TecnoMate 2018
COMPETENCIA DE PROGRAMACIÓN

Nivel “Secundaria”
Problemas

19 de Octubre de 2018

Problema 1: Celulares

Autor: Luis Santiago Re

Dentro de poco tiempo, Pascual y sus amigos se irán de campamento. El bosque en el que planean acampar es muy hermoso y van a poder realizar muchas actividades al aire libre, sin embargo no van a tener energía eléctrica y entonces no podrán cargar sus celulares; lo que es un grave problema, ya que como casi todos hoy en día, los chicos no pueden vivir sin el celular. Para solucionar el problema, Pascual quiere llevar la menor cantidad posible de cargadores portátiles que les permita usar los celulares durante todo el viaje.

Ninguno de los chicos está muy seguro de cómo elegir los cargadores portátiles para no llevar de más, pero ellos saben que vos sos muy inteligente y por eso te pidieron que los ayudes con este problema. Para facilitarte un poco el trabajo, Pascual armó una lista con los cargadores que tienen disponibles para llevar y además calculó la cantidad de energía eléctrica que van a necesitar para todo el viaje.

Entrada

La primera línea de la entrada contiene dos enteros **N** $(1 \leq N \leq 2000)$ y **K** $(1 \leq K \leq 10^9)$ indicando la cantidad de cargadores que disponen y la cantidad total de energía que necesitarán en el viaje, respectivamente.

La siguiente línea contiene **N** enteros C_i $(1 \leq C_i \leq 10^9)$, la capacidad **C** del **i**-ésimo cargador.

Salida

Imprima un entero indicando la mínima cantidad de cargadores necesaria para que Pascual y todos sus amigos puedan usar sus celulares en el campamento. Si ni siquiera todos los cargadores alcanzan, imprima **-1**.

Ejemplos

Entrada	Salida
5 10 1 2 3 4 5	3
5 10 1 2 10 4 5	1
5 6 1 1 1 1 1	-1

Problema 2: Juego con piedras

Autor: Luis Santiago Re

Eric y Matias están jugando con una pila de **N** piedras. En su turno, cada uno de ellos debe sacar entre **1** y **K** piedras de la pila. El que saque la última piedra será el ganador.

Se sabe que ambos son muy inteligentes y por lo tanto siempre juegan de forma óptima. Además siempre empieza jugando Eric.

¿Quién será el ganador?

Entrada

La entrada consiste de una única línea con dos enteros **N** $(1 \le N \le 10^9)$ y **K** $(1 \le K \le 10^9)$, que indican respectivamente la cantidad inicial de piedras en la pila y el número máximo de piedras que se puede sacar en un turno.

Salida

Imprimir el nombre del ganador.

Ejemplo

Entrada	Salida
`1 2`	`Eric`
`4 3`	`Matias`
`6 3`	`Eric`
`20 4`	`Matias`

Explicación

Caso 1: Eric comienza el juego, saca la única piedra de la pila y gana.

Caso 2: Eric puede sacar 1, 2 o 3 piedras en el primer turno. Sin importar cuantas saque, cuando sea el turno de Matias habrá 3 piedras o menos y podrá sacarlas todas, resultando ganador.

Caso 3: la jugada óptima de Eric es quitar 2 piedras en el primer turno para que Matias caiga en la situación perdedora explicada en el caso 2.

Problema 3: Armar los equipos

Autor: Luis Santiago Re

A Nicolás y sus amigos les encanta jugar al fútbol 5. Tienen reservada una cancha para todos los viernes a la tarde, donde se juntan, arman dos equipos de 5 jugadores cada uno y juegan un partido. Algunos partidos resultan más divertidos que otros, y claramente esto depende de cómo se armen los equipos. Mientras más parejos sean, más divertido será el partido.

Nicolás quiere que lo ayudes a armar los equipos de la forma más pareja posible, para que el próximo partido sea súper divertido.

Se sabe que cada uno de los chicos tiene un nivel de habilidad determinado, y que el valor de cada equipo será la suma de las habilidades de sus jugadores. Entonces, para armar los equipos lo más parejo posible, hay que minimizar la diferencia entre los valores de cada equipo.

Entrada

La entrada consiste de una única línea con **10** enteros $\mathbf{h_i}$ $\mathbf{(1 \leq h_i \leq 10^8)}$, la habilidad **h** del **i**-ésimo jugador.

