



Primer Concurso de Programación

Facultad de Ciencias/Club Pu++

23 de Octubre de 2015

Consideraciones Importantes

- Este conjunto contiene 8 problemas; las páginas se encuentran numeradas del 1 al 13 excluyendo esta portada. Verifica que tu copia esté completa.
- Tu programa debe leer los datos de la entrada estándar.
- El formato de la entrada se encuentra especificada en cada problema y puede asumir en forma segura que nunca se pasarán valores o formato no especificado.
- La salida de tu programa debe ser impresa en la salida estándar.
- Cada ejecución de tu solución será limitada por tiempo con el valor especificado en cada problema.
- Se considera una solución correcta, si el programa termina sin errores, produciendo la salida esperada, en el tiempo establecido, para todos los casos de prueba.
- Si tu programa genera un error o excede el límite de tiempo para algún caso de prueba, se recibirá una indicación del error (Time limit Exceeded, Wrong Answer, Runtime Error). La causa del error no será identificada.
- Para soluciones en C/C++, es importante que la ejecución termine con la instrucción "return 0" ya que esto le indica al SO que la ejecución termina sin errores. En java no es necesario lo anterior, pero es **indispensable** que la clase se llame Main y el archivo Main.java



A - La visión de Ana

Tiempo Límite: 1 Segundo

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Autor(es): Manuel Alcántara, Norma Verónica Trinidad

Ana siempre ha sido muy puntual pagando sus impuestos y este año tuvo un saldo a favor en su declaración anual, por lo que está pensando en qué gastará ese dinero. Una noche entre sus sueños, tuvo la visión de que el dólar sería una buena inversión y alcanzó a memorizar los precios de venta en los próximos días. Sin embargo, Ana no es muy buena con las matemáticas, por lo que te ha pedido ayuda para indicarle el día de compra que maximice el número de dólares que puede conseguir.

Entrada

La entrada comienza con un número entero T indicando el número de casos de prueba a seguir. Cada caso comienza con dos enteros S, N , separados por un espacio con $1 \leq S, N \leq 5 * 10^4$ que representa la cantidad de dinero que tiene Ana y el número de días de los cuales pudo memorizar el precio de venta de los dólares. Posteriormente en la siguiente línea se dan valores a_i con $0 < a_i \leq 100$, $1 \leq i \leq N$, con dos decimales y separados entre sí por un espacio, indicando el precio de venta del dólar en el día i .

Salida

Para cada caso de prueba, imprimir una línea con el formato “Case k : comprar en día d ” donde k es el número de caso empezando desde 1 y d el día en que Ana debe de comprar para obtener el mayor número de dólares. En caso de que existan varios días de compra, regresar el más próximo.

Ejemplos de Prueba

Entrada	Salida
2	Case 1: comprar en día 4
15 5	Case 2: comprar en día 1
14.50 14.20 17.80 13.11 15.21	
20 6	
8.90 8.92 10.11 9.27 8.90 9.57	



B - El acertijo del profesor

Tiempo Límite: 1 Segundo
Limite de la Pila: 10MB
Límite de la Memoria: 32MB
Autor(es): Manuel Alcántara

Un día en la clase de Timmy el profesor Crocker propuso el siguiente acertijo: *Dos amigos matemáticos se encuentran después de un largo periodo de no verse, y entablan la siguiente conversación:*

- *¡Cuánto tiempo sin verte! Cuéntame, ¿Ya te casaste?*
- *Sí, desde hace dos años.*
- *Excelente noticia, ¿Y tienes hijos?*
- *Tengo tres hermosas hijas.*
- *¿Qué edades tienen?*
- *Pues no te voy a decir sus edades, pero la multiplicación de ellas es 36 y la suma es el número de la casa de enfrente.*
- *(El amigo voltea a ver el número y después de 1 minuto contesta) Lo siento, pero me hace falta información.*
- *¡Oh! Tienes toda la razón, me faltó decirte que la más grande toca el piano. Timmy te ha pedido ayuda con un programa que encuentre todas las formas no equivalentes en que puede factorizarse un número utilizando tres términos.*

Entrada

La entrada inicia con un número entero $1 \leq N \leq 10^3$, que indica los casos de prueba a seguir. Posteriormente le siguen N líneas, cada una con un valor entero $1 \leq M \leq 10^5$, que representa el número a factorizar.

