



acm International Collegiate
Programming Contest

2011



event
sponsor

Nacional de Programación Bolivia

Sábado 24 de septiembre del 2011

Contents

Estafa	1
Sin Amigos	2
Fuzzy	3
Escalera	6
IAST	7
Vacaciones	9
Pintando	11

Las paginas están numeradas desde el 1 al 12.

Problem A

Estafa

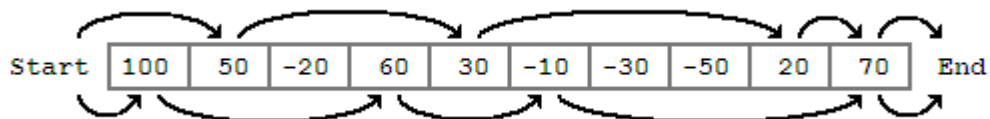
La entrada de datos es: la entrada standard,
la salida de datos es: la salida standard.

El negocio ha sido lento en el la venta de autos usados. En un esfuerzo por atraer a nuevos clientes, la administración ha creado el Programa de Incentivos de Reembolso de Diversión Fabulosa. Este es un juego simple que permite a los clientes para tratar de ganar un descuento en la compra de un automóvil. Es un juego de mesa donde se marca cada cuadrado con la cantidad de la rebaja. El cliente avanza haciendo girar una ruleta. Cada cuadro se cae en una casilla se suma a su cantidad marcada. Cuando llega al extremo del tablero se ve recompensado con la cantidad del reembolso total.

Por supuesto, dada la empresa en cuestión, no debería ser ninguna sorpresa que hay un par de condiciones por escrito y en la letra muy pequeña. La primera es que hay un límite para el número de intentos que el cliente tiene que terminar el juego, si no llega a la final en el número asignado de intentos entonces pierde su reembolso. La segunda es que algunas de las casillas en realidad tienen un valor negativo que resta en lugar de añadir. Un cliente en particular con mala suerte, incluso puede salir del juego con un descuento negativo.

A pesar de estas condiciones, a la gerencia le preocupa que alguien pueda ganar una gran rebaja en particular, algo que les gustaría evitar a toda costa. Su trabajo es tomar una configuración especial para el juego y decidir el descuento máximo que un cliente podría obtener.

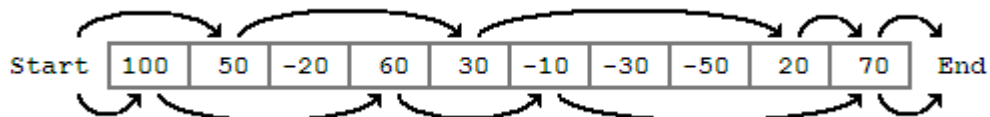
Consideremos, por ejemplo, el tablero de juego siguiente



Supongamos que tenemos 5 vueltas para terminar el juego, y cada vez nos podemos mover entre 1 y 4 espacios en función de lo que hacemos girar. Tenga en cuenta que tenemos que empezar justo antes de la primera casilla, por lo que gira a 1 nos lleva a la primera casilla. Observe también que debemos poner fin al caer más allá del final de la última casilla. No tiene por qué ser exacta, es suficiente cualquier número que nos lleva fuera del tablero.

La ilustración muestra dos maneras diferentes. Es posible que el juego podría ir siguiendo las flechas en la parte superior, si hacemos girar un 2, 3, 4, 1, 1, respectivamente, que va a ganar un reembolso total de $50 + 30 + 20 + 70 = 170$. Sin embargo, la rebaja mejor que podíamos ganar sería de 220. Esta cantidad se obtiene, si se giró la ruleta 1, 3, 2, 4, 1, respectivamente, como se muestra por el camino inferior. Tenga en cuenta que si tomamos todas las casillas con un número positivo, no habríamos sido capaces de llegar a la final de la tabla antes de 5 vueltas.

