de la ciudad, respectivamente. Las siguientes  $m$  líneas contienen  $n$  caracteres que describen la ciudad. Habrá sólo una '**x**' en la ciudad y a lo más 15 posiciones de oro.

### Salida

Para cada caso de prueba debes imprimir un entero indicando el mínimo número de pasos requeridos para recolectar todo el oro de la ciudad y regresar a la posición inicial.

### Entrada ejemplo

```
2
5 5
x....
g....
g....
.....
g....
5 5
x....
g....
g....
.....
.....
```

### Salida Ejemplo

Case 1: 8  
Case 2: 4



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema C - Cortando Pizza

Límite de tiempo: 1 segundo

### Problema

Ricardo dice ser muy inteligente y que su inteligencia y perspicacia le ayudarán en todo. Él dice que si su inteligencia le permite hacer una actividad realizando un esfuerzo físico menor ¿Por qué debe de cansarse más de la cuenta? También él dice que si usa su cerebro para realizar un menor esfuerzo no es por holgazanería sino una muestra elegante de su superioridad intelectual.

En alguna ocasión Ricardo se preguntó cómo cortar una pizza en siete rebanadas y repartirlas entre sus amigos, para ello el tamaño de las rebanadas puede no ser el mismo. Pensó un poco y llegó a la conclusión de que podía obtener las siete rebanadas realizando sólo tres cortes -de orilla a orilla- a la pizza, con un cortador de pizza. Aquí mostramos la forma en que Ricardo cortó aquella pizza:

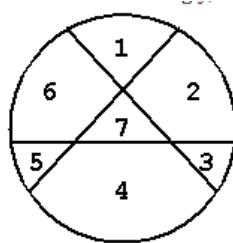


Figura 1: 7 rebanadas

Uno de sus amigos, quien nunca creyó en la inteligencia de Ricardo, pensó “Si Ricardo puede hacerlo, ¿Por qué no podría hacerlo mi computadora?” Así que este amigo intento hacer algo similar (pero no exactamente igual a él porque Ricardo lo criticaría por haberle robado la idea) con ayuda de su computadora. Él escribió un programa que dado el número de cortes en la pizza dice el número máximo de rebanadas que se pueden obtener con exactamente ese número de cortes.

Tu trabajo aquí es escribir un programa similar. Es seguro que el amigo de Ricardo no te criticará por hacer el mismo trabajo que él.

### Entrada

El archivo de entrada contiene un entero por linea  $n$  ( $0 \leq n \leq 4,295,113,503$ ) que representa el número de cortes que se deben hacer de lado a lado de la pizza. Un número negativo termina con la entrada.

### Salida

La salida debe ser un entero, el número máximo de rebanadas que se pueden producir.

**Entrada Ejemplo**

5  
10  
-100

**Salida Ejemplo**

16  
56

---

UVA Online Judge



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema D - Dark Souls

Límite de tiempo: 1 segundo

### Problema

¿Alguna vez has jugado Dark Souls? ¡Se dice que es uno de los juegos más difíciles que hay! Sus "boss fights" son casi imposibles, normalmente suele ser un enemigo enorme con muchos puntos de vida, pero cuando aparecen enemigos más chicos, aparecen en mayor cantidad e incluso puede ser mas difícil que un solo enemigo enorme. Un ejemplo de esto es cuando estás en el Valle de los Dragones y te atacan muchos al mismo tiempo. Cuando pasa esto tienes que buscar un punto ciego donde el ataque de los dragones no te alcance. En esta batalla hay dos tipos de dragones: dragones de fuego y dragones eléctricos. La forma de atacar de los dragones se ilustra en las siguientes cuadriculas.

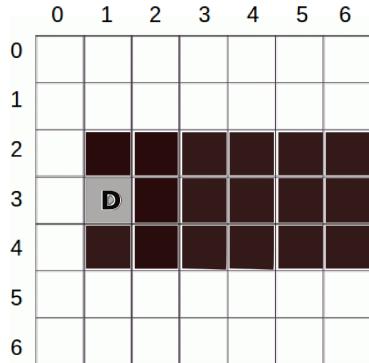


Figura 1: Dragon de fuego

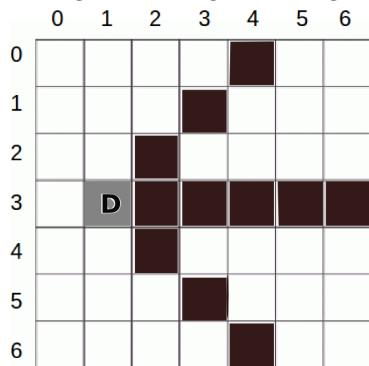


Figura 2: Dragon eléctrico

Los ataques se hacen en la dirección hacia donde los dragones están volteando (arriba, abajo, izquierda o derecha).

Tu trabajo es, dada una cuadricula de  $N \times M$ , la cantidad de dragones de fuego y eléctricos y sus posiciones, encontrar si existe una casilla que no es alcanzada por los ataques. No se puede estar en la misma casilla en la que está un dragón.

## Entrada

- Un entero  $T \leq 100$ , el número de casos.
- Dos enteros  $2 \leq N, M \leq 30$ , la dimensión de la cuadrícula.
- Dos enteros  $F$  y  $E$ , la cantidad de dragones de fuego y eléctricos respectivamente. - Después siguen  $F$  líneas y  $E$  líneas con las siguientes especificaciones: Dos enteros,  $X$  y  $Y$  y un carácter  $C$ , donde  $X$  y  $Y$  son las coordenadas del dragón en la cuadrícula ( $X$  indica en qué renglón y  $Y$  indica en qué columna) y  $C$  es la dirección hacia donde apunta el dragón: Arriba, Abajo, Izquierda Y Derecha, representados por los caracteres U, D, L y R respectivamente.

## Salida

Por cada caso de entrada se debe imprimir “Pelear” si existe una casilla que no es alcanzada por los ataques o imprimir “Escapar” en el caso contrario.

## Entrada ejemplo

```
2
9 9
3 0
0 1 D
8 4 U
0 7 D
3 4
1 2
0 1 D
0 0 R
1 0 R
```

## Salida Ejemplo

```
Escapar
Pelear
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema E - Escribiendo Mensajes

Límite de tiempo: 3 segundos

### Problema

El flojo fue víctima de un asalto, le robaron su celular y ahora tiene un viejo celular que tiene un teclado como el siguiente:

		abc	def	
	ghi	jkł	mno	
	pqrs	tuv	wxyz	
		<SP>		

Figura 1: Teclado

Al flojo Mau no le gusta ese celular porque para escribir una letra tiene que presionar varias veces una misma tecla. Por ejemplo, para escribir la “a”, él tiene que presionar la misma tecla una vez, pero para escribir la “b” tiene que presionar la misma tecla dos veces, y para la “c”, tres veces. Para escribir un espacio, basta con presionar una vez la tecla <SP>.

El flojo Mau tiene que mandar un mensaje de texto con su celular, pero como es muy flojo no quiere mandar el mensaje si debe presionar muchas teclas. Tu tarea es ayudar a contar el número de teclas que el flojo Mau debe presionar para poder escribir un mensaje.

### Entrada

Un entero  $T$ , el número de casos de prueba. Las siguientes  $T$  líneas contienen sólo espacios y caracteres en minúscula.

### Salida

Para cada caso de prueba debes imprimir una línea especificando el número del caso seguido del número de teclas que el flojo Mau debe presionar para escribir el mensaje de ese caso.

## **Entrada ejemplo**

```
2
el flojo mau es muy flojo
come frutas y verduras
```

## **Salida Ejemplo**

```
Case #1: 52
Case #2: 48
```

---

UVA Online Judge



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema F - ¿Y el problema F?

Límite de tiempo: 1 segundo

### Problema

Se tiene una enorme cadena de caracteres la cual está codificada y tiene ciertas contraseñas escondidas, dependiendo del tipo y cantidad de contraseñas que contenga la cadena original se le asigna un valor, sabemos que podemos dividir dicha cadena en  $N$  subcadenas que son propuestas como útiles, estas  $N$  subcadenas pueden caer en uno de los siguientes 4 tipos de cadenas: cadenas palíndromos regulares, cadenas espejo, cadenas palíndromos espejo y cadenas basura, cada una de estas cadenas tiene un significado especial y un valor :

1. Palíndromos Regulares - Valor 1:

Cadena de caracteres que se lee igual hacia atras que hacia adelante. Ejemplo: "RECONOCER"

2. Cadena Espejo - Valor 2:

Cadena de caracteres que, cuando se lee de atrás hacia adelante, cada caracter tiene su respectivo "caracter espejo". Ejemplo: "3SI2E".

Aquí se muestra la tabla de caracteres con sus correspondientes espejos, no todos los caracteres tienen un espejo.

Caracter	Espejo	Caracter	Espejo	Caracter	Espejo
A	A	M	M	Y	Y
B		N		Z	5
C		O	O	1	1
D		P		2	S
E	3	Q		3	E
F		R		4	
G		S	2	5	Z
H	H	T	T	6	
I	I	U	U	7	
J	L	V	V	8	8
K		W	W	9	
L	J	X	X		

Figura 1: Tabla de caracteres

### 3. Cadena Palíndromo Espejo - Valor 3:

Existen caracteres que son su mismo espejo, por ejemplo el espejo de “A” es “A” y el espejo de “8” es “8”, una cadena palíndromo espejo es un palíndromo compuesto únicamente por caracteres que son su mismo espejo. Ejemplo: “A8OMO8A”.

### 4. Cadena Basura - Valor 0:

Toda cadena que no cumple con alguna de las condiciones anteriores.

Las cadenas palíndromo espejo tengan son las más especiales debido a que los investigadores están seguros de que son contraseñas de cuentas bancarias codificadas, sabiendo ésto, podemos obtener el valor de la cadena orginal, que es la suma de los valores de sus subcadenas.

Tu tarea es, dado un grupo de  $M$  cadenas obtener cual es la que tiene mayor valor.

## Entrada

La primer línea de entrada será un número entero  $N$ , que indica el número de casos de prueba. Cada caso de prueba comienza con dos enteros  $M$ , ( $1 \leq M \leq 10$ ) y  $P$ , ( $1 \leq P \leq 50$ ), que indican el número de cadenas a evaluar y el número de subcadenas que contiene cada cadena, respectivamente. Cada cadena a evaluar está compuesta por dos líneas, la primera contiene la cadena y la segunda contiene  $P$  enteros que son los índices donde termina cada subcadena, la subcadena 1 comienza en el índice 0. La longitud de cada cadena no sobrepasa los 10000 caracteres.

## Salida

Para cada caso de prueba debe de imprimirse: la cadena que tiene mayor valor, su número de cadenas espejo, la suma de los valores de sus subcadenas y sus subcadenas útiles ordenadas lexicográficamente, las cadenas basura no se consideran como cadenas útiles. Si existen varias cadenas con el mayor valor debe seleccionarse la que aparezca primero en la entrada.

## Entrada ejemplo

```
1
2 3
NOTAPALINDROMEISAPALINILAPASIMIRRORED
13 28 36
2A3MEASATOYOTACOSAS
6 13 18
```

## Salida Ejemplo

```
2A3MEASATOYOTACOSAS
1 5
2A3MEAS
ATOYOTA
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema G - Guiando a Mau por el Bosque

Límite de tiempo: 3 segundos

### Problema

El flojo Mau se estresa mucho en su trabajo. Para relajarse después de un día difícil a él le gusta caminar a casa. Para hacer las cosas aún mejores, su oficina está en un lado del bosque y su casa del otro. Una caminata tranquila por el bosque viendo los pájaros y las ardillas es muy agradable. El bosque es hermoso y el flojo Mau quiere tomar una ruta diferente cada día. Él también quiere llegar a su casa antes del anochecer, por lo que siempre toma un camino para avanzar hacia su casa. Él toma un camino de A a B si existe una ruta desde B a su casa que es más corta que cualquier ruta posible desde A. Calcula cuántas rutas diferentes puede tomar el flojo Mau desde su oficina a la casa.