Salida

Para cada caso de prueba, imprimir una línea con formato “# M:” donde M es el número a factorizar utilizando tres términos, seguido de todas las combinaciones no equivalentes ordenando sus números en forma ascendente y posteriormente en forma lexicográfica. Utiliza para cada una de éstas el formato (x,y,z). Se considera que una combinación es equivalente si contiene los mismos números sin importar el orden en que aparecen, así (2,8,4) es equivalente a (8,2,4)



Ejemplos de Prueba

Ejemplo de Entrada	Ejemplo de Salida
2	# 3:
3	(1, 1, 3)
36	# 36:
	(1, 1, 36)
	(1, 2, 18)
	(1, 3, 12)
	(1, 4, 9)
	(1, 6, 6)
	(2, 2, 9)
	(2, 3, 6)
	(3, 3, 4)



C - Casilla para votar

Tiempo Límite: 1 Segundo

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Autor(es): Manuel Alcántara, Norma Verónica Trinidad

Las votaciones en Nlognoría están a punto de llevarse a cabo, sin embargo, los ciudadanos no saben con certeza en dónde se encuentra su casilla para votar. Debido a este problema, el CEN (Consejo Nacional Electoral) de Nlognoría, ha decidido intervenir y te ha contratado para que les ayudes en esta importante labor. Lo interesante en esto, es que desde su fundación, Nlognoría siempre ha sido un país bastante regular, en primer lugar, se conforma de un territorio rectangular de anchura N y altura M , el cual se encuentra dividido en $N \times M$ regiones cuadradas, en donde en cada una puede haber una montaña, un pastizal, un tramo de agua o una casa. A su vez, los habitantes se organizaron en pueblos, construyendo nuevas casas en forma ortogonal a las ya existentes, o aislándose para crear un nuevo pueblo. Debido a esto, dos casas se encuentran en el mismo pueblo solo si existe un camino de movimientos ortogonales compuesto únicamente por casas que las interconecta. Cada habitante conoce las coordenadas donde se encuentra su casa. Para este año el CEN ha decidido poner una casilla en cada pueblo y lo hará primero buscando la casa que se encuentra más abajo y después la más a la derecha de cada uno. Así que el programa que te han encomendado tiene como objetivo que los ciudadanos puedan ingresar las coordenadas de su casa y éste imprima las coordenadas de la casilla en donde le toca votar.

	1	2	3	4	5	6	7
1							
2							
3							
4							
5							

Entrada

La entrada comienza con dos números enteros N y M separados por un espacio con $1 \leq N, M \leq 10^2$, que indica la anchura y longitud que tiene Nlognoría, posteriormente le siguen N líneas, cada una con M enteros $0 \leq m_i \leq 3$ separados entre sí por un espacio,



en donde el valor 0 indica una porción de agua, el 1 una montaña, el 2 un pastizal y el 3 una casa. Seguido a esto se tiene un entero Q con $1 \leq Q \leq 10^4$ que indica el número de peticiones que se harán al programa, seguido de Q coordenadas enteras en formato $x y$ con $1 \leq x \leq N$ y $1 \leq y \leq M$ separadas por un espacio.

Salida

Para cada petición, se deberá imprimir una línea en formato (x, y) la cual representa las coordenadas de la casilla en la que le toca votar al ciudadano, o la palabra `montana`, `pastizal` o `agua` indicando del tipo de terreno si es que no existe una casa en ese lugar.