Salida

Imprima la diferencia entre los valores de cada equipo que resulta de armar los equipos de la forma más pareja posible. Notar que este número no debe ser negativo.

Ejemplo

Entrada	Salida
`1 2 3 4 5 6 7 8 9 10`	`1`
`10 2 4 11 5 8 7 17 13 5`	`0`
`31 5 10 90 1 1 8 12 10 9`	`33`

Problema 4: Adivinar el palíndromo

Autor: Luis Santiago Re

A Federico le gusta mucho resolver problemas de programación y de matemática. Particularmente le gustan los problemas sobre números capicúas, o también conocidos como números palíndromos. Sin embargo necesita de tu ayuda para poder resolver el siguiente:

Dado un par de valores enteros positivos **K** y **N**, consideramos el conjunto de números naturales **C**, que contiene a todos los palíndromos de exactamente **K** cifras, ordenados de menor a mayor. La pregunta es: ¿Cuál es el valor del **N**-ésimo número del conjunto **C**?

Por ejemplo, si **K=2**, los números que pertenecen a **C** son: **11, 22, 33, 44, 55, 66, 77, 88, 99**.

Entrada

La entrada consiste de dos enteros, **K** $(1 \leq K \leq 9)$ y **N** $(1 \leq N \leq 10^9)$.

Salida

Imprimir el valor del **N**-ésimo número del conjunto **C**, definido en el enunciado. Si el conjunto **C** contiene menos de **N** números, imprimir **-1**.

Ejemplo

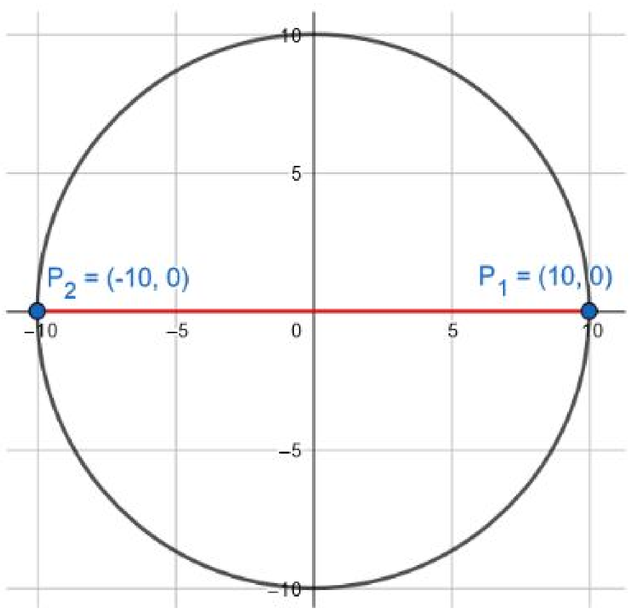
Entrada	**Salida**
`2 1`	`11`
`2 5`	`55`
`2 50`	`-1`
`3 1`	`101`

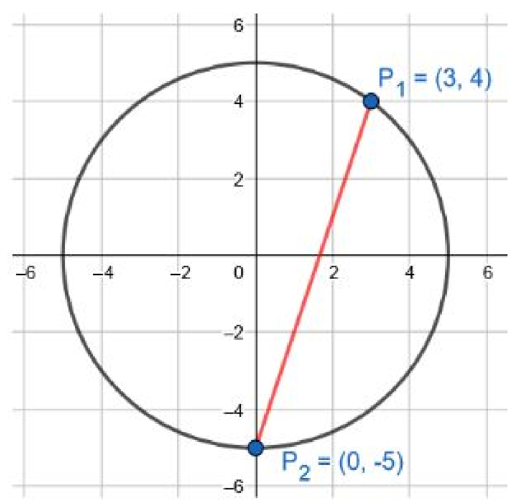
Problema 5: La última galletita (ó masita?)

Autor: Luis Santiago Re

A Julián le encantan las galletitas dulces. Según él, todo momento es un buen momento para comer cualquier tipo de galletitas dulces. Las come en el desayuno, en la merienda y casi siempre de postre en el almuerzo y la cena. A pesar de su obsesión con las galletitas, Julián es un chico muy generoso y solidario que siempre le ofrece a sus amigos cuando él está comiendo y le convida a cualquiera que le pida.