### Entrada

La entrada contiene varios casos de prueba seguidos por una línea que contiene un 0. El flojo Mau ha numerado cada intersección de diferentes caminos, empezando con el 1. Su oficina está numerada con el 1 y su casa con el 2. La primera linea de cada caso contiene el número de intersecciones  $N$  ( $1 < N \leq 10^3$ ), y el número de caminos  $M$ . Las siguientes  $M$  líneas contienen, cada una, un par de intersecciones y un entero  $d$  ( $1 \leq d \leq 1000$ ) indicando que hay un camino de longitud  $d$  entre dichas intersecciones. El flojo Mau puede atravesar un camino en cualquier dirección. Hay a lo más un camino entre cada par de intersecciones.

### Salida

Para cada caso de prueba debes imprimir un sólo entero módulo 1000000009 indicando el número de rutas diferentes que el flojo Mau puede tomar.

### Entrada ejemplo

```
5 6
1 3 2
1 4 2
3 4 3
1 5 12
4 2 34
5 2 24
```

7 8  
1 3 1  
1 4 1  
3 7 1  
7 4 1  
7 5 1  
6 7 1  
5 2 1  
6 2 1  
0

### Salida Ejemplo

2  
4

---

Waterloo Local Contest 2005 September 24 (Richard Krueger)



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema H - Haciendo Ananagramas

Límite de tiempo: 4 segundos

### Problema

Muchos aficionados a los crucigramas suelen utilizar anagramas (grupos de palabras con las mismas letras en diferente orden) por ejemplo: AMOR, ROMA, OMAR, MORA, RAMO, ARMO y MARO. Sin embargo, no todas las palabras cumplen con este atributo, ya que no importa como trates de ordenar sus letras no podrás formar otra palabra. Estas palabras se llaman ananagramas, un ejemplo es SEXY.

Es obvio que la definición anterior depende del dominio con el que nosotros estemos trabajando, tú podrías pensar que APARCAMIENTO es un ananagrama, pero cualquier médico te podría desmentir fácilmente ya que el te diría que existe METACARPIANO. Un posible dominio podría ser todo el idioma español, pero esto podría acarrear algunos problemas. Nosotros podemos restringir el dominio, por ejemplo, Geografía, en cuyo caso NEPAL se vuelve un anagrama relativo (PANEL no esta en el mismo dominio), pero QUERETARO no lo es ya que puede transformarse a TERRAQUEO.

Tú debes escribir un programa que lea un diccionario de un dominio restringido de palabras y determinar todos los posibles ananagramas relativos. Debes notar que las palabras de una sola letra son, ipso facto, ananagramas relativos ya que no se pueden reorganizar” de ningún otra forma. Nuestro diccionario no contendrá más de mil palabras.

### Entrada

La entrada consiste en una serie de líneas. Ninguna línea tendrá más de 80 caracteres de largo, pero puede contener cualquier número de palabras. Las palabras consisten de hasta 20 caracteres (mayúsculas y minúsculas) letras y no se cortarán entre líneas. Los espacios pueden aparecer libremente entre las palabras y por lo menos un espacio separa múltiples palabras en la misma linea. Tenga en cuenta que las palabras que contienen las mismas letras pero con diferente capitalización se consideran anagramas entre si, por ejemplo CoNsErVaDorA y cOnvErsAdOra son anagramas. El archivo de entrada termina con una linea consistente del símbolo #.

### Salida

La salida consistirá de una serie de líneas. Cada línea consistirá de una sola palabra que será un ananagrama relativo en el diccionario de entrada. Las palabras deberán ser puestas en orden lexicográfico (sensible a capitalización). Se asegura que siempre habrá por lo menos un ananagrama relativo.

### Entrada ejemplo

```
ladder came tape soon leader acme RIDE lone Dreis peat
ScAlE orb eye Rides dealer NotE derail LaCeS drIed
noel dire Disk mace Rob dries
#
```

## **Salida Ejemplo**

Disk  
Note  
derail  
drIed  
eye  
ladder  
soon

---

Waterloo Local Contest 2005 September 24 (Richard Krueger)



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema A - Annie la Hija de la Obscuridad

Límite de tiempo: 3 segundos

### Problema

League of Legends es un videojuego de género MOBA, desarrollado por Riot Games para Microsoft Windows y OS X.

Los jugadores (invocadores), se agrupan en 2 equipos de campeones (champions, o bien champs), 3 vs 3 o 5 vs 5. A partir de diciembre del 2013 hay 118 campeones disponibles en los servidores normales, pero este número aumenta periódicamente. Cada equipo comienza en lados opuestos de un mapa en un área llamada Base, cerca de lo que se llama Nexo (o Nexus). El objetivo del juego es destruir el Nexo del equipo rival. Para destruir un nexo, cada equipo debe llegar a la base enemiga eliminando unas torretas que la protegen.

Cada jugador gana niveles al matar súbditos (NPCs que aparecen constantemente y atacan al otro equipo) del equipo contrario y derrotar a monstruos neutrales. Matar monstruos, campeones enemigos y destruir torretas proporciona oro necesario para mejorar al Campeón y facilitar así las batallas.



Figura 1: Arriba a la izquierda “Tibbers”, arriba al centro “Incinerar”, Abajo a la izquierda “Desintegrar”

Uno de estos champs es Annie, tu objetivo es ayudarla a determinar cuando podrá encadenar sus tres hechizos ofensivos a un campeón enemigo. Los hechizos que conoce Annie son los siguientes:

- “Desintegrar” Annie lanza una bola de fuego imbuida de Maná a cualquier campeón en un area de  $600u^1$  a su alrededor.

<sup>1</sup>u son unidades de longitud en los campos de la justicia

- “Incinerar” Annie lanza un abrasador cono de fuego, dañando a todos los enemigos de la zona. Este cono tiene una apertura de  $35^\circ$  y además tiene un alcance de  $500u$  (imagine una rebanada de pizza circular, con centro en Annie).
- “Invocar: Tibbers” Tibbers aparece en un estallido de llamas e infinge daño mágico a los enemigos en la zona objetivo. Annie puede invocar Tibbers en cualquier lugar a no más de  $700u$ , Tibbers inflinge daño de área donde aparece en un círculo de radio  $75u$ .

Puedes asumir de manera segura que no hay obstáculos entre Annie y su objetivo; además puedes despreciar el tiempo entre el lanzamiento de cada hechizo.

## Entrada

La primer línea de la entrada será un número  $2 \leq T \leq 20$  que será el número de casos. Cada caso comenzará con tres números  $0 \leq A_x, A_y \leq 10000$  y  $1 \leq Q \leq 50$  donde los primeros dos son las coordenadas de Annie y el tercero el número de consultas para ese caso. Las siguientes  $Q$  líneas tendrán dos números  $0 \leq E_x, E_y \leq 10000$  que es la posición del campeón objetivo. Se asegura que todos los números en la entrada son enteros.

## Salida

Para cada consulta deberás imprimir una línea que dirá “FULL COMBO” si puedes encadenar los 3 hechizos contra el campeón objetivo al mismo tiempo, si no puedes, imprime “OUTPLAYED”.

## Entrada Ejemplo

```
2
0 1 1
490 2
23 4 1
700 900
```

## Salida Ejemplo

```
FULL COMBO
OUTPLAYED
```

## Sugerencias

$$c^2 = a^2 + b^2$$



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo



## Problema B - Colores

Límite de tiempo: 1 segundo

### Problema

Manuel Nicolás tiene éstos códigos de colores:

0 0 0 = Blanco

0 0 1 = Azul

0 1 0 = Rojo

0 1 1 = Morado

1 0 0 = Amarillo

1 0 1 = Verde

1 1 0 = Naranja

1 1 1 = Negro

Pero su amigo Mir Ist Kalt cambió los códigos, tal que los 1's los volvió 0's, y visceversa.

Tu trabajo es ayudar a Manuel Nicolás a decirle a qué color corresponde cada código cambiado.

### Entrada

La primera línea contendrá un entero  $N$  ( $1 \leq N \leq 20$ ), el número de casos de prueba. Las siguientes  $N$  líneas tendrán 3 dígitos,  $d_{i1}, d_{i2}, d_{i3}$  separados por un espacio ( $0 \leq d_{i1}, d_{i2}, d_{i3} \leq 1$ ) ( $1 \leq i \leq N$ ).

## **Salida**

Se tendrá que imprimir  $N$  líneas, una por cada caso de prueba, todas terminando con salto de línea; cada línea tendrá el color correspondiente a su código.

## **Entrada Ejemplo**

```
4
0 0 0
1 1 1
0 0 1
0 1 1
```

## **Salida Ejemplo**

```
Negro
Blanco
Naranja
Amarillo
```

---

Sergio Adrián Lagunas Pinacho - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema C - CandyLand

Límite de tiempo: 1 segundo

### Problema

Mágicamente has sido transportado a la asombrosa tierra de CandyLand, donde hay tantos dulces de tantos sabores como nunca habías imaginado. Siendo un fanático de los dulces no puedes contener la felicidad y empiezas a probar todos los dulces existentes y asignas a cada dulce un valor de sabrosidad. Sin embargo no todo puede ser tan perfecto, tu tiempo en CandyLand está a punto de acabar y serás transportado de regreso a tu vida cotidiana, pero no quieres irte con las manos vacías.

Enfrente de ti hay una línea con  $1 \leq N \leq 10^6$  dulces numerados del 1 al  $N$ , donde al  $i$ -ésimo dulce le has asignado un valor de sabrosidad  $-100 \leq S_i \leq 100$ . Y convenientemente tienes a tu disposición un brazo robótico capaz de agarrar exactamente  $1 \leq K \leq 10^3, K \leq N$  dulces consecutivos de dicha línea, y una computadora. Debido a que hay algunos dulces que no te gustan tanto, quieres agarrar  $K$  dulces tales que la suma de su valor de sabrosidad sea el máximo posible. Como la cantidad de dulces es muy grande necesitas hacer un programa que te diga cuál es el índice del primer dulce de la izquierda que debe agarrar el brazo robótico para lograr tu objetivo.

### Entrada

La primer línea contendrá el número  $1 \leq T \leq 50$  de casos. Para cada caso habrá dos líneas. La primer línea de cada caso contendrá dos enteros  $N$  y  $K$ . La segunda línea de cada caso contendrá  $N$  enteros  $S_i, 1 \leq i \leq N$ .

### Salida

Para cada caso deberás imprimir dos enteros en una línea, el índice del primer dulce de la izquierda que debe agarrar el brazo robótico para lograr tu objetivo, si hay más de una opción imprime el que tenga el menor índice, y la suma máxima del valor de sabrosidad que puedes conseguir.

### Nota

El brazo robótico tiene que agarrar exactamente  $K$  dulces.