La ilustración siguiente:



muestra un juego donde tenemos 4 vueltas para terminar el juego, y puede mover hasta 3 plazas en cada turno. Una vez más, se muestran dos caminos diferentes, el de arriba ganan un descuento de -150 , y el de abajo -100 . De hecho, -100 es el descuento más alto que puede ganar en este juego (un hecho que sin duda agradará a la gestión de la empresa). Por supuesto, también puede haber una secuencia de movimientos en los que no llegamos a la final por ejemplo, limitar la girar un 1 cada vez. A pesar de no haber terminado en realidad sería preferible terminar con un descuento negativo, este problema sólo vamos a considerar las secuencias de movimientos que nos permiten llegar al final antes del límite de turnos.

Input

La entrada consiste de casos de prueba , seguido por una línea que contiene sólo indicando el final de los casos de prueba.

La primera línea de un conjunto de datos contiene tres enteros separados por espacios N, S, T , donde

- N es el número total de casillas del tablero, $2 \leq n \leq 200$.
- S es el número máximo de plazas es posible avanzar en cada turno, $2 \leq S \leq 10$.
- T es el número máximo permitido de vueltas, donde $N + 1 \leq ST$ y $T \leq N + 1$.

Luego continúan N una o más líneas con N enteros, que representan los números en el tablero. Cada número tiene una magnitud inferior a 10000.

Output

Para cada conjunto de datos escriba en una línea el descuento máximo posible que se puede ganar al completar el juego.

Para completar el juego al que debe avanzar a un total de $N + 1$ en T turnos, cada uno a su vez avanzar de 1 a S casillas inclusive.

Sample input	Output for the sample input
10 4 5 100 50 -20 60 30 -10 -30 -50 20 70 9 3 4 150 100 -200 -100 -300 -100 -200 100 150 0	220 -100

Problem B

Sin Amigos

La entrada de datos es: la entrada standard,
la salida de datos es: la salida standard.

Los números amigos son aquellos en los que la suma de los divisores de uno es el otro, por ejemplo:

$$220 = 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$$

$$284 = 1 + 2 + 4 + 71 + 142 = 220$$

Input

La entrada consiste de varios casos de prueba cada caso de prueba consiste en un número $2 \leq N \leq 1000000$. y termina cuando $N = 0$.

Output

Por cada caso de prueba imprima en una linea su número amigo. Si no tiene un amigo imprima -1.

Sample input	Output for the sample input
2	-1
6	6
1298	-1
5637	-1
284	220
7962	-1
220	284
496	496
0	

Problem C

Fuzzy

La entrada de datos es: la entrada standard,
la salida de datos es: la salida standard.

Con la identificación biométrica se han reducido bastante los fraudes y delitos a causa de la usurpación de identidad, por la poca fiabilidad de los métodos de identificación tradicionales, ya que en identificación biométrica se utilizan rasgos únicos de la persona.

Pero esta tecnología no es sencilla aunque así lo parezca. La arquitectura de esta consiste básicamente en 4 etapas: Captura y mejora de la impresión dactilar, Extracción de características, Coincidencia y Toma de decisiones.

En la primera se mejora la impresión dactilar proporcionada por un escáner dactilar, debido a que esta tiene mucho ruido y no se puede extraer las características. Una vez mejorada la imagen se divide en sectores y se clasifican para capturar características propias de la huella dactilar. Luego sigue la etapa de Extracción de Características, en esta se debe entrenar al computador para que aprenda un determinado patrón en la imagen clasificada, y proporcionar una salida como característica útil para formar una plantilla, que es la representación digital de la huella dactilar, esta es utilizada en la Coincidencia con otra plantilla y en la Toma de decisiones. La Toma de decisiones está de acuerdo a un modelo probabilístico donde se determina una tolerancia de aceptación de la Coincidencia.

Suponga que un sector de la imagen representa un patrón en una matriz de $M \times N$ elementos como un conjunto de entrenamiento, y queremos que el computador aprenda este, para proporcionar una salida deseada en un conjunto de M elementos.