Muchas veces le pasa que le piden justo cuando le queda sólo una, y como a todos, a él no le gusta dar la última del paquete, por lo que en estos casos siempre decide partir la galletita en dos partes, para comer una y darle la otra a su amigo. Partir la galletita en dos partes exactamente iguales no es tarea fácil, por eso cuando no lo logra, y uno de los pedazos resulta más grande que el otro, Julián siempre intenta darle el pedazo más grande a su amigo. Sin embargo tampoco es fácil determinar a simple vista cual de las partes es la más grande, y después de compartir esa última galletita siempre se queda preocupado, pensando si logró partirla exactamente a la mitad, o si no lo logró y tal vez le dio a su amigo el pedazo más pequeño.





Si representamos la galletita como un círculo en el plano cartesiano con centro en el origen **(x=0, y=0)**, y representamos al corte como un segmento interno al círculo cuyos extremos son dos puntos **P_1 (x_1, y_1)** y **P_2 (x_2, y_2)**, que pertenecen al borde de la galletita. ¿Julián se puede quedar tranquilo de que los dos pedazos eran de igual tamaño?

Entrada

La entrada consiste de **4** enteros x_1, y_1, x_2, y_2 **($-1000 \leq x_1, y_1, x_2, y_2 \leq 1000$)**, que representan al corte realizado en la galletita.

Se asegura que los puntos **P_1 (x_1, y_1)** y **P_2 (x_2, y_2)** son puntos distintos y pertenecen a una circunferencia con centro en **(0, 0)**.

Salida

Imprimir **Iguales** si los dos pedazos resultantes son de igual tamaño. En caso contrario imprimir **Distintos**.

Ejemplo

Entrada	Salida
`10 0 -10 0`	`Iguales`
`3 4 0 -5`	`Distintos`

Problema 6: Peaje

Autor: Luis Santiago Re

Bruno estudia ingeniería en la ciudad de Santa Fe, pero vive en Paraná. Por eso todos los días de la semana viaja para poder ir a cursar. Casi siempre va en colectivo, pero otras veces tiene la suerte de que a él o a alguno de sus amigos le prestan el auto y entonces se organizan para ir todos juntos.

Ir en auto suele ser más cómodo porque demoran mucho menos tiempo. Además dividen los gastos y terminan gastando aproximadamente la misma cantidad de dinero que cuando viajan en colectivo.

Cuando viajan en colectivo, los chicos se sorprenden del poco tiempo que demoran en pasar el peaje, porque siempre que van en auto terminan en la cola más lenta y demoran mucho tiempo allí. Bruno sabe que los colectiveros usan algún programa que les dice cual es la cola más rápida en el momento que llegan al peaje, pero cuando quiso pedirles una copia del mismo a los colectiveros, no se la quisieron dar, y por eso necesita tu ayuda para desarrollar una herramienta similar.

Se sabe que el peaje del túnel subfluvial tiene exactamente 4 cabinas, y por lo tanto 4 colas, numeradas de 1 a 4, que están activas todo el tiempo. Para facilitar el problema, Bruno propuso clasificar los autos en N tipos, según cuantos segundos demoran en pagar el peaje cuando llega su turno. Además se asume que cuando un auto termina de pagar, instantáneamente avanza y comienza a ser atendido el siguiente en la cola.

Dados los autos que hay en cada cola en un momento, decidir en cuál conviene colocarse.

Entrada

La primer línea contiene un entero **N** $(1 \leq N \leq 10^5)$, indicando la cantidad de tipos de auto que existen. Cada uno de estos tipos se identifican con un entero entre **1** y **N**, inclusive. La siguiente línea contiene **N** enteros t_i $(1 \leq t_i \leq 45678)$, el tiempo que demoran en pagar el peaje los autos del tipo **i**.

Siguen 4 líneas describiendo las colas del peaje, donde la **j**-ésima de estas describe a la cola número **j**. Cada una de estas líneas comienza con un entero **C** $(1 \leq C \leq 45678)$ que indica la cantidad de autos en la cola correspondiente, y le siguen **C** enteros k_i $(1 \leq k_i \leq N)$, indicando el tipo de auto que se encuentra en la posición **i** de la cola correspondiente.

Salida

Imprimir el número de la cola en la que hay que colocarse para minimizar el tiempo de espera en ser atendido, seguido de este tiempo que se deberá esperar en dicha cola. Si hay más de una cola con mínima demora, selección la de menor número.