Ejemplos de Prueba

Ejemplo de Entrada	Ejemplo de Salida
5 7	(4,1)
3 3 2 3 0 3 3	(5,7)
1 3 2 0 0 3 3	agua
3 3 1 0 3 3 3	pastizal
3 2 3 0 0 0 2	montana
0 0 3 3 1 3 3	
5	
3 2	
5 7	
4 5	
4 2	
3 3	



D - Gráfica de Dulces

Tiempo Límite: 3 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Autor(es): Manuel Alcántara, Norma Verónica Trinidad

Debido a la nueva ley que prohíbe la venta de comida chatarra en las escuelas, la dulcera **MiBombón**, ha tenido ventas muy bajas últimamente. Así que para amortizar las pérdidas, ha decidido suspender la producción de algunos de sus dulces, sin embargo, la junta directiva aún no logra decidir qué dulces se dejarán de producir. Para tratar de dar solución a este problema, se han recopilado las ventas de los últimos 3 meses y te han contratado para crear un programa que muestre los resultados en una gráfica de barras, de esta manera se pueden visualizar los dulces que han tenido mayores ventas. Dado que la empresa fabrica muchos dulces, solo se quiere graficar aquellos productos que se hayan vendido al menos una vez.

Entrada

La primera línea consiste en un número entero M con $1 \leq M \leq 10^5$, que representa el número de productos vendidos en los últimos 3 meses. Posteriormente, le siguen M líneas cada una con un entero m_i con $1 \leq m_i \leq 100$ indicando el ID del dulce vendido. Puede suponer con seguridad que ningún dulce se vendió más de 1000 veces.

Salida

Sea $P = \{D_1, D_2, \dots, D_m\}$ el conjunto de los ID con al menos una venta, ordenado en forma ascendente, z el número de dígitos que tiene D_m , k el número de ventas del dulce más vendido, s el caracter espacio y u el caracter guión bajo ($_$).

$[c]_a^l$ indica que la cadena c debe imprimirse con una longitud l rellenando a la izquierda con caracteres a de ser necesario. De esta manera $[4]_0^3$ equivaldría a imprimir 004 y $[14]_s^4$ imprime \$\$14.

Con esto en mente, imprimir k líneas en formato:

$$[i]_0^k | [c_1]_s^z [c_2]_s^z \dots [c_m]_s^z$$

para i desde k hasta 1 y en donde c_j es un espacio si las ventas realizadas del producto con ID D_j son menores que i y un $*$ en caso contrario.

Posteriormente imprimir la línea:

$$[s]_s^{k+1} \underbrace{[u]_u^z [u]_u^z \dots [u]_u^z}_{m \text{ términos}}$$

Finalmente la última línea:

$$[s]_s^{k+1} [D_1]_0^z [D_2]_0^z \dots [D_m]_0^z$$

Notar que cada término a su vez se encuentra precedido por un espacio, excepto aquellos que se encuentran al inicio de cada línea.



Ejemplos de Prueba

Ejemplo de Entrada	Ejemplo de Salida
20	11 *
6	10 *
15	09 *
3	08 *
15	07 *
6	06 * *
6	05 * *
3	04 * *
15	03 * *
4	02 * * *
15	01 * * * *
15	
15	-- -- -- --
6	03 04 06 15
15	
15	
15	
6	
15	
15	
6	
15	
15	
6	

Ejemplo de Entrada	Ejemplo de Salida
6	3 *
5	2 * *
5	1 * * *
5	
1	- - -
2	1 2 5
2	



E - El programa misterioso

Tiempo Límite: 2 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Autor(es): Manuel Alcántara, Norma Verónica Trinidad

Programar es una de las más grandes pasiones para Alejandro, sin embargo, este semestre no le ha podido dedicar el tiempo que él quisiera, ya que esta cursando Combinatoria y no le ha resultado nada fácil. En su clase de hoy, la profesora mencionó que el número de formas en que se pueden alinear n personas en una fila equivalía a $n * (n - 1) * \dots * 2 * 1$ representada por la función factorial denotada por $n!$.