### Entrada ejemplo

```
1
10 4
1 -5 5 10 -1 3 -2 -3 9 4
```

### Salida Ejemplo

```
3 17
```

---

David Felipe Castillo Velázquez - Grupo de Algoritmia Avanzada y Programación Competitiva



## Problema D - Raíces digitales

Límite de tiempo: 1 segundo

### Problema

Recientemente Manuel Nicolás León descubrió qué es la raíz digital, y decidió compartir su conocimiento contigo.

Digamos que  $S(n)$  es la suma de los dígitos de  $n$ , por ejemplo,  $S(4089) = 4 + 0 + 9 + 8 = 21$ , entonces la raíz digital del número  $n$  es:

1.  $rd(n) = S(n)$       si  $S(n) < 10$
2.  $rd(n) = rd(S(n))$     si  $S(n) \geq 10$

Por ejemplo,  $rd(4098) = rd(21) = rd(3) = 3$ .

Manuel le tiene miedo a los números grandes, por eso los números con los que trabajará serán a lo más  $10^{100}$ . Para todos esos números, Manuel ha probado que  $rd(n) = S(S(S(S(n))))(n \leq 10^{100})$ .

Ahora Manuel quiere encontrar números rápidamente dada su raíz digital. El problema es que todavía no ha aprendido a hacer lo que te va a preguntar. Tu tarea es, dados los números  $k$  y  $d$ , encontrar números exactamente de  $k$  dígitos (sin ceros al principio), con su raíz digital igual a  $d$ .

### Entrada

La primera línea tendrá un número  $T$  ( $T \leq 500$ ) que representa el número de casos de prueba.

Las siguientes  $T$  líneas contendrán los casos de prueba, cada caso tendrá 2 números,  $k$  y  $d$  ( $1 \leq k \leq 100$ ;  $1 \leq d \leq 9$ ).

### Salida

Se imprimirán  $2T$  líneas, 2 por cada caso de prueba: la primera línea de cada caso de prueba tendrá el número  $n$  más grande tal que  $rd(n) = d$ , y el número de dígitos de  $n$  sea igual a  $k$ ; la segunda línea de cada caso de prueba tendrá el número  $n$  más chico tal que  $rd(n) = d$ , y el número de dígitos de  $n$  sea igual a  $k$ .

Puedes tener la seguridad de que dichos números siempre existen, y son únicos.

El primer dígito de cada número impreso no tiene que ser un 0.

**Entrada Ejemplo**

2  
1 3  
1 7

**Salida Ejemplo**

3  
3  
7  
7

---

Sergio Adrián Lagunas Pinacho - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

# Acatlán

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo



## E - Buscando asiento

Límite de tiempo: 1 segundo

### Problema

Manuel Nicolás León es un chico muy especial... cuando va a conferencias le gusta que la silla de su izquierda y la silla de su derecha estén desocupadas.

Éste año irá a una conferencia a la cuál irán sólo chicos igual de especiales que él (que les gusta que las sillas adyacentes a la suya estén desocupadas). Para el problema tienes que decir si es posible que todos los chicos, incluyendo Manuel, estén a gusto, ésto es, que todos los chicos tengan libres las sillas adyacentes a la suya.

### Entrada

La primera línea tendrá un número  $T$  que representa el número de casos de prueba.

Las siguientes  $T$  líneas contendrán los casos de prueba, cada caso tendrá 2 números,  $A$  y  $B$  ( $1 \leq A \leq B \leq 500$ ) indicando el número de chicos, y el número de sillas disponibles respectivamente.

Los chicos escogen las sillas uno por uno, escogiendo una silla que tenga sus 2 sillas adyacentes desocupadas (Si es una de las sillas de la orilla, basta con que la única silla adyacente que tiene esté desocupada).

Tu tarea es, dadas éstas condiciones, decir si los chicos se sentirán a gusto o no.

### Salida

Se imprimirán  $T$  líneas, una por cada caso de prueba, con alguna de las siguientes palabras: "Si" si siempre es posible que todos los chicos estén a gusto, "No" si es imposible que todos los chicos estén a gusto, o "Tal vez" si depende de cómo se vayan sentando los chicos.

### Entrada Ejemplo

```
3
1 3
2 3
3 3
```

### Salida Ejemplo

```
Si
Tal vez
No
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo



## F Fibonacci

Límite de tiempo: 1 segundo

### Problema

Manuel Nicolás León es un chico muy curioso, tanto que durante una de sus clases se le ocurrió una sucesión de dígitos parecida a la de fibonacci, la cuál empieza con 2 dígitos, 1 y 1, y para la cual cual(es)quiera dígito(s) que se le vaya a añadir es igual a los dígitos que equivalen a la suma de los valores numéricos de los 2 dígitos anteriores, así el tercer dígito sería  $1+1=2$ , el cuarto sería  $1+2=3$ , el quinto  $2+3=5$ , el sexto  $3+5 = 8$ , y así sucesivamente.

Lo interesante de ésta sucesión empieza del dígito 7 en adelante; dado que  $5+8 = 13$ , los dígitos 7 y 8 serían el 1 y el 3 respectivamente; el dígito que ocuparía la posición 9 sería  $1+3=4$ , y así sucesivamente, dando lugar a algo como ésto:

112358134...

Manuel Nicolás León se pregunta si hay alguna forma de saber cualquier dígito de la sucesión con ayuda de un programa, para lo cuál te ha pedido tu ayuda: tienes que programarlo.

### Entrada

La primera línea contendrá un entero  $N$  ( $1 \leq N \leq 20$ ), el número de casos de prueba. Las siguientes  $N$  líneas tendrán un número entero positivo  $S_i$  ( $1 \leq S_i \leq 10^9$ ) ( $1 \leq i \leq N$ ) que representa el dígito requerido.

### Salida

Se tendrán que imprimir  $N$  líneas, una por cada caso de prueba, todas terminando con salto de línea; cada línea tendrá un dígito  $D_i$  ( $0 \leq D_i \leq 9$ ) ( $1 \leq i \leq N$ ) que representa el dígito que está en la posición  $S_i$  requerido.

**Entrada Ejemplo**

```
6  
1  
2  
3  
7  
8  
9
```

**Salida Ejemplo**

```
1  
1  
2  
1  
3  
4
```

---

Sergio Adrián Lagunas Pinacho - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

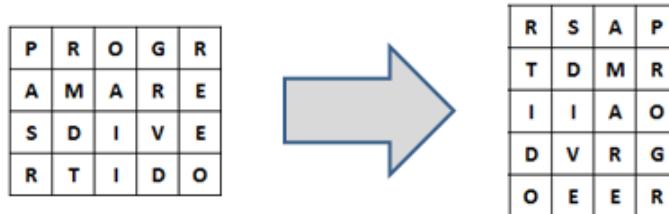
## Problema G - Código

Límite de tiempo: 1 segundo

### Problema

Mau es un chico al que le encanta descifrar códigos. Recientemente un amigo suyo le contó acerca de un nuevo código que ha estado desarrollando. Este código consiste en colocar el mensaje a codificar sin espacios en una cuadrícula de  $M \times N$ , siendo  $M$  el número de renglones y  $N$  el número de columnas. Luego se rota la cuadrícula 90 grados. Así, al leer normalmente el mensaje colocado en la cuadrícula, éste será incomprensible.

Un ejemplo se muestra en la siguiente imagen:



Mau puede decodificar este código, pero no lo suficientemente rápido, por eso te ha pedido ayuda a ti, su amigo programador, para implementar un programa que descifre este tipo de códigos.

Tu tarea será, dado el mensaje codificado, imprimir el mensaje original que fue colocado en la cuadrícula.

### Entrada

La primera línea contiene un entero  $T(1 \leq T \leq 50)$ , el número de casos de prueba. Para cada uno de los siguientes  $T$  casos siguen 2 líneas.

La primera de ellas contiene dos enteros  $M$  y  $N$ ,  $(1 \leq M, N \leq 50)$ , que representan el número de renglones y columnas respectivamente en las que esta colocado el mensaje original.

La segunda contiene el mensaje codificado, de tamaño  $M \times N$  el cual podrá estar formado de letras mayúsculas, letras minúsculas y números.

### Salida

Para cada caso imprime solo una línea que debe contener el mensaje original, con la distinción de mayúsculas y minúsculas.

### **Entrada Ejemplo**

```
1
4 5
RSAPTDRIIAODVRGOER
```

### **Salida Ejemplo**

```
PROGRAMARESDIVERTIDO
```

---

Maximiliano Vera Luna - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema A - Analizando Exámenes

Límite de tiempo: 1 segundos

### Problema

Acaba de pasar la semana de exámenes en la FES Acatlán y los profesores están preocupados porque sospechan que muchos alumnos hicieron trampa durante las pruebas. Los profesores ya calificaron los exámenes y ahora están buscando exámenes sospechosos. Ayúdalos con su tarea (Aunque hayas hecho trampa en algún examen).

Un profesor desea comparar dos exámenes y contar el número de coincidencias en las respuestas.

Haz un programa que lea los resultados de dos alumnos y cuente el número de respuestas iguales que tienen.

### Entrada

La primera línea contiene un entero  $n$ ,  $1 \leq n \leq 1000$ , el número de preguntas del examen. Las siguientes dos líneas contienen cada una  $n$  enteros  $a_i \in \{0, 1\}$ , los resultados de cada pregunta del examen donde 0 es respuesta equivocada y 1 es respuesta correcta.

### Salida

Deberás imprimir un entero  $k$  que indica la cantidad de coincidencias entre los dos exámenes.

#### Entrada Ejemplo

```
5
0 1 0 0 1
0 1 1 0 1
```

#### Salida Ejemplo

4



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema B - Buscando matriz

Límite de tiempo: 1 segundo

### Problema

Anselmo es alumno de MAC, como ustedes sabrán, al igual que la mayoría se encuentra en la semana de exámenes y tareas. Por fortuna cuenta con un buen equipo, por lo que hasta el momento todo ha marchado bien. Ha concluido casi todos sus trabajos, pero, como sabemos, siempre hay algo que se nos escapa, y para Anselmo no fue la excepción.

Hoy mientras platicaba con sus amigos en las jardineras, se percató de que una de sus tareas se entrega hoy a las 15.00 hrs !. Como dijimos, trataban de llevar los trabajos al día, por lo que parte de esta tarea ya está hecha, y como en todo equipo, a cada integrante le tocó una actividad, lo malo es que, de entre las partes que faltan, está la que le corresponde a Anselmo. Vamos a resumir el objetivo de la tarea a presentar en un enunciado:

Dados los diferentes tipos de pagos ( $S$ ) que se realizan en las cajas de la FES, y la Matriz de probabilidades de transición de un tipo de pago a otro, queremos saber, cuál es la probabilidad condicional de que la persona  $t + k$ ,  $k > 0$ , realice un pago  $X = b$  si sabe que la persona  $t$  hizo un tipo de pago  $X = a$ , donde  $a, b \in \{1, \dots, S\}$ .

A una parte del equipo le correspondió obtener una muestra de  $N$  personas observando que tipo de pago realizaba una persona al llegar a las cajas de la facultad (esto es lo que ya tienen). A a partir de ésta se debe obtener la Matriz de transiciones que se mencionó antes y con ella se podrán aplicar los métodos necesarios para poder responder la pregunta del párrafo anterior.

Bien, pues la creación de la matriz de transiciones es la parte que Anselmo debe de terminar. Como no quiere cometer errores porque sabe que de eso depende la calificación de todo el equipo te ha pedido a ti, (que eres muy listo y muy rápido) que lo ayudes, para que pueda acabar pronto.

Definimos la matriz de transición de la siguiente manera:

$$\begin{matrix} & \begin{matrix} 1 & 2 & \cdots & S \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ S \end{matrix} & \left( \begin{matrix} a_{11} & a_{12} & \cdots & a_{1S} \\ a_{21} & a_{22} & \cdots & a_{2S} \\ \vdots & \ddots & & \\ a_{S1} & a_{S2} & \cdots & a_{SS} \end{matrix} \right) \end{matrix}$$

Donde  $a_{ij}$  corresponde a la probabilidad de que la persona  $t + 1$  haga un pago  $j$  dado que la persona  $t$  hizo un pago  $i$ .

## Entrada

Un entero C,  $1 \leq C \leq 100$ , que es el número de casos a procesar seguido de una línea en blanco. Por cada caso de entrada, se recibirán dos enteros  $5 \leq N \leq 500$ ,  $3 \leq S \leq 15$  y a continuación N observaciones dónde aparecerán los tipos de pagos  $X = \{1, 2, \dots, S\}$ .

## Salida

Deberán aparecer C matrices de transición una por cada caso de la entrada, donde los elementos deberán estar redondeados a dos lugares decimales. Habrá una linea en blanco después de cada matriz.

## Entrada Ejemplo