Ejemplo

Entrada	Salida
5 10 1 2 5 15 2 1 5 10 2 3 3 2 2 3 2 2 4 2 2 1 1 4 1 2 3 4	2 17
2 1 2 2 2 1 2 2 2 2 1 1 1 2	3 2
2 1 2 1 2 2 2 1 2 1 1 1 2	1 2
4 1 3 2 1 1 3 1 2 1 4 1 1	3 1

Explicación

La demora en cada cola es:

- Caso 1: 25, 17, 20, 18
- Caso 2: 3, 4, 2, 2
- Caso 3: 2, 3, 2, 2
- Caso 4: 2, 3, 1, 1

Problema 7: Comprimir matrices

Autor: Luis Santiago Re

En TecnoMate 2017 hubo un problema que pedía expandir una matriz de caracteres. En pocas palabras este proceso de expansión se definía por dos enteros **R** y **C**. Entonces la matriz expandida resulta de repetir **R** veces cada fila y **C** veces cada columna de la matriz inicial.

Por ejemplo, si **R=2**, **C=3** y la matriz inicial es la siguiente:

```
ABC
DEF
```

La matriz expandida resultante será:

```
AAABBBCCC
AAABBBCCC
DDDEEEFFF
DDDEEEFFF
```

Notar que los valores válidos para **R** y **C** son todos los enteros positivos, sin importar cual es la matriz inicial.

La cuestión es que ahora necesitamos comprimir matrices, de tal forma que si a la matriz comprimida le aplicaríamos el proceso de expansión explicado, obtendremos la matriz que teníamos antes de comprimirla.

Entonces, una compresión queda definida por los enteros **R** y **C**, de tal forma que si a la matriz comprimida la expandimos con estos valores, obtenemos la matriz inicial (como era antes de ser comprimida).

Dada una matriz, necesitamos saber cuantas compresiones válidas diferentes tiene.

Entrada

La primer línea contiene dos enteros **N (1 ≤ N ≤ 100)** y **M (1 ≤ M ≤ 100)** indicando las dimensiones de la matriz. Las siguientes **N** líneas contienen **M** caracteres cada una, describiendo la matriz. Cada uno de los caracteres será una letra mayúscula del abecedario inglés.

Salida

Imprima la cantidad de formas distintas en que se puede comprimir la matriz dada.

Ejemplo

Entrada	Salida
4 9 AAABBBCCC AAABBBCCC DDDEEEFFF DDDEEEFFF	4
4 6 AAAAAA AAAAAA AAAAAA AAAAAA	12

Explicación

Las formas en que se puede comprimir la matriz en el 1er caso son las siguientes:

- R=1, C=1

```
AAABBBCCC
AAABBBCCC
DDDEEEFFF
DDDEEEFFF
```

- R=1, C=3

```
ABC
ABC
DEF
DEF
```

- R=2, C=1

```
AAABBBCCC
DDDEEEFFF
```

- R=2, C=3

```
ABC
DEF
```

Problema 8: Botellero

Autor: Daniel Ambort

A Raúl, el papá de Maximiliano le encanta tomar buenos vinos (durante el fin de semana, porque no tiene que manejar). En el quincho de su casa (su lugar en el mundo los fines de semana) tiene un botellero que luce así:

B	B				
			B	B	
	B			B	
			B		B
		B			B
B		B			

Es una estantería que permite almacenar hasta 36 botellas. Raúl tiene además un TOC (Trastorno Obsesivo Compulsivo) fuerte, y una de sus manifestaciones es la siguiente: En el botellero, los días que vienen sus amigos, sólo se pueden colocar 12 botellas distribuídas de tal forma que: haya dos en cada fila, dos en cada columna, dos en la diagonal principal y dos en la secundaria. La imagen que se vé a la izquierda representa una configuración que cumple con el TOC de Raúl. Las 'B' indican la ubicación de las botellas.

Maximiliano estudia Ingeniería y le gusta programar, pero es muy vago…, algunas veces su padre le pide una configuración nueva para las 12 botellas, que siga respetando las restricciones que le dicta su TOC. Maximiliano estuvo investigando, descubrió que las computadoras son realmente rápidas, y además descubrió el generador de números seudoaleatorios de C++, y como es muy creativo, diseñó el siguiente programa:

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

void MuestraMatriz(int matriz[][6]);
bool CumpleTOC(int matriz[][6]);

void CargaBotellas(int matriz[][6]){
     // 0: vacio, 1: botella
     for(int i=0; i<6; i++)
        for(int j=0; j<6; j++)
              matriz[i][j]=0;

     for(int i=1; i<=12; i++){
        int aux= rand()%36;
        matriz[aux/6][aux%6]=1; }
}
```