Pensando en que necesitaba estudiar más, Alejandro decidió ir a la biblioteca para consultar algunos libros acerca del tema. Hojeando uno de ellos, cayó al piso una hoja, la cual levanto sin mucho apuro y vió escrito en ella el siguiente mensaje:

```
foo(M,N):  
k:=0  
mientras N % M es igual a 0  
    N:=N/M  
    k:=k+1  
regresar k
```

Decidido en que el destino le intentaba decir algo, inmediatamente fue a la sala de estudio y lo programó, sin saber realmente que es lo hacía aquel misterioso programa. Después de algunas ejecuciones se percató de que si $N < M$ la función siempre regresa 0, así que decidió probar para $N!$, al fin y al cabo es un tema que había visto recientemente. Sin embargo, su programa genera un error, ya que $N!$ es un número muy grande. Invasado por la intriga y sabiendo que no existe cosa que no puedas lograr, Alejandro te ha pedido ayuda para calcular el valor que regresaría la función $foo(M, N!)$ para valores grandes de N y así por fin resolver de una vez este misterio que le tiene dando vueltas la cabeza.

Entrada

La primera línea contiene de entero $1 \leq T \leq 10^4$ que representa el número de casos de prueba a seguir. Posteriormente las siguientes T líneas se componen de dos números enteros M, N con $2 \leq M, N \leq 2^{31} - 1$ separados por un espacio.

Salida

Imprimir una línea en formato $foo(M, N!) = k$ en donde M y N son los números pasados como parámetro y k es el resultado que tendría que devolver la función foo , aplicada a esos valores.



Ejemplo de Prueba

Ejemplo de Entrada	Ejemplo de Salida
5 12 49 2 120 48 11 2147483646 2147483647 7 5	foo(12,49!) = 22 foo(2, 120!) = 116 foo(48,11!) = 2 foo(2147483646,2147483647!) = 6507525 foo(7,5!) = 0



F - Sumando Dígitos

Tiempo Límite: 2 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Autor(es): Manuel Alcántara, Norma Verónica Trinidad

AnaLu apenas va en 1° de primaria y esta aprendiendo a sumar, pero ella es realmente buena, ya que hasta el momento ninguna suma ha sido un desafío para ella. Con el único propósito de molestar a su hermana (como es probable que lo hagan todos los hermanos) Luis le ha dado un número muy grande para que sume todos sus dígitos, sin embargo, no pudo hacer que AnaLu se equivocara. Lleno de cólera y frustrado por no poder vencer a su hermana, se le vino a la mente la siguiente idea macabra, dados dos números N y M bastante grandes sumar todos los dígitos de los números en ese intervalo. ¹

Agotada por hacer tantas sumas y a punto de darse por vencida, AnaLu recordó que en algún momento tú le comentaste que las computadoras podían hacer muchísimas sumas por segundo, así que sin dudarlo te ha contactado para que le ayudes en esta tarea y así poder hacer que su hermano Luis deje de molestarla.

Entrada

La primera línea consiste en un número entero T indicando los casos de prueba a seguir. Posteriormente le siguen T líneas, conformadas por dos enteros N y M separados por un espacio, en donde $1 \leq N \leq M \leq 10^6$, los cuales representan el intervalo de los números a sumar sus dígitos.


Salida

Imprimir una línea en formato **Case #k = s** donde k es el número del caso de prueba empezando en 1 y s es la suma de los dígitos de los números comprendidos en el intervalo cerrado $[N, M]$.

Ejemplo de Prueba

Ejemplo de Entrada	Ejemplo de Salida
3	Case #1 = 46
1 10	Case #2 = 99
5 22	Case #3 = 3205
9 328	

¹Para el primer caso de prueba $46 = 1 + 2 + \dots + 9 + (1 + 0)$

 **G - 2048**

Tiempo Límite: 1 Segundo
Limite de la Pila: 10MB
Límite de la Memoria: 32MB
Autor(es): Manuel Alcántara

2048 es un juego que consiste en mover casillas, las cuales contienen un número que es potencia de dos, hasta conseguir que una de ellas sume 2048. El tablero consiste de una malla de 4×4 con una configuración inicial y cuyos movimientos posibles son: Izquierda, Derecha, Arriba, Abajo. Cada vez que se realiza un movimiento, todas las casillas se desplazan en esa dirección todo lo que sea posible y si dos casillas con el mismo número quedan juntas, se sumarán. Finalmente, después de realizar el movimiento, aparece una nueva casilla con el número 2 de forma aleatoria, siempre que se pueda.

