



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 X_i, Y_i , ($-500 \leq X_i, Y_i \leq 500$) 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
```



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:

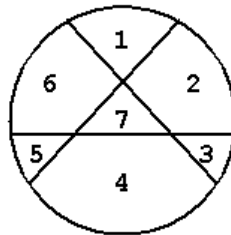


Figura 1: 7 rebanadas

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 línea 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

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 cuadrículas.

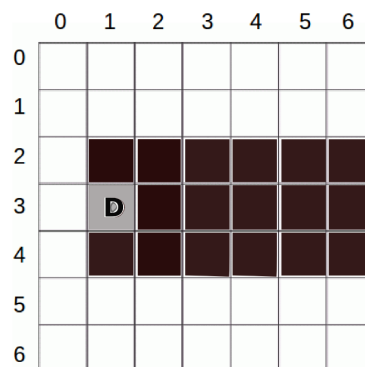


Figura 1: Dragon de fuego

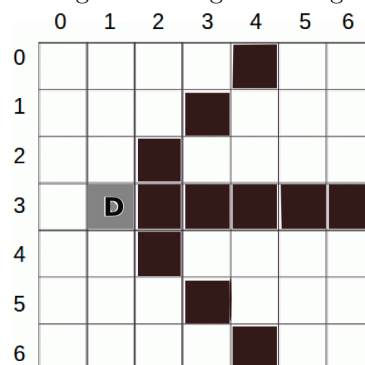


Figura 2: Dragon eléctrico

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

Tu trabajo es, dada una cuadrícula 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
```

Hernán Tellechea Garduño - Grupo de Algorimia Avanzada y Programación Competitiva



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  | jkl | 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 siguiente 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



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 atrás 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 tiene 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 tienen son las más especiales debido a que los investigadores están seguros de que son contraseñas de cuentas bancarias codificadas, sabiendo esto, podemos obtener el valor de la cadena original, 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 línea 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 línea. 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 línea consistente del símbolo #.

Salida

La salida consistirá de una serie de líneas. Cada línea consistirá de una sola palabra que será un anagrama 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)