```
2
8 4
1 2 3 1 1 3 2 4
5 3
1 2 3 3 1
```

## Salida Ejemplo

```
0.33 0.33 0.33 0.00
0.00 0.00 0.50 0.50
0.50 0.50 0.00 0.00
0.00 0.00 0.00 0.00

0.00 1.00 0.00
0.00 0.00 1.00
0.50 0.00 0.50
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema C - Contando nombres

Límite de tiempo: 1 segundo

### Problema

Kenny es un ser polinomial, es decir, se le conoce por varios nombres, e.g, “Kenny”, “Kevin”, “Kenan”. Queremos creer que a Kenny no le molesta tener varios nombres, pero al parecer a su novia sí; “¡Se llama Kenny!”, grita cada vez que alguien lo llama con otro nombre. Para intentar reducir su enojo, decidimos utilizar nombres que contengan las mismas letras que “Kenny”, i.e, anagramas. Después de un tiempo nos dimos cuenta que ahora teníamos nombres de sobra para Kenny, y nos dió curiosidad saber exáctamente cuántos. ¿Y por qué detenernos sólo con “Kenny”?

### Entrada

La primer línea contendrá  $t$ ,  $1 \leq t \leq 100$ , el número de casos de prueba. Para cada caso de prueba, en una línea, habrá una cadena  $S$ , compuesta por puras letras minúsculas,  $1 \leq \text{longitud}(S) \leq 200$ .

### Salida

Para cada caso de prueba tendrás que imprimir un entero  $k \bmod 10^9 + 7$  (1000000007), que indica el número de anagramas distintos que existen para la cadena  $S$ .

#### Entrada Ejemplo

```
3
kenny
algoritmia
aaaa
```

#### Salida Ejemplo

```
60
907200
1
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema D - Dados

Límite de tiempo: 1 segundo

### Problema

Seguramente has visto la serie “The Big Bang Theory”, y seguramente has visto el episodio donde Sheldon se basa en unos dados para tomar todas sus decisiones. Cada vez que Sheldon tenía que tomar una decisión lanzaba los dados y le asignaba a cada opción un número, dependiendo de la suma de los dados, elegía la opción asociada.

Nosotros queremos adoptar la estrategia de Sheldon, sin embargo hay momentos en que definitivamente tenemos preferencia por una opción, por ejemplo, cuando decidimos entre «hacer la tarea» o «programar», preferimos que salga la segunda.

De acuerdo a la cantidad de dados que usamos para decidir y el número de caras que tienen los dados, queremos que nos ayudes a saber qué número asociarle a nuestra opción preferida para tener la mayor probabilidad de elegirla.

### Entrada

La primer línea contendrá el número de casos de prueba  $t$ ,  $1 \leq t \leq 100$ . Para cada caso de prueba, en una línea, habrá dos enteros  $n$  y  $m$ ,  $1 \leq n, m \leq 50$ , el número de dados que usamos y el número de caras de los dados, respectivamente. Las caras de los dados están marcadas del 1 al  $m$ .

### Salida

Para cada caso de prueba tendrás que imprimir la suma de los dados que tiene la mayor probabilidad de salir, si existen varias soluciones, imprime la menor de ellas.

### **Entrada Ejemplo**

```
3
1 6
2 6
50 50
```

### **Salida Ejemplo**

```
1
7
1275
```

---

David Felipe Castillo Velázquez - Grupo de Algoritmia



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema E - Encriptando mensajes

Límite de tiempo: 1 segundo

### Problema

David es un chico singular, muy discreto, no le gustan las imprudencias. Por tanto es de esperarse que sus actividades las realice de manera muy reservada. Tanto así que en estos últimos días se le ha ocurrido la idea de encriptar sus mensajes que manda por Face a sus amigos. Una manera muy fácil de hacerlo, y muy conocida además, es utilizar la operación binaria XOR, esta es, dados dos enteros  $x$  y  $k$  donde al último lo usamos como clave, obtenemos un entero  $y = x \text{ XOR } k$  que puede ser usado para devolvernos el valor de  $x$  ejecutando la operación  $y \text{ XOR } k$ . Sabiendo esto, nos percatamos que con reconocer el valor de la clave  $k$  podemos enviar cualquier mensaje entre todos los que conozcan tal valor. Bien, como tú formas parte del círculo social de David deberás crear tu propio programa que encripte los mensajes que quieras enviarle, proporcionándote la clave que David ha elegido para ti.

### Entrada

La primera línea contendrá un entero  $C$  que son los casos a procesar. En los siguientes  $C$  bloques aparecerá un entero  $1 \leq k \leq 31$  en la primera linea y en la segunda linea el mensaje a encriptar. La linea no usará más 1000 caracteres.

### Salida

Serán  $C$  líneas, en cada una el mensaje encriptado para cada clave correspondiente.

## **Entrada Ejemplo**

```
2
5
Hola, que tal estas?
31
Oye, el uva esta caido verdad?
```

## **Salida Ejemplo**

```
Mjid)%tp`%qdi%`vqdv:
Pfz3?zs?ji~?zlk~?|~v{p?izm{~{
```

---

Kenny Yahir Méndez Ramírez - Grupo de Algoritmia



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema F - Focos

Límite de tiempo: 1 segundo

### Problema

El interruptor de la luz del cubo de Algoritmia se descompuso, pero afortunadamente, en el CEDETEC, hay un interruptor universal que cambia de estado, de forma un poco peculiar, los focos de todos los cubos. Después de un rato jugando con el interruptor descubrimos que en la  $i$ -ésima vez que accionamos el interruptor, los focos de los cubos cuya numeración es divisible por  $i$ , cambian de estado; si están apagados se prenden y si están prendidos se apagan. Por ejemplo, si es la segunda vez que accionamos el interruptor, los focos de los cubos 2, 4, 6, 8, ... cambian de estado. El número del cubo de algoritmia es  $n$  y suponemos que después de  $n$  veces que accionemos el interruptor, el cubo va a tener luz. Ayúdanos a comprobar si ésto es cierto. El foco del cubo está inicialmente apagado.

### Entrada

La primer línea contendrá el número de casos de prueba  $t$ ,  $1 \leq t \leq 100$ . Para cada caso de prueba, en una línea, habrá un entero  $n$ ,  $1 \leq n \leq 10^18$ , descrito anteriormente.

### Salida

Para cada caso de prueba tendrás que imprimir la cadena “les hace falta estudiar” si es que el foco del cubo de algoritmia termina apagado, o “son unos genios” en caso contrario.

#### Entrada Ejemplo

3  
7  
1  
16

#### Salida Ejemplo

les hace falta estudiar  
son unos genios  
son unos genios



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema G - Grupos

Límite de tiempo: 1 segundo

### Problema

La coordinación de MAC necesita asignarle los grupos a los alumnos de nuevo ingreso, sin embargo no quieren asignarlos completamente de forma aleatoria. Ellos quieren evitar que en un grupo haya dos o más personas con exactamente la misma personalidad, pues de ser así, esos alumnos podrían causar muchos problemas.

Se te ha pedido que ayudes a verificar si es posible realizar una asignación de los alumnos que cumpla los requerimientos de la coordinación, dados el número de alumnos, la personalidad de cada uno de ellos y el número de alumnos que debe tener cada grupo.

La personalidad de un alumno estará representada por un entero  $a_i$ . Todos los grupos deben tener la misma cantidad de alumnos.

### Entrada

La primera línea contiene un entero  $t$ ,  $1 \leq t \leq 100$ , el número de casos de prueba. Para cada caso de prueba habrá dos líneas. La primera contendrá dos enteros  $n$  y  $m$ ,  $1 \leq n, m \leq 100$ , el número de grupos y el número de alumnos que debe tener cada grupo, respectivamente. La segunda línea tendrá  $nm$  enteros  $a_i$ ,  $0 \leq a_i \leq 10^5$ , separados por espacio, que representan la personalidad del alumno  $i$ .

### Salida

Para cada caso de prueba debes imprimir, en una línea, la cadena “si”, si es posible asignar los alumnos de tal forma que no existan dos o más alumnos con la misma personalidad en el mismo grupo. De no existir una asignación, imprime la cadena “no”.

### **Entrada Ejemplo**

```
3
4 4
5 5 2 3 3 4 8 7 2 5 1 5 2 4 3 2
1 1
5
2 3
1 1 1 1 1 1
```

### **Salida Ejemplo**

```
si
si
no
```

---

David Felipe Castillo Velázquez - Grupo de Algoritmia



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema A - Asombroso League of Legends

Límite de tiempo: 3 segundos

### Problema

League of Legends es un juego clasificación MOBA (Multiplayer Online Battle Arena), en el que combaten varios campeones en enfrentamientos épicos, en los que el equipo que juegue mejor en equipo gana.

En éste problema nos centraremos en 2 campeones, Lux y Volibear; Volibear está persiguiendo a Lux, la cuál tiene poca vida, tal que si Volibear alcanza a Lux, ésta morirá sin poder hacer nada al respecto.

Afortunadamente, Lux tiene un hechizo que le permite hacer que Volibear deje de moverse durante cierto tiempo( $T$ ); el hechizo lo puede usar cada  $C$  segundos (a  $C$  también se le llama “enfriamiento” de hechizo).

Tu tarea consiste en, sabiendo las velocidades de los campeones (Lux y Volibear), el tiempo que el hechizo de Lux deja atrapado a Volibear, y el enfriamiento del hechizo de Lux, determinar si Lux será alcanzada por Volibear o no.

### Entrada

La primera línea contendrá un número  $N$  ( $0 < N \leq 100$ ), siendo  $N$  el número de casos de prueba. Cada una de las siguientes  $N$  líneas tendrá 4 enteros,  $V_1$ ,  $V_2$ ,  $T$  y  $C$ , ( $0 < V_1, V_2 \leq 500$ ) ( $0 < T, C \leq 10$ ) las velocidades de Volibear, Lux, el tiempo que el hechizo de Lux atrapa a Volibear, y el enfriamiento de su hechizo respectivamente.

(Las velocidades  $V_1$  y  $V_2$  están dadas en unidades/segundo, y  $T, C$  están dados en segundos. También supondremos que Volibear y Lux empiezan en puntos diferentes, Lux está adelante de Volibear, y ambos corren en línea recta)

### Salida

Para cada caso de prueba, se tendrá que imprimir una línea, imprimiendo “Se muere” si Volibear alcanza a Lux, y “Se salva” en caso contrario.

### Entrada Ejemplo

```
2
10 300 1 2
300 100 1 10
```

### Salida Ejemplo

```
Se salva
Se muere
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema B - Boby el Minotauro

Límite de tiempo: 3 segundos

### Problema

Desafortunadamente para el “flojo Mau”, un día se topó con el malvado minotauro devorador de hombres conocido como ... Boby, y cayó directo en su trampa. Boby no es cualquier minotauro, es un minotauro que habla, pero él sólo entiende frases que son palíndromos. Mau se dió cuenta que si le hablaba con palíndromos a Boby, él lo dejaría ir. Afortunadamente Mau tiene consigo un teléfono inteligente con un programa que identifica frases palindrómicas, el cual tú codificaste para él ... ¿o no?.

