



2º Concurso de Programación Facultad de Ciencias/Club Pu++ 14 de Octubre de 2016

Consideraciones Importantes

- Al iniciar el concurso se te dará un conjunto de hojas que describen 10 problemas a resolver; las páginas se encuentran numeradas del 1 al 21. Antes de iniciar, verifica que tu copia esté completa, de lo contrario reporta las hojas faltantes al grupo de voluntarios.
- Inmediatamente ingresa a la página del Juez, ve a la sección de “perfil” y cambia el nombre por el de tu equipo.
- El objetivo del concurso es acumular más puntos que todos los demás participantes. En caso de empate, se toma en cuenta la suma de los penalties (tiempo desde que inició el concurso hasta que enviaste la solución).
- Jamás se te quitarán puntos por intentar resolver un problema: si ya habías enviado una solución con x puntos, únicamente se cambiará tu puntuación y penalty si obtienes más de x puntos.
- Toda la entrada y salida se hace mediante entrada estándar (“teclado” y “consola”).
- Imprime **exactamente** lo que te pide el problema en el formato especificado, ni más ni menos. Por lo tanto, **NO** agregues mensajes del tipo “Dame un número”, “Introduce un número” o similar.
- El formato de entrada se encuentra especificado en cada problema y se puede suponer de forma segura que nunca se pasarán valores fuera del rango o formato especificado.
- Utilizar solo bibliotecas estándar.
- Cada ejecución de tu solución será limitada por tiempo, con el valor especificado en cada problema.

- Se considera una solución 100 % correcta, si el programa termina sin errores, produciendo la salida esperada, en el tiempo establecido, para todos los casos de prueba. Cada problema puede aportar un porcentaje parcial a la puntuación de acuerdo al número de casos de prueba en los que se tuvo éxito.
- Cada envío incorrecto de tu programa agregará 10 minutos de penalización a la suma global de tiempos en el equipo. Recuerda que este parámetro sirve para desempatar en el caso que se tengan la misma cantidad de problemas resueltos.
- 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, 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`
- En ocasiones, leer la entrada con `cin` o `Scanner` puede dar `TimeLimit` debido a que éstos son muy lentos. Si es el caso, intenta probar con alternativas más rápidas como `printf` y `BufferedReader`.
- El orden en que aparecen los problemas no dice nada de su complejidad, por lo que no siempre el primer problema es el más fácil de resolver, así que es bueno leer todos y cada uno de los problemas.
- No se te olvide imprimir un salto de línea al final de cada impresión.
- Por último queremos agradecer el apoyo y el patrocinio de los premios a:



Usuario	Contraseña
<hr/>	<hr/>

Problema Ejemplo

Tiempo Límite: 1 Segundo

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

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

Entrada

La primera línea es un entero T , con $1 \leq T \leq 10^5$ que indica el número de casos de prueba a seguir. En las siguientes T líneas, aparecerán 2 enteros A y B , con $1 \leq A, B \leq 10^3$ separados por un espacio.

Salida

Para cada caso imprimir la suma de los enteros A y B .

Entrada Ejemplo	Salida Ejemplo
2	5
2 3	13
5 8	

Soluciones al Problema Ejemplo

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;
}
```

■ Ruby 1.9

```
T = gets.to_i

T.times{
  A, B = gets.split.map(&:to_i)
  puts (A+B)
}
```

■ Python 2.7

```
T = int(raw_input())

for i in range(0,T):
    A, B = map(int, raw_input().split())
    print A+B
```

■ Java 7

```
import java.util.Scanner;
class Main{
    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);
        }
    }
}
```

■ Haskell GHC 7.6

```
res :: IO ()
res = do
    ints <- fmap (map read.words) getLine
    print (ints!!0 + ints!!1)