```
…
…


int main(int argc, char *argv[]){
    int matriz[6][6];
    srand(time(NULL));
    do
       CargaBotellas(matriz);
    while (!CumpleTOC(matriz));

    MuestraMatriz(matriz);
    return 0;
}
```

Maximiliano sabe que su programa genera una tras otra configuraciones con 12 botellas en el botellero y te pidió ayuda: necesita que codifiqués un programa que realice la tarea de la función CumpleTOC() de su código.

Entrada

La entrada consiste de un número entero T (1 <= T <= 100) indicando los casos de prueba, y luego para cada caso, 6 líneas con 6 números enteros (0 ó 1) que indican el contenido del botellero (generado por la función CargaBotellas()).

Salida

Su programa debe imprimir una única línea para cada caso de prueba, indicando "TOC" si el botellero cumple con las restricciones de Raúl, y "CUAC" en caso contrario.

Ejemplos

Entrada	Salida
2 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 1 0 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1	CUAC TOC

Problema 9: Números de Eudoxus V1.0

Autor: Daniel Ambort (Adaptación problema del libro “Ejercicios creativos y recreativos en C++”)

Eudoxus (408-355 a.C) como era habitual en la época griega, compaginó la filosofía, la astronomía y las matemáticas. Dicen que fué el primer griego en hacer un mapa de las estrellas. Sus trabajos matemáticos tuvieron repercusión en todo el mundo de aquella época (aunque no tenía un canal de YouTube con miles de suscriptores... :). El libro V de los Elementos de Euclides, está basado en sus descubrimientos sobre las proporciones.

A los 23 años asistió a varias conferencias en Atenas, posiblemente en la Academia de Platón (abierta c. 387 a. C.). En estos días seguramente hubiera seguido varios cursos online, y Platón hubiera tenido un canal de YouTube con miles de seguidores, pero mejor no nos distraigamos...

Los números de Eudoxus se definen como sigue:

$$X_r = Y_r + Y_{r-1} \qquad \text{si } r >= 1$$
$$Y_r = X_{r-1} + Y_{r-1} \qquad \text{si } r >=1$$
$$X_0 = 1$$
$$Y_0 = 0$$

Input:
La primera línea de entrada contiene un entero T (1 <= T <= 10) con el número de casos de prueba. Luego, viene una línea para cada caso, con un número entero N (1 <= N <= 20).

Output:
Para cada número entero N, Ud. debe mostrar en una línea los valores Xn e Yn de los números de Eudoxus, separados por un espacio.

Ejemplo

Entrada	Salida
3 2 4 5	3 2 17 12 41 29

Problema 10: Desperdiciando Tiempo

Autor: Pablo A. Heiber

Cierta pareja de jóvenes suele aprovechar el tiempo al máximo. Esto produce que se sobrecarguen de estrés, por lo que decidieron “desperdiciar” un poco de tiempo viendo su serie de televisión favorita.

La serie tiene N temporadas. Cada temporada tiene una cantidad de cap´ıtulos que puede variar dependiendo de su éxito, de la disponibilidad de los actores, del equipo de producción y de otros factores externos. Todos los capítulos tienen una duración fija de M minutos.

Para no perder el hilo, antes del comienzo de cada temporada la pareja vuelve a ver compulsivamente todos los capítulos de todas las temporadas anteriores. La preocupación que tienen ahora es si van a desperdiciar demasiado tiempo con este hobby que deber´ıa tranquilizarlos. Necesitan la ayuda de ustedes para no volver a caer en el estrés.

Input:
Cada caso de prueba se describe utilizando dos líneas. La primera línea contiene dos enteros N y M que indican respectivamente la cantidad de temporadas de la serie y la duración en minutos de cada capítulo ($1 \leq N \leq 105$, $1 \leq M \leq 106$). La segunda línea contiene N enteros Ci que representan la cantidad de capítulos de las sucesivas temporadas en el orden en que son emitidas ($1 \leq Ci \leq 100$ para $1 \leq i \leq N$). El final de la entrada se indica con una línea que contiene dos veces el número −1.

Output:
Para cada caso de prueba, imprimir en la salida una línea conteniendo un entero que representa la cantidad de minutos que va a utilizar la pareja para ver la serie completa.

Ejemplo

Entrada	**Salida**
6 20 24 23 15 22 24 17 1 100 100 10 1000000 99 99 99 99 99 99 99 99 99 99 -1 -1	9000 10000 5445000000