Para aprender un patrón se debe tener en cuenta 3 aspectos:

1. La salida se define como:

$$salida = salida + pesos_i * entradas_i; \text{ Para } i = 1, 2, \dots, M$$

A este se le aplica un discriminador:

$$salida = 0 \text{ si el valor anterior fue } salida \leq 0, \text{ de lo contrario } salida = 1.$$

2. El error se define como:

$$error = salidadeseada - salida$$

3. El si el error es diferente de cero se recalculan los pesos con:

$$pesos_i = (\text{factor de aprendizaje} * error) * entradas_i + pesos_i; \text{ Para } i = 1, 2, \dots, N$$

Donde:

- entradas es la matriz de $M \times N$ elementos que representa un patrón.

- pesos es un conjunto de N elementos constantes.
- factor de aprendizaje es la velocidad de aprendizaje, $0 \leq factor_i \leq 1$.
- salida deseada es un elemento del vector de salida deseada.

La salida que debemos obtener debe ser igual a la salida deseada para aprender el determinado patrón. En este contexto el patrón se reduce a que cada elemento de la salida y/o la salida deseada representan al patrón de cada fila de la matriz de entrada. Normalmente el patrón es una de las 7 características locales de la huella dactilar, pero en este caso no nos interesa, por lo cual el patrón en la matriz no tiene un orden definido y podría ser cualquiera. Se toma como aprendido el patrón, cuando se ha logrado obtener la salida deseada en menos del Máximo de Iteraciones y además la secuencia de No Error es menor a la determinada.

Pero todo esto es únicamente el principio para que el aprendizaje del patrón de la huella dactilar funcione, por lo que tenemos que hacer pruebas a nuestro método para que funcione correctamente.

Su trabajo consiste en calcular las iteraciones necesarias para el aprendizaje del patrón de la huella dactilar.

Input

Para su trabajo tendrá varios casos de entrada, el primer caso contiene las dimensiones de tipo entero, de la matriz de entrada M, N , donde $2 \leq M, N \leq 5$ además siempre $M \geq N$, separados por espacios. Las siguientes M líneas contienen cada una, N elementos separados por espacios, donde cada elemento es uno de los valores 0,1 de tipo flotante. La siguiente línea contiene los M elementos del conjunto de salida deseada separados por espacios, donde cada elemento es uno de los valores 0,1 de tipo flotante. La última línea contiene el factor de aprendizaje F , cuyo valor es $0 \leq F \leq 1$, de tipo flotante.

El conjunto de pesos será constante de N elementos, el primer elemento siempre es 0.3, y va en secuencia hasta completar los N elementos, por ej., si $N = 3$ entonces $pesos = 0.3, 0.4, 0.5$; si $N = 5$, $pesos = 0.3, 0.4, 0.5, 0.6, 0.7$. Además considere el número Máximo de Iteraciones $MAX = 1000$ y la secuencia máxima de $Nro.Error = 5$.

La entrada termina cuando $M = N = 0$. La entrada debe ser leída por Standard Input

Output

La salida es una cadena con el número de iteraciones que le tomo al algoritmo en aprender el determinado patrón como el ejemplo de salida. Si en el número Máximo de Iteraciones no se ha podido aprender el patrón entonces debe imprimir "No aprendio.". La salida debe ser escrita por Standard Output.

Sample input	Output for the sample input
4 3 1 0 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0.3 5 5 1 0 0 0 0 1 1 0 0 0 1 1 1 0 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 0.7 0 0	Se aprendio en 2 iteraciones. Se aprendio en 8 iteraciones.

Problem D

Escalera

La entrada de datos es: la entrada standard,
la salida de datos es: la salida standard.

Se quiere construir una escalera lo mas anatómica posible, cuidando de que entre, escalón y escalón no exista una altura superior a dos unidades. Se desea cubrir una distancia horizontal de n con esta escalera. Para la construcción han provisto de una cantidad limitada de ladrillos (m) con la esperanza de alcanzar la máxima altura posible.