main :: IO ()
main = do line <- getLine
    let T = read line
        let ops = replicate T res
        foldl1 (>>) ops
```



A - Mezclando Colores

Tiempo Límite: 2 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

Después de un largo periodo de escuela, Norman por fin ha salido de vacaciones y decidió tomar clases de pintura al óleo para distraerse un rato y liberar un poco su estrés. Entre las cosas que compró para su primer día de clases, se encuentran un bastidor, varios pinceles y un conjunto de n pinturas distintas. Cuando empezó a pintar su cuadro, se dió cuenta de que siempre que mezclaba dos pinturas distintas obtenía un nuevo color diferente al de todos los demás, por lo que se ha preguntado ¿Cuántos colores diferentes se pueden formar con las n pinturas iniciales?



Entrada

La entrada comienza con un número entero T , con $1 \leq T \leq 10^4$ indicando el número de casos de prueba a seguir. Posteriormente le siguen T líneas, cada una con un número entero n con $1 \leq n \leq 10^4$, que representa el número de pinturas diferentes que Norman compró inicialmente.

Salida

Para cada caso de prueba, imprimir un número entero que representa el número de colores nuevos que Norman puede generar con el conjunto de pinturas iniciales.

Entrada Ejemplo	Salida Ejemplo
3	1
2	15
6	55
11	



B - Tabla para Pastel

Tiempo Límite: 1 Segundo

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

Se acerca la fiesta de cumpleaños de la pareja de Norman, pero como actualmente no tiene mucho dinero (porque lo gastó en los materiales para su curso de pintura), pensó que un buen regalo sería hacerle un pastel circular puesto que contaba con un molde de esa figura. Después de tres largas horas en la cocina terminó haciendo un pastel delicioso y decidió montarlo sobre una tabla de madera que serviría para transportarlo. Sin embargo, luego de buscar por toda su casa solo pudo hallar una tabla cuadrada cuyas esquinas coincidían exactamente con la circunferencia del pastel, tal como muestra el cuadrado blanco sobre la figura 2. Debido a que la tabla era muy pequeña para montar el pastel, decidió ir a la maderería a comprar otra. Estando ahí, el encargado le preguntó de que área necesitaba la tabla, ya que dependiendo de eso, era el precio que se tenía que pagar. Norman conoce el área de la tabla que encontró, pero no sabe como calcular el área de la tabla que contenga completamente al pastel y que sea de área mínima para no pagar mucho. ¿Podrías ayudarle a Norman a resolver este problema?

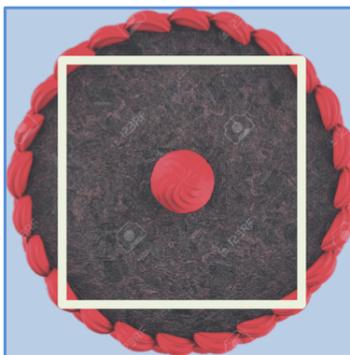


Figura 2: El cuadrado marcado en blanco representa la tabla que encontró Norman. El cuadrado azul representa la tabla que necesitaría comprar para transportar el pastel.

Entrada

La entrada comienza con un número entero T , con $1 \leq T \leq 10^4$ indicando el número de casos de prueba a seguir. Posteriormente le siguen T líneas, cada una con un número entero n con $1 \leq n \leq 10^8$ que representa el área de la tabla cuadrada que encontró Norman en su casa.



Salida

Para cada caso de prueba, imprimir una línea con formato: `Area a comprar: d`, donde `d` es un entero que representa el área mínima de la tabla que contiene al pastel.

Entrada Ejemplo

2
3
5

Salida Ejemplo

Area a comprar: 6
Area a comprar: 10



C - 7 Maravillas

Tiempo Límite: 1 Segundo

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

Después de cantar las mañanitas, soplarle a las velas y degustar del delicioso pastel que Norman hizo, los invitados en la fiesta decidieron divertirse con un juego de mesa llamado *7 Maravillas*, el cual se basa en cartas y es muy entretenido, ya que se pueden hacer tanto alianzas como enemistades. Al final del juego, el jugador que tenga más puntos de victoria se considera el ganador. En el juego existen unas cartas especiales de poder con un cierto tipo que otorgan puntos extra al final de la partida. Sin embargo, la regla para puntuar dichas cartas es algo complicada y menciona lo siguiente:

1. Cada jugador separa por tipo todas las cartas de poder que tenga en su mano formando montículos. Cada montículo, otorga puntos de victoria equivalente al cuadrado de las cartas que contenga.
2. Una vez hecho lo anterior, se otorgan otros 7 puntos de victoria por cada conjunto de tres cartas que se logre formar con tipos distintos. Cada carta de poder solo se podrá contabilizar para un solo conjunto.



Figura 1: Imagen que representa al primer caso de ejemplo.

Jugar 7 Maravillas fue bastante divertido, pero ahora los jugadores no saben cual es el máximo número de puntos de victoria que pueden conseguir con las cartas de poder que tienen en su mano, por lo que te han pedido ayuda para crear un programa que reciba los tipos de las cartas de poder y calcule el máximo número de puntos de victoria que se pueden conseguir.



Entrada

La entrada comienza con un número entero T , con $1 \leq T \leq 10^2$ indicando el número de casos de prueba a seguir. Cada caso de prueba comienza con dos números K y N separados por un espacio, donde $1 \leq K \leq 10^2$ y $0 \leq N \leq 10^4$, que representa el número de tipos distintos que tiene el juego y el número de cartas de poder que el jugador tiene en su mano respectivamente. En la siguiente línea se dan N enteros n_i con $1 \leq n_i \leq K$ separados por un espacio, que indica el tipo de poder que tiene la carta i -ésima.

Salida

Para cada caso de prueba, imprimir la cantidad máxima de puntos de victoria que el jugador puede conseguir.

Entrada Ejemplo	Salida Ejemplo
3	21
3 6	8
2 1 3 3 2 3	51
5 4	
3 5 3 5	
4 10	
1 2 2 3 3 3 4 4 4 4	



D - Robo Race

Tiempo Límite: 2 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

Terminando de jugar *7 Maravillas*, la pareja de Norman bajó de su habitación un nuevo juego de mesa cooperativo (se ve que es muy fan de los juegos de mesa), llamado *Robo Race*. Su objetivo principal es hacer que un pequeño robot