### Entrada

Te serán dadas muchas frases. Cada frase sólo contendrá letras mayúsculas de la ‘A’ hasta la ‘Z’ y los siguientes caracteres: ‘.’, ‘,’, ‘!’, ‘?’ . El final de la entrada será una línea conteniendo la palabra “HECHO”, que no deberá ser procesada. Cada frase tendrá a lo más 200 caracteres.

### Salida

Para cada frase imprime una línea con la palabra “NO SERAS COMIDO” si la frase es un palíndromo, o “OH NO!” si no es un palíndromo.

### Entrada Ejemplo

```
ROMA TIBI SUBITO MOTIBUS IBIT AMOR.  
ME DEJARIAS IR?  
ARRIBA LA BIRRA  
TRAIGAN AL MINOTAURO!  
HECHO
```

### Salida Ejemplo

```
NO SERAS COMIDO  
OH NO!  
NO SERAS COMIDO  
OH NO!
```

### Anotaciones

Un palíndromo es una frase que se lee igual de atrás hacia adelante y de adelante hacia atrás. Tienes que determinar si son palíndromos o no, ignorando signos de puntuación.

---

Modificado de un problema original de UVA Online Judge



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema C - Coleccionista

Límite de tiempo: 3 segundos

### Problema

Hernán es un joven apasionado por los videojuegos, le gustan de todo género, desde shooters, plataformas, carreras y estrategia hasta llegar a los de aventura. Uno de sus juegos favoritos es Pokémon. Pokémon es un juego en el que tu objetivo es conseguir todos los monstruos disponibles(llamados pokémon) y las medallas de todos los gimnasios para así demostrar que has vencido a los mejores.

Aunque a Hernán le encantan todas las versiones de este juego, desde Rojo y Azul hasta X y Y, él no siempre quiere capturar a todos los pokémon disponibles en algunas versiones, ya sea porque considera que algunos son débiles o pequeños, e incluso feos.

Debido a esto(y a que no le gusta hacer algo tan fácil), te ha pedido que le ayudes con una sencilla tarea.

Dados el número total de pokémon disponibles en cada versión y dos listas de pokémon, la primera será la lista de los que ya posee y la segunda la lista de los que no le interesan, determina las siguientes dos cosas:

- Cuántos y cuáles pokémon le faltan por capturar (solo de los que le interesan).
- Cuantos y cuales pokémon que posee puede cambiar, ésto es, los pokémon que ya tiene y no le interesan.

Los pokémon se identificarán por enteros para facilitar su procesamiento.

Es sabido que Hernán no posee mas de una especie del mismo pokémon por versión.

### Entrada

La entrada consta de varios casos de prueba.

Cada caso consta de 3 líneas, la primera línea será un entero  $N$ , ( $10 \leq N \leq 1000$ ) que indica el total de pokémon en la versión del juego.

La segunda línea contiene un entero  $A$ , ( $0 \leq A \leq N$ ) el número de pokémon que Hernán ha capturado en esa versión, seguido por una lista de  $A$  enteros que representan a estos pokémon. Todos estos números están separados por un espacio.

La tercera línea contiene un entero  $X$ , ( $0 \leq X \leq N$ ) el total de pokémon que no le interesan a Hernán, seguido por  $X$  enteros que representan a los mismos. Todos estos números están separados por un espacio.

Para facilitar su identificación, los pokémon estan representados por números desde 1 hasta  $N$ .

## Salida

Para cada caso imprime dos líneas:

- La primera línea será un entero  $F$  que representará el número de pokémon que le interesa capturar a Hernán y aún no tiene, seguido por  $F$  enteros que representan estos pokémon impresos en orden ascendente.
- La segunda constará de un entero  $C$  que representará el número de pokémon que Hernán puede cambiar, seguido por  $C$  enteros que serán los pokémon cambiados impresos en orden ascendente.

## Entrada ejemplo

```
11
5 3 5 9 10 11
6 2 3 4 5 9 11
```

## Salida ejemplo

```
4 1 6 7 8
4 3 5 9 11
```

---

Maximiliano Vera Luna - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema D - Dominó

Límite de tiempo: 1 segundo

### Problema

Mauricio tiene  $N$  piezas de dominó en fila. Cada pieza se divide en dos partes iguales en tamaño, - la superior y la inferior -. Cada una de las partes contiene un número del 1 al 6.

A Mauricio le encantan los números pares, así que quiere que la suma de los números de las mitades superiores y la suma de los números de las mitades inferiores sea par.

Para lograr eso, Mauricio puede rotar las piezas de dominó 180 grados. Después de una rotación las mitades cambian de lugares. Esta acción toma 1 segundo. Ayuda a Mauricio a averiguar el tiempo mínimo que debe gastar girando piezas de dominó para que se cumpla lo que él quiere.

### Entrada

La primera línea contiene un entero  $T(1 < T < 50)$ , el número de casos de prueba. Para cada uno de los siguientes  $T$  casos, hay una línea que contiene un entero  $N(1 \leq N \leq 100)$ , el número de piezas de dominó que tiene Mau. Las siguientes  $N$  líneas contienen dos enteros  $X_i, Y_i, (1 \leq X_i, Y_i \leq 6)$  cada una, separados por un espacio. El número  $X_i$  es el que está escrito inicialmente en la mitad superior del  $i$ -ésimo dominó, mientras que  $Y_i$  es el que está escrito en la mitad inferior.

### Salida

Para cada caso imprime sólo una línea, esta deberá decir “Caso T: ” seguido por el mínimo de segundos requeridos para realizar la tarea. T es el caso actual empezando a numerarlos desde 1. Si resulta imposible lograr tal tarea, imprime un ‘-1’ (sin comillas).

### Entrada Ejemplo

```
2
2
4 2
6 4
1
2 3
```

### Salida Ejemplo

```
Caso 1: 0
Caso 2: -1
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema E - Esto es fácil

Límite de tiempo: 5 segundos

### Problema

Este problema es fácil, tienes que buscar 2 enteros en una lista, tal que sumados den otro número.

#### Entrada

La primera línea tendrá un número  $T$  que representa el número de casos de prueba.

Las siguientes  $3T$  líneas contendrán los casos de prueba, cada caso de prueba tendrá 3 líneas, la primera línea tendrá un número  $N$  ( $2 \leq N \leq 10000$ ) que representa la cantidad de números que tendrá el arreglo, la segunda línea tendrá  $N$  números enteros  $K_i$  ( $1 \leq i \leq N$ ) ( $0 \leq |K_i| \leq 10000$ ) los números sobre los cuales tienes que buscar la pareja de números, y la tercera línea tendrá un número entero  $M$  ( $0 \leq |M| \leq 20000$ ).

#### Salida

Se imprimirán  $T$  líneas, una por cada caso de prueba, con la palabra “SI” si es que existen 2 números  $K_i$  y  $K_j$  tales que  $K_i + K_j = M$ , y “NO” en caso contrario.

#### Entrada Ejemplo

```
2
3
1 2 3
4
5
6 7 8 9 10
11
```

#### Salida Ejemplo

```
SI
NO
```

---

Sergio Adrián Lagunas Pinacho - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema F - Falsa Simulación

Límite de tiempo: 1 segundo

### Problema

El siguiente problema consta de una simulación muy concreta... ¿o no?. Tendrán una matriz  $M$  de tamaño  $1234(\text{filas}) \times 5678(\text{columnas})$ . Está inicialmente llena con enteros desde el  $1, 2, \dots, 1234 \times 5678$  en orden ascendente de izquierda a derecha y de arriba a abajo. (La primera fila está llena con los enteros  $1, 2, \dots, 5678$ ; la segunda con los enteros  $5679, 5680, \dots, 11376$ , etc) Existen 4 tipos de comandos:

- “R  $x$   $y$ ” intercambia la  $x$ -ésima y  $y$ -ésima fila de  $M$ .
- “C  $x$   $y$ ” intercambia la  $x$ -ésima y  $y$ -ésima columna de  $M$ .
- “Q  $x$   $y$ ” imprime  $M(x, y)$  (El entero que está en la posición  $(x,y)$  en la matriz  $M$ ).
- “W  $z$ ” imprime  $x$  e  $y$  donde  $z = M(x, y)$ .

### Entrada

Te será dada una lista de comandos válidos, uno por línea. La entrada termina con el fin de archivo.

### Salida

Cada que recibas el comando “Q  $x$   $y$ ” imprime una línea con el valor  $M(x, y)$ , y cada que recibas el comando “W  $z$ ” imprime una línea con los valores  $x$  e  $y$  separados por un espacio, la posición en la que se encuentra el número  $z$ .

### Entrada Ejemplo

```
R 1 2
Q 1 1
Q 2 1
W 1
W 5679
C 1 2
Q 1 1
Q 2 1
W 1
W 5679
```

## **Salida Ejemplo**

5679

1

2 1

1 1

5680

2

2 2

1 2

---

Sphere Online Judge



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema G - Geómetra Hermann

Límite de tiempo: 3 segundos

### Problema

Durante el siglo XIX el matemático Hermann Minkowski investigó acerca de un tipo de geometría no eucliana, llamada geometría de taxi. En la geometría de taxi la distancia entre dos puntos  $T_1(x_1, y_1)$  y  $T_2(x_2, y_2)$  es definida como  $D(T_1, T_2) = |x_1 - x_2| + |y_1 - y_2|$ . Cualquier otra definición se define igual que en la geometría eucliana, incluyendo la definición del círculo: Un círculo es el conjunto de todos los puntos en un plano a una distancia dada(radio) de un punto dado(centro del círculo). Nosotros estamos interesados en la diferencia de las áreas de dos círculos con radio  $R$ , uno de los cuales está dado en un espacio normal(euclídeano) y el otro en una geometría de taxi. Dicha tarea ha sido encomendada a tí.

### Entrada

Cada línea tendrá un solo número entero  $1 \leq R \leq 10000$ . Deberás leer hasta el final del archivo.

### Salida

Para cada caso deberás imprimir dos líneas, la primera contendrá el area del círculo con radio  $R$  en una geometría eucliana y la segunda línea será el area del círculo con radio  $R$  en una geometría de taxi. Tu salida deberá ser redondeada a 4 lugares decimales. Además puedes asumir de manera segura de usando valores flotantes de precisión doble y  $\pi$  igual 3.141592653589793 será suficiente.

### Entrada Ejemplo

21  
9384  
887

### Salida Ejemplo

1385.4424  
882.0000  
276646940.0487  
176118912.0000  
2471707.7105  
1573538.0000



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema I - Investigando laberintos

Límite de tiempo: 3 segundos

### Problema

Un laberinto es representado en una cuadrícula de dos dimensiones como está ilustrado en la figura siguiente.