El reto de este problema consiste en mostrar el estado final del juego dada la posición inicial y la serie de movimientos realizados, suponiendo que la nueva casilla que se agregará después de cada movimiento, será la primera casilla disponible (si es que la hubiera) recorriendo la malla de izquierda a derecha y de arriba hacia abajo. Si una casilla suma más de 2048 entonces ésta debe conservar el mismo valor.

2			
64	4	2	
16	128	16	
2	32	4	

Entrada

Las primeras 4 líneas contienen 4 enteros separados por un espacio y que representan el estado inicial del juego. Cada valor es una potencia de dos menor o igual a 2048 o un 0 si es que no hay una casilla en ese lugar. Posteriormente se da un entero entero M , con $1 \leq M \leq 10^5$ que indica la cantidad de movimientos que seguirán. Las próximas M líneas contienen alguna de las siguientes palabras: **Arriba**, **Abajo**, **Izquierda**, **Derecha** indicando el movimiento a realizar.

Salida

Imprimir 4 líneas cada una con 4 enteros separados por un espacio, que representan el estado final del juego 2048 después de haber realizado los movimientos partiendo de la configuración inicial.



Ejemplo de Prueba

Ejemplo de Entrada	Ejemplo de Salida
0 2 0 4	2 4 2 0
4 4 4 4	8 8 0 0
32 16 16 2	32 32 2 0
0 8 0 8	16 0 0 0
1	
Izquierda	

Ejemplo de Entrada	Ejemplo de Salida
0 2 0 4	2 2 4 2
4 4 4 4	0 0 0 16
32 16 16 2	0 0 64 2
0 8 0 8	0 0 0 16
2	
Izquierda	
Derecha	



H - El hijo perdido Anacleto

Tiempo Límite: 1 Segundo

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Autor(es): Manuel Alcántara, Armando Ballinas

Un empresario acaudalado llamado Anacleto, cree que ha encontrado a su hijo Lucas que le fue arrebatado al nacer. Anacleto ha declarado que si se reconoce su hijo le heredará toda su fortuna, pero como no está seguro si Lucas es su hijo perdido o no, te ha pedido ayuda para averiguarlo. Anacleto ha logrado conseguir una muestra de ADN tanto suya como del supuesto Lucas. Estas secuencias constan de las bases nitrogenadas que componen el ADN: Adenina (A), Guanina (G), Timina (T) y Citocina (C).

Anacleto quiere saber cual es el mínimo número de bases nitrogenadas que hay que eliminar de la muestra de Lucas para que ésta este completamente contenida en su propia muestra, en el mismo orden pero no necesariamente de forma contigua. Si este valor es a lo más un 15% entonces Anacleto reconocerá a Lucas como su legitimo hijo.

Entrada

La entrada del programa comienza con un número entero T , que indica el número de casos de prueba a seguir. Posteriormente, cada caso consta de dos líneas del mismo tamaño, en donde la primera contiene la muestra de ADN de Anacleto y la segunda la de Lucas. Cada muestra se representa como una cadena de caracteres compuesta solo por las letras C, A, G y T. Puedes suponer que las muestras tienen una longitud de a lo más 1000 bases nitrogenadas.

Salida

Para cada caso de prueba imprimir la leyenda `Yo soy tu padre!` si el mínimo número de caracteres que se tienen que eliminar de la muestra de Lucas para que este contenida en la de Anacleto es a lo más un 15%. De lo contrario imprimir el mensaje `No eres mi hijo.`

Ejemplo de Prueba

Ejemplo de Entrada	Ejemplo de Salida
2 AATGCG ATGCTG ATGCTGCA ATGCGCAG	No eres mi hijo. Yo soy tu padre!