La figura muestra la construcción de una escalera de acuerdo a estas especificaciones: $m = 12$, $n = 6$

```

      =
      =
    = =
    = =
  = = = = = =

```

Figura 1: Construcción de la escalera

Input

La entrada contiene varias lineas, la primera es un numero entero c que es igual a los casos de entrada. Seguidamente se tiene c lineas, en cada linea de entrada se tiene un par de números que especifican m y n que son enteros positivos, m esta en $n \leq 1000$. m es el número de ladrillos y n la distancia que se debe cubrir.

Output

Por cada linea de entrada se debe generar una linea de salida, que muestre la cantidad de ladrillos que se apilan para formar la escalera, cada valor debe estar separado con un espacio en blanco, si fuera posible construir la escalera. SI no es posible construir la escalera se debe mostrar el mensaje *IMPOSIBLE*.

Cada linea de salida debe especificar el caso al que corresponde el resultado, para ello se prefija la cadena *Caso#* \therefore .

Sample input	Output for the sample input
4	Caso 1: 1 1 1 1 3 5
12 6	Caso 2: 1 1 3 4 6
15 5	Caso 3: 1 3 5 7
16 4	Caso 4: IMPOSIBLE
65 7	

Problem E

IAST

La entrada de datos es: la entrada standard,
la salida de datos es: la salida standard.

Una IAST es una cadena que cumple ciertas propiedades:

- Es una interjección de asombro seguida de una interjección de atención.
- Siempre termina con un simbolo de '!' o '?'

Una Interjección de atención (IAT) sigue las siguientes propiedades:

- Empieza con 'u' o 'e'
- a la primera le siguen 'f's 1+ seguido de 'ar'
- a las 'f's le sigue una IAT o una 'iag'.
- nada mas es una IAT

Una Interjección de Asombro(IAS) sigue las siguientes propiedades:

- Empieza con 'o'
- si es de longitud 2, la siguiente es 'h'
- si es de mayor longitud:
 - 'o' seguido de 'w' seguido de IAS seguido de 'a'
 - 'o' seguido de lat seguido de 'a'
- nada mas es una IAS

Ejemplos:

IATs:	ufar, efiag, uffffar, ufufufufar, ufeffffiag
No son IATs:	ufeff, efohar, ufar, uar, effffuar
IASs:	oH, owoHa, owowohaa, oufara, ouffffiaga, owoefaraa, oufufara
No son IASs:	owa, owoh, ufara, owowoha, ouiaga
IASTs:	ohufar!, oufarauffffiag?, owoefaraaufeffffiag!
No son IASTs:	ohufiago, ufaroh, owowaa

Input

Primera linea contiene un numero n ($0 < n < 1000$), representa la cantidad de cadenas a verificar en cada linea va una cadena de longitud entre 1 a 100 caracteres

Output

La salida es de n lineas, en cada linea se escribe **ES IAST** si la cadena cumple con las propiedades de una IAST, caso contrario escribe **NO IAST**.

Sample input	Output for the sample input
5	ES IAST
ohufar!	NO IAST
ohufiago?	NO IAST
ufaroh!	ES IAST
owoefaraaufeffffiag!	NO IAST
ohufar	

Problem F

Vacaciones

La entrada de datos es: la entrada standard,
la salida de datos es: la salida standard.

Geotopía es un bellissimo país, su geografía y ecosistema gozan de una diversidad incomparable, lo que lo convierte en un objetivo turístico muy popular para todo el mundo.

Pero, debido a su geografía, es complicado ir de un punto a otro, muchas veces los nativos deben cambiar de tipo de medio de transporte para poder llegar de un pueblo a otro.