```
XXXXXXXXXXXXXXXXXXXXXXXXXX
X   X   X   X   X   X
X           X   X   X
X   X   X   X   X   X
XXXXXX XXX XXXXXXXXXX
X   X   X   X   X   X
X   X   *       X
X   X   X   X   X   X
XXXXXXXXXXXXXXXXXXXXXX
```

Cada punto de la cuadrícula es representado por un carácter. Un carácter espacio ( ) representa los lugares en donde puedes caminar.

Las paredes del laberinto son representadas por letras mayúsculas (de A a Z), esto es, lugares dónde no puedes caminar.

Tu tarea es, dado un punto de inicio dentro del laberinto, representado por un asterisco (\*), debes de marcar todos los posibles lugares a donde puedes llegar caminando con el carácter gato (#). El asterisco debe ser reemplazado por el carácter gato.

La forma en que puedes caminar es 4 conexidad, esto significa que solo puedes moverte hacia arriba, abajo, izquierda o derecha.

La cuadrícula presentada anteriormente quedaría:

```
XXXXXXXXXXXXXXXXXXXXXX
X###X###X###X   X   X
X#####X#####X   X   X
X###X###X###X   X   X
XXXXXX#XXX#XXXXXXX
X   X###X###X###X#X#
X   X#####X#####X#
X   X###X###X###X#X#
XXXXXXXXXXXXXXXXXXXX
```

## Entrada

La primera línea de entrada es  $T$ , el número de casos. La primera línea de cada caso son dos enteros,  $N$  y  $M$ , el número de renglones y el número de columnas de la cuadrícula respectivamente.

Las siguientes  $N$  líneas contienen  $M$  caracteres, las cuales representan al laberinto.

## Salida

$N$  líneas de  $M$  caracteres cada una, las cuales representan a la cuadrícula marcada en la forma en que se indicó. Imprime una línea en blanco al final de cada caso.

## Entrada Ejemplo

```
2
5 5
XXXXX
X* X
X X
XX XX
XXXXX
5 9
AAAAAAA
C* D B
C D B
C D B
EEEEEEE
```

## Salida Ejemplo

```
XXXXX
X##X
X##X
XX#XX
XXXXX

AAAAAAA
C##D B
C##D B
C##D B
EEEEEEE
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema H - Historia de los relojes

Límite de tiempo: 3 segundos

### Problema

El interés medieval en contribuciones mecánicas es bien ilustrado por la invención del reloj mecánico, el más viejo el cual es controlado por contrapesos, y un brazo oscilante con un engranaje. Ésto fué en 1386.

Los relojes controlador por resortes aparecieron por la mitad del siglo 15, haciendo posible construir mecanismos más compactos, y preparando el camino para el reloj portatil.

Los primeros relojes de péndulo controlador por resortes fueron comúnmente puestos en pequeñas repisas de pared, y después sobre repisas. Muchos relojes de soporte tenían un cajoncito para guardar la llave del vidrio. Los relojes de soporte más recientes, hechos después de 1660, tenían diseño arquitectónico, con pilares a los lados.

En los siglos 17 - 18 en francia, los relojes de mesa se convirtieron en un objeto de diseño monumental.

Uno de los primeros relojes atómicos fué un reloj controlado por amoniaco . Éste fué hecho en 1949 en el "National Bureau of Standards", Washington, D.C.

La historia de los relojes es fascinante, pero no tiene que ver con éste problema. En éste problema te darán el ángulo entre las manecillas de un reloj en forma de círculo, como el que se muestra en la figura, y tienes que indicar si existe una hora del día tal que el ángulo entre ambas manecillas sea el ingresado por el usuario.

Para éste problema, supondremos que el reloj tiene 60 marcas, una para cada minuto, y que tanto la manecilla de los minutos, tanto como la de las horas sólo pueden estar apuntando a alguna de las marcas.



## **Entrada**

Está dada por varias líneas, cada una describiendo un caso de entrada. Cada linea contiene un entero A que representa el ángulo entre las manecillas ( $0 \leq A \leq 180$ ).

## **Salida**

Para cada caso de entrada imprime una línea con la palabra “Caso i: ” (i es el número de caso), seguido por un espacio, seguido por un carácter. Si existe al menos una hora del día tal que el ángulo mínimo entre las manecillas del reloj sea exactamente A grados, imprime el carácter será ”Y”. En caso contrario será ”N”.

## **Entrada Ejemplo**

```
90  
65  
66  
67  
128  
0  
180
```

## **Salida Ejemplo**

```
Caso 1: Y  
Caso 2: N  
Caso 3: Y  
Caso 4: N  
Caso 5: N  
Caso 6: Y  
Caso 7: Y
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema D - El Desafío

Límite de tiempo: 1 segundo

### Problema

Fer y Dianita son grandes amigas, están la mayor parte del tiempo juntas y siempre se la pasan muy bien. Por alguna extraña razón a Fer le gustan los números naturales que son múltiplos del 3 y a Dianita los que son múltiplos del 5.

Entre las cosas que disfrutan hacer es comer frituras, pero a su amigo Kenny no le agrada, (el piensa que no son para nada saludables), así que un día deliveradamente ellas le propusieron lo siguiente: "Si nos dices cuál es la suma de nuestros números favoritos que son menores a  $N$ , hoy no comeremos papitas, más sin embargo si te equivocas, tú nos las tendrás que invitar". Kenny se preocupa por sus amigas, así que aceptó el desafío y no quiere equivocarse (además también es algo tacaño); la cuestión es que no es muy bueno haciendo cálculos tan rápido. ¿Podrías ayudar a nuestro amigo Kenny diciéndole cuál es la respuesta correcta dado un cierto  $N$ ?

### Entrada

La primera línea de entrada contendrá un entero  $C$ , que indica el número de casos de prueba. En las siguientes  $C$  líneas aparecerá un entero  $N$ .

### Salida

Para cada caso de entrada deberás imprimir una sola línea que contenga un entero, la respuesta al problema planteado.

### Restricciones fácil

- $C = 100$
- $1 \leq N \leq 50000$

## Restricciones difícil

- $C = 100000$
- $1 \leq N \leq 10^9$

## Entrada Ejemplo

3  
10  
15  
100

## Salida Ejemplo

23  
45  
2318

## Notas

- Se asegura que la respuesta cabe en un entero de 64 bits.



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema A

Límite de tiempo: 1 segundo

### Problema

Marta es una chica a la que le encanta conducir, ella es muy hábil al volante y conduce su auto a todas horas y a todos lugares en cualquier tipo de clima. Debido a esto y a la lluvia, el pobre automóvil de Marta ha sufrido una pequeña avería.

Durante estas fuertes lluvias, los limpiaparabrisas de su auto se han descompuesto. Ahora cada uno de los dos limpiaparabrisas realiza su recorrido (ida y vuelta) en tiempos distintos, lo que ocasiona que si ella los detiene cuando alguno de los dos llega al final de su primer recorrido, el otro se encuentre obstruyendo su vista y la distraiga, por lo que podría tener un accidente.

Por lo tanto Marta te ha pedido ayuda a tí. Dado el tiempo en segundos en que cada uno de los limpiaparabrisas de su auto hace su recorrido, calcula el mínimo de segundos que deben pasar para que ambos regresen a su posición de partida. Toma en cuenta que su posición de partida es la misma que tienen los limpiaparabrisas en buenas condiciones cuando están apagados.



### Entrada

La primera línea de entrada contiene un entero  $T$ , el número de casos de prueba. Las siguientes  $T$  líneas contienen dos enteros diferentes  $A$  y  $B$ , que representan los segundos que tarda cada uno de los limpiaparabrisas en su recorrido.

### Salida

Para cada caso de prueba imprime una sola línea con un entero que represente el resultado del cálculo explicado anteriormente.

### Entrada Ejemplo

2  
3 5  
16 12

## Salida Ejemplo

15  
48

## Restricciones

### Fácil

- $1 \leq T \leq 100$
- $1 \leq A, B \leq 500$  con  $A$  diferente de  $B$ .

### Difícil

- $1 \leq T \leq 10000$
- $1 \leq A, B \leq 40000$  con  $A$  diferente de  $B$ .

---

Maximiliano Vera Luna - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema C - Conejos

Límite de tiempo: 1 segundo

### Problema

Johanna es una brillante matemática amante de los conejos que recientemente consiguió una pareja de bellos conejos a los que nombró Fibonacci y Lucas (se desconoce cuál de los dos conejos es la hembra). Al cabo de unos meses los pequeños conejos se reprodujeron y Johanna empezó a darse cuenta que el número de parejas de conejos que habían cada mes eran: 1, 1, 2, 3, 5, 8, ... y no tardó en deducir que el número de parejas  $f_n$  en el mes  $n$  estaba dada por la recurrencia  $f_n = f_{n-1} + f_{n-2}$ .

Johanna tiene dos hermanas menores, Isabella y Karen, quienes también aman a los conejos, y es por eso que Johanna ha decidido regalarles todos sus conejos, sin embargo no quiere que se peleen entre ellas, por lo tanto necesita regalarles exactamente la misma cantidad de parejas de conejos a cada una.

Para poderles regalar la misma cantidad de pares de conejos a Isabella y Karen, Johanna necesita que el número total de parejas de conejos sea par, así que a ella sólo le interesan los meses en lo que esto sucede y a dichos meses los ha nombrado conejo-meses. Así en los primeros 4 conejo-meses el número de parejas de conejos son: 2, 8, 34 y 144. Johanna necesita tu ayuda para determinar cuántos conejos tendrá en el  $N$ -ésimo conejo-mes. Como este número puede ser muy grande Johanna te pide que imprimas el resultado módulo  $10^9 + 7$  (1000000007).

### Entrada

La primera línea contiene un entero  $t$ , el número de casos de prueba. Las siguientes  $t$  líneas contienen un entero  $N$

### Salida

Para cada caso de prueba debes imprimir, en una línea, el número de pares de conejos que tendrá Johanna en el  $N$ -ésimo conejo-mes.

## Restricciones

### Fácil

- $t = 100$
- $1 \leq N \leq 1000$

### Difícil

- $t = 1000$
- $1 \leq N \leq 10^{18}$

## Entrada Ejemplo

```
5
1
2
5
10
43
```

## Salida Ejemplo

```
2
8
610
832040
461493940
```

## Notas

Recordar que:

- $(a + b) \%M = (a \%M) + (b \%M)$
- $(a \times b) \%M = ((a \%M) \times (b \%M)) \%M$



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## 3er Concurso de Programación Competitiva MAC 2015

### Indicaciones

- La entrada del programa se debe leer de la entrada estándar `stdin`.
- La salida de programa se debe imprimir a la salida estándar `stdout`.
- No imprimir cadenas como: “Dame la entrada”, “La salida es:”
- No usar librerías no estándar

Ejemplo:

```
#include <conio.h> /*Esta libreria no es estandar/

int main()
{
    system("pause");//No usar esto
    getch();//Ni esto

    return 0;
}
```

- Para los programas hechos en Java el nombre de la clase será la letra del problema en mayúscula seguido de la palabra “main” por ejemplo `Amain.java`.



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema Ejemplo

Límite de tiempo: 1 segundos

### Problema

Dados 2 enteros  $A$  y  $B$  debes imprimir la suma de ellos.

#### Entrada

Una línea con un entero  $T$  que indica el número de casos. Las siguientes  $T$  líneas aparecerán 2 enteros  $A$  y  $B$ .

#### Salida

$T$  líneas con la suma que se pide.

#### Entrada Ejemplo

```
2
2 3
5 8
```

#### Salida Ejemplo

```
5
13
```

## Solución

Los siguientes códigos resuelven el problema anterior:

### ■ C

```
#include <stdio.h>

int main()
{
    int casos,a,b;

    scanf("%d",&casos);

    while(casos--)
    {
        scanf("%d %d",&a,&b);
        printf("%d\n",a+b);
    }

    return 0;
}
```

### ■ C++

```
#include <iostream>
using namespace std;

int main()
{
    int casos,a,b;

    cin >> casos;

    while(casos--)
    {
        cin >> a >> b;
        cout << a+b << endl;
    }

    return 0;
}
```

### ■ Java

```
import java.util.*;

class Bmain{//Ejemplo del nombre de la clase
//para el problema B

public static void main(String []args)
{
    Scanner sc = new Scanner(System.in);

    int casos,a,b;

    casos = sc.nextInt();

    while(casos-- > 0)
    {
        a = sc.nextInt();
        b = sc.nextInt();

        System.out.println(a+b);
    }
}
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema A - Armando elecciones

Límite de tiempo: 5 segundos

### Problema

En Maclandia se llevarán a cabo elecciones presidenciales muy pronto, pero dado que conllevan un costo muy elevado, su Instituto Electoral te ha encargado que les digas si es posible determinar quién ganará las elecciones presidenciales, ya que si esto es posible, les ahorrarías mucho dinero.

En Maclandia ocurren muchas situaciones que no ocurren en los demás países de su planeta Namek, por ejemplo: debido a ciertas influencias (dinero, dinero, dinero, nepotismo, etc.), el voto de algunas personas vale más que el de otras; además, si una persona vota por cierto candidato todos sus amigos votarán por el mismo candidato; aquellas personas que no tengan amigos con un voto definido, el día de las elecciones votarán por algún candidato según su sano juicio.

Las personas en Maclandia tienen agregados a todos sus amigos en su red social, y si una persona es amiga de otra, entonces esta otra también es amiga de esa persona. Es bien sabido que los candidatos votarán por ellos mismos.

Sin embargo en este país existen dos genios: uno malvado y uno bueno. Si alguien es amigo del genio malvado entonces el candidato por el que votaría perderá las elecciones, sin importar el valor de los votos que haya obtenido. En caso contrario si alguien es amigo del genio bueno, el candidato por el que votaría ganará las elecciones.

Si los genios intentan hacer que el mismo candidato gane y pierda las elecciones entonces se anularán sus poderes y no tendrán poder sobre las elecciones. Si algún genio no tiene amigos, entonces este no podrá ejercer su poder para influir en las elecciones. Los genios sólo pueden ser amigos (directa o indirectamente) de personas que ya tienen predeterminado su voto. Ninguna persona votará por dos candidatos, puesto que ninguna persona será amiga indirecta de dos candidatos al mismo tiempo. Un genio es considerado como una persona en sus relaciones de amistad.

### Entrada

La primera línea será un entero  $1 \leq T \leq 100$ , indicando la cantidad de casos de prueba. Cada caso de prueba se compondrá de varias líneas: la primera tendrá tres números  $n, c$  y  $r$  ( $2 \leq n \leq c \leq 500$ ,  $0 \leq r \leq (n + 2 - c)^2$ ) indicando el número de votantes, el número de candidatos y el número de relaciones de amistad respectivamente. Los votantes son identificados con los números 1 hasta  $n$  inclusive. La siguiente línea tendrá  $n$  números enteros ( $1 \leq a_j \leq 10000$ ), el  $j$ -ésimo número indicará la cantidad de puntos que vale el voto de la  $j$ -ésima persona. En la siguiente línea vendrán  $c$  números, los cuales indican el número de cada candidato.  $r$  líneas vendrán después, cada una con dos números  $-1 \leq a, b \leq n$ , indicando que las personas  $a$  y  $b$  son amigas. La persona 0 es el genio bueno, y la persona -1 es el genio malvado. Los casos de prueba están separados por una línea en blanco.

## Salida

Para cada caso, indica en una línea distinta: el número del candidato que ganará las elecciones, si es posible determinar quien las ganara; si no es posible determinar esto, imprime “**Es mas facil salir de la friendzone que saber quien ganara**” sin las comillas.

## Entrada Ejemplo

```
2
4 3 3
1 10 30 1
1 2 3
1 4
2 0
3 -1
```

```
5 3 2
1 50 10 2 20
1 2 3
1 4
2 -1
```

## Salida Ejemplo

```
2
Es mas facil salir de la friendzone que saber quien ganara
```

## Explicación de la entrada ejemplo

En el segundo caso, la persona 2 pierde las elecciones automáticamente porque es amiga del genio malvado, la persona 1 puede juntar 3 puntos resultado de la suma de sus puntos y los de sus amistades, la persona 3 puede juntar 10 puntos. Sin embargo no es posible determinar quien ganará la elección, a pesar de que la persona 3 tiene más puntos, porque la persona 5, que no es amiga directa/indirecta de algún candidato, cuando vote puede hacerlo por alguno de los dos (1 ó 3) inclinando la balanza a favor de cualquiera.



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema B - Buscando amigos

Límite de tiempo: 3 segundos

### Problema

Después de una intensa jornada de actividades en la semana de MAC, algunos alumnos han encontrado a quien consideran “el amor de su vida”. Sin embargo muchos estudiantes de MAC se consideran a sí mismos matemáticos puros y no le hablan a cualquier persona. Los alumnos que se inscribieron a la semana de MAC recibieron un id (numero de identificación) para poder entrar a las conferencias. Los matemáticos puros sólo le hablarán a la persona cuyo id  $a$  sea amigo de su id  $b$ . Un número  $a$  es amigo de  $b$  si la suma de los divisores de  $a$  (exceptuando sólo a  $a$ ) es igual a  $b$ , nota que si  $a$  es amigo de  $b$ ,  $b$  no necesariamente es amigo de  $a$ . Algunos alumnos de MAC han recurrido a ti debido a tus grandes conocimientos en estas áreas del saber para saber si “el amor de su vida” les hablará. Por causas fortuitas todos estos amores son matemáticos puros. Tu tarea es determinar si los alumnos tienen oportunidad de hablar con quienes consideran el amor de su vida.

### Entrada

La primer línea de la entrada contendrá un entero  $T \leq 200$ , el número de casos de prueba, a continuación se darán  $T$  renglones, cada uno con un caso de prueba. Cada renglón tendrá dos números  $a, b$ , tales que  $1 \leq a, b \leq 10000$

### Salida

Para cada caso de prueba imprime “Quiza y sin quiza”, sin las comillas si  $a$  es amigo de  $b$ ; imprime “Llora desconsoladamente” en caso contrario.

#### Entrada Ejemplo

```
3
6 5
3 10
1184 1210
```

#### Salida Ejemplo

```
Llora desconsoladamente
Llora desconsoladamente
Quiza y sin quiza
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema C - Caballo Sin Mecate

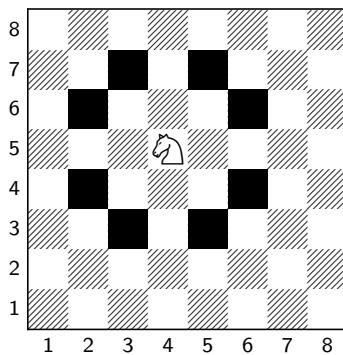
Límite de tiempo: 1 segundos

### Problema

Últimamente los integrantes del Grupo de Algoritmia se han obsesionado con el milenario juego de ajedrez, pero tienen un inconveniente, puesto que Moro siempre les gana, han decidido inventar un nuevo juego al que llamaron “Caballo sin Mecate”.

El juego consiste en un tablero de ajedrez de tamaño  $n \times m$  en el que interactúan dos jugadores. Dada una configuración de piezas tienes que llegar en la menor cantidad de pasos con el caballo de un punto a otro del tablero; gana el jugador que pueda dar la menor cantidad de movimientos. Además, se estipula que en el tablero sólo hay un caballo y múltiples piezas de otros tipos de las cuales no puede ocupar su lugar, pero sí saltarlas.

El caballo no se desplaza en línea recta, sino que tiene un movimiento característico llamado salto. El salto del caballo se parece a una L, y se compone de un desplazamiento de dos casillas en dirección horizontal o vertical, y otra casilla más en ángulo recto (en el diagrama siguiente, los cuadros marcados son los posibles lugares hacia donde puede desplazarse el caballo).



Tu tarea es ayudarnos a encontrar el número mínimo de movimientos que le tomará al caballo llegar a cierto punto, dada una configuración, para así poderle ganar a Moro.

## Entrada

La primer línea contendrá un solo número  $T$  ( $1 \leq T \leq 100$ ) que representa la cantidad de casos de prueba, que se seguirá de  $T$  casos de prueba. La primer línea de cada caso contendrá dos números enteros  $n$  y  $m$  ( $1 \leq n, m \leq 100$ ). Seguirán  $n$  líneas, una línea por cada fila del tablero, donde cada línea tendrá  $m$  caracteres  $c_{ij}$ , que representan el estado del escaque<sup>1</sup> de la  $i$ -ésima fila y la  $j$ -ésima columna (numerados de abajo hacia arriba y de izquierda a derecha como se muestra en el diagrama anterior). Donde  $c_{ij}$  puede ser una casilla vacía (\*), un peón ♙(p), una torre ♜(t), un alfil ♘(b), un rey ♚(k), una reina ♛(q) o un único caballo ♖(c) para todo el tablero. La última línea de cada caso tendrá dos enteros  $x, y$  ( $1 \leq x \leq n$ ,  $1 \leq y \leq m$ ) que representan la fila y columna a las que quieres llegar.

## Salida

Por cada caso deberás imprimir una sola línea que contendrá un entero que es la cantidad mínima de movimientos del caballo para llegar a su destino o -1 en caso de que sea imposible llegar a esa casilla.

### Entrada Ejemplo

```
2  
7 8  
*****  
*****  
*****  
*****  
*****  
*****  
bbbqkptt  
*c*****  
1 6  
2 2  
**  
c*  
2 2
```

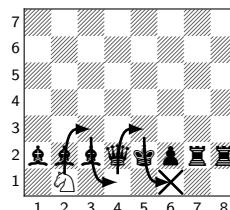
### Salida Ejemplo