Este país es muy ordenado, y tiene un registro de todos los medios de transporte habilitados entre cada par de pueblos, los medios de transporte son:

- Tren
- Bus
- Barco
- Lancha
- Planeador
- Helicóptero (pero ellos no usan acentos por lo que escriben: Helicoptero)
- Subterráneo (pero ellos no usan acentos por lo escriben: Subterraneo)

Por un principio histórico, a cada turista solo se le permite usar un único tipo de medio de transporte (cuantas veces lo desee).

Tu tarea, si decides aceptarla, es ayudar a los turistas a averiguar que medios de transporte podría utilizar en caso de viajar de un pueblo a otro sin importar cuantos pueblos intermedios visiten.

Entrada

La entrada contiene un conjunto de sets de datos, cada set comienza con una línea con tres enteros que son el número de pueblos P ($2 \leq P \leq 100$), la cantidad de caminos C ($P \leq C \leq \frac{n(n-1)}{2}$) y los posibles planes de viaje V ($1 \leq V \leq \frac{n(n-1)}{2}$). Seguidos por C líneas que indican cada una, el pueblo origen, el pueblo destino y todos los medios de transporte disponibles para ir del pueblo origen al pueblo destino. En la última línea se listan los posibles origen y destino para ser tomados en cuenta en las vacaciones.

Salida

Por cada set de datos se desplegarán dos líneas, en la primera se debe indicar el número de caso correspondiente “Caso #:”, y por cada plan de viaje V imprima una línea que contendrá el origen, destino y posibles medios de transportes ordenados alfabeticamente, en caso de no existir medios de transporte posibles, escriba la palabra “Imposible” como se muestra en los ejemplos de entrada y salida.

Sample input	Output for the sample input
<pre> 7 12 2 Antique Belin Tren Subterraneo Antique Carel Subterraneo Lancha Antique Duran Tren Planeador Belin Carel Subterraneo Planeador Belin Eefess Lancha Planeador Carel Duran Tren Subterraneo Carel Eefess Subterraneo Lancha Carel Goliar Tren Planeador Duran Goliar Subterraneo Planeador Eefess Goliar Lancha Planeador Eefess Falken Subterraneo Planeador Goliar Falken Tren Subterraneo Antique Falken Belin Goliar 3 2 1 Agmeth Bolin Tren Bolin Coner Bus Agmeth Coner 0 0 0 </pre>	<pre> Caso 1: De Antique a Falken puedes ir en: Planeador Subterraneo Tren De Belin a Goliar puedes ir en: Lancha Planeador Subterraneo Tren Caso 2: Imposible </pre>

Nota

Por cada línea con medios de transporte posibles de un viaje V se debe imprimir todo en la misma línea, la salida debe verse así:

```

Caso 1:
De Antique a Falken puedes ir en: Planeador Subterraneo Tren
De Belin a Goliar puedes ir en: Lancha Planeador Subterraneo Tren
Caso 2:
Imposible

```

Problem G

Pintando

La entrada de datos es: la entrada standard,
la salida de datos es: la salida standard.

Se necesita pintar una pared, para lo cual se necesita cubrir el borde del piso para evitar que las manchas lo ensucien. Para esto se tienen varias piezas rectangulares de papel, de las cuales se tiene solo los puntos donde se encuentran en el plano. Con esto se requiere calcular la longitud máxima de pared que se puede cubrir, teniendo en cuenta que se usaría el lado más largo del rectángulo.

Input

La primera línea contiene un entero $1 \leq n \leq 100$ que es la cantidad de piezas rectangulares. Las siguientes n líneas contienen 8 enteros, que son $0 \leq x_i, y_i \leq 100000$ representando cada vértice del rectángulo. La entrada termina con $n = 0$.

Output

Por cada caso de entrada se obtendrá un número de salida, que representa la mayor longitud a ser cubierta.

Sample input	Output for the sample input
1 0 0 0 3 4 0 4 3	4
2 0 0 0 3 4 0 4 3 0 0 0 12 5 0 5 12	16
0	