```
4  
-1
```

## Notas

En el primer caso se puede llegar de la siguiente manera:

En el segundo caso no hay manera de llegar a la casilla (2, 2).



<sup>1</sup>Cuadrito del tablero



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema D - Doncella Jean

Límite de tiempo: 1 segundos



### Problema

En un lejano reino, existe una Doncella llamada Jean, es una chica muy bella pero también muy peculiar. Con lo anterior nos referimos a que tiene costumbres poco casuales. Para resumir, ella cuenta con muchos pretendientes en los reinos vecinos, incluyendo el suyo. Algunos de ellos son príncipes, duques y algunos otros con buen estatus social. No obstante hay un plebeyo que está enamorado de ella, para su fortuna (o no) es un “friend” de nuestra querida doncella, y como se imaginarán ambos tienen una amistad por Facebook.

Nuestro desdichado plebeyo (llamado Anselmo) últimamente ha tenido charlas muy frecuentes con la Doncella, él siente que varios de los mensajes de ella son indirectas pero no está muy seguro. Para cerciorarse de ello ha seleccionado  $N$  mensajes y los ha marcado como “Le gusto” y “No le gusto”. Él piensa que si la proporción de “Le gusto” es significativamente mayor a la de “No le gusto” entonces decidirá si está en la friendzone o no. Para no complicarse define que si la proporción de “No le gusto” es  $\geq 0,2$  entonces se encuentra en la friendzone.

Ayuda a Anselmo a determinar si se encuentra en la friendzone o no dados los  $N$  mensajes de las últimas conversaciones.

### Entrada

La primer línea es un número  $C$  que denota la cantidad de casos de entrada. En los siguientes  $C$  bloques aparecerán un número  $2 \leq N \leq 10000$  y  $N$  mensajes. Los mensajes serán “si” y “no” para denotar “Le gusto” y “No le gusto” respectivamente.

### Salida

Para cada caso deberás imprimir como respuesta “friendzone” cuando  $r \geq 0.2$  donde  $r$  denota la proporción de “No le gusto”, en caso contrario la carita feliz “:)”.

## Entrada Ejemplo

3

4

si

si

no

si

5

si

si

si

si

si

si

8

si

no

si

si

si

si

si

si

## Salida Ejemplo

```
friendzone
:)
:)
:
```



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema E - Explanada

Límite de tiempo: 1 segundos

### Problema

Uno de tus amigos ha decidido visitarte en la escuela, sin embargo él nunca ha venido a la FES y no sabe cómo encontrarte, por lo tanto has decidido ayudarle para que llegue a la explanada de MAC. La explanada la podemos ver como un polígono convexo<sup>2</sup> en el plano, y tú sabes muy bien las coordenadas de todos sus vértices, pues fue lo primero que te enseñaron en PIMAC. Para poder darle indicaciones a tu amigo necesitas saber si ya ha llegado a la explanada, así que le has pedido que te mande sus coordenadas. Ahora sólo necesitas hacer un programa que te diga si tu amigo está dentro de la explanada o no.

### Entrada

La primera línea contendrá un entero  $1 \leq t \leq 100$ , el número de casos de prueba. La primera línea de cada caso de prueba será un entero  $3 \leq n \leq 100$  que denota el número de vértices del polígono que representa a la explanada. En las siguientes  $n$  líneas de cada caso de prueba habrá dos enteros  $0 \leq x_i, y_i \leq 100$  representando las coordenadas del  $i$ -ésimo vértice del polígono. Los puntos estarán dados en orden de tal forma que forme un polígono convexo (puede ir en el sentido de las manecillas del reloj o en sentido contrario a las manecillas del reloj). La última línea de cada caso de prueba serán dos enteros  $0 \leq x_a, y_a \leq 100$ , las coordenadas de tu amigo. El punto  $(x_a, y_a)$  no va a estar nunca sobre el perímetro del polígono.

### Salida

Para cada caso de prueba deberás imprimir “si” (sin comillas) si tu amigo está dentro de la explanada, o “no” en caso contrario

<sup>2</sup>Un polígono convexo es una figura en la que todos sus ángulos interiores miden menos de 180 grados o  $\pi$  radianes.

## **Entrada Ejemplo**

```
2
5
1 3
2 5
4 5
5 3
2 1
4 2
5
1 3
2 5
4 5
5 3
2 1
3 3
```

## **Salida Ejemplo**

```
no
si
```

---

David Felipe Castillo Velázquez - Grupo de Algoritmia



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema F - Fiesta

Límite de tiempo: 1 segundos

### Problema

Después del intenso concurso de programación de la XIV Semana de MAC tu equipo piensa dar una gran fiesta en tu casa para celebrar su aplastante triunfo sobre los demás equipos.

El problema es que para entonces tu cerebro ya no podrá trabajar muy bien, puesto que lo exprimiste al máximo en el concurso, y necesitas saber cuál es el valor esperado de la cantidad de personas que asistirán a tu fiesta, por lo que desde ahora harás un programa que te ayude en esta tarea.

Lamentablemente es probable que no todas las persona asistan a tu fiesta, porque algunas no resolvieron muchos problemas y no se sienten de ánimo.

Se han invitado  $n$  personas a tu fiesta y cada persona tiene una probabilidad  $p_i$  ( $i = 1, 2, \dots, n$ ) de asistir, que está dada en función de su desempeño, es decir,

$$p_i = \sum_{j=1}^m \frac{r_{ij}}{m}$$

, con

$$r_{ij} = \begin{cases} \frac{1}{a_{ij}} & \text{si } a_{ij} \neq 0 \\ 0 & \text{si } a_{ij} = 0 \end{cases}$$

, donde  $a_{ij}$  es igual a la cantidad de intentos que le tomó a la persona  $i$  resolver el problema  $j$ ,  $a_{ij} = 0$  en caso de que la  $i$ -ésima persona no haya resuelto el  $j$ -ésimo problema, ( $j = 1, 2, \dots, m$ ), y  $m$  es la cantidad de problemas en total .

## Entrada

La primer línea contendrá un solo número  $T$  ( $1 \leq T \leq 100$ ) que representa la cantidad de casos de prueba.

La primera línea de cada caso contendrá dos números enteros  $n$  y  $m$  ( $1 \leq n \leq 100$ ,  $1 \leq m \leq 11$ ).

Seguirán  $n$  líneas, una línea por cada persona, donde la  $i$ -ésima línea tendrá  $m$  enteros  $a_{ij}$  ( $0 \leq a_{ij} \leq 100$ ) separados por un espacio.

## Salida

Por cada caso deberás imprimir una sola línea que contendrá el valor esperado de personas en tu fiesta redondeado a 2 decimales.

### Entrada Ejemplo

```
2
5 3
1 0 1
1 1 1
0 19 2
11 0 2
3 0 7
4 7
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
1 1 1 1 1 1 1
```

### Salida Ejemplo

```
2.21
4.00
```

## Notas

En el primer caso la primera persona pudo resolver el primer y tercer problema en un intento, mientras que la segunda resolvió todos, por lo que sus probabilidades de ir son 0,66 y 1 respectivamente.

En el segundo caso todos resolvieron todos los problemas en un solo intento, por lo que se espera que todos asistan.



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema G - Guíando a Edgar

Límite de tiempo: 1 segundos

### Problema

Edgar es una persona a la que le gusta optimizar su tiempo, por lo tanto cada que se tiene que desplazar de un lugar a otro dentro de la FES, le gusta irse por el camino que minimice el tiempo que le toma desplazarse. Además a él le gusta siempre pasar por lugares nuevos, porque asegura que eso ayuda a crear nuevas conexiones neuronales, entonces cada que planea la trayectoria para llegar de un lugar a otro, él considera un camino en específico por el que debe pasar obligatoriamente.

En este momento Edgar está demasiado ocupado encargándose del concurso, así que te ha pedido que le ayudes a organizar sus siguientes  $Q$  rutas para no tener que perder el tiempo.

Dado que la FES es muy grande, nos vamos a concentrar sólo en los  $N$  lugares favoritos de Edgar, numerados del 1 al  $N$ , con caminos entre cada uno de los  $N$  lugares. Los caminos entre cada par de lugares  $i, j$ , son bidireccionales, es decir, Edgar puede recorrer el camino de  $i$  a  $j$  o de  $j$  a  $i$ .

### Entrada

La primera línea tendrá un entero  $1 \leq T \leq 100$ , el número de casos de prueba. La primer línea de cada caso de prueba será un entero  $2 \leq N \leq 50$ , los lugares favoritos en la FES de Edgar, seguido vendrán  $N$  líneas, con  $N$  enteros cada una. El  $j$ -ésimo valor de la  $i$ -ésima línea,  $1 \leq w_{ij} \leq 1000$ , indicará el tiempo que le toma a Edgar recorrer el camino del lugar  $i$  al lugar  $j$  ( $w_{ij} = w_{ji}$ ). La  $N + 2$  línea de cada caso de prueba será un entero  $1 \leq Q \leq 100$ , el número de rutas que tienes que planear para Edgar, seguido de  $Q$  líneas, cada una con cuatro enteros  $1 \leq a, b, c, d \leq N$ , indicando que Edgar quiere ir del lugar  $a$  al lugar  $b$  pasando obligatoriamente por el camino entre  $c$  y  $d$  (en cualquier sentido),  $a \neq b, c \neq d$ .

### Salida

Para cada caso de prueba deberás imprimir  $Q$  enteros, representando el tiempo que el toma a Edgar realizar cada una de las rutas en el mismo orden en que fueron solicitadas.

## **Entrada Ejemplo**

```
1
5
0 9 5 2 1
9 0 5 2 2
5 5 0 2 6
2 2 2 0 7
1 2 6 7 0
2
5 1 5 2
5 1 2 3
```

## **Salida Ejemplo**

```
5
11
```

## **Explicación de la entrada ejemplo**

Para la primera ruta, Edgar puede tomar la siguiente sucesión de caminos: 5-2-5-1, con un tiempo total de  $2+2+1=5$ .

Para la segunda ruta, Edgar puede tomar la siguiente sucesión de caminos: 5-2-3-4-1, con un tiempo total de  $2+5+2+2=11$ .

---

David Felipe Castillo Velázquez - Grupo de Algoritmia



Facultad de Estudios Superiores

**Acatlán**

Centro de Desarrollo Tecnológico  
Departamento de Servicios de Cómputo

## Problema H - Hurgando en el CEDETEC

Límite de tiempo: 1 segundos

### Problema

Por fortuna el CEDETEC acaba de recibir presupuesto ilimitado y ha aumentado el número de cubículos desde  $-\infty$  a  $\infty$ . El grupo de Algoritmia ha aprovechado la expansión del CEDETEC y se mudó de cubículo.

Lamentablemente Kenny no se enteró de la mudanza y ahora no sabe en qué cubículo quedó el grupo de Algoritmia, pero como es muy importante que lo encuentre decidió ir buscando de cubículo a cubículo, sin embargo resulta que Kenny quiere ir preguntando de una manera un poco peculiar: puede empezar a buscar en cualquier cubículo, una vez que haya buscado en el cubículo  $n$  sólo puede buscar en el cubículo  $n \pm a$  o  $n \pm b$ .

Tu tarea consiste en saber si Kenny podrá algún día encontrar el cubículo de Algoritmia, sin importar a qué cubículo se mudó, conociendo los valores  $a$  y  $b$  que escogió Kenny.

### Entrada

La primera línea será un entero  $1 \leq T \leq 100$ , indicando los números de casos de prueba. Para cada caso de prueba habrá una única línea con dos enteros  $1 \leq a, b \leq 10^8$ , explicados anteriormente.

### Salida

Para cada caso de prueba debes imprimir “si” (sin las comillas) si Kenny podrá encontrar el cubículo de Algoritmia o “no” (sin las comillas), en caso contrario

**Entrada Ejemplo**

```
2
3 4
2 4
```

**Salida Ejemplo**

```
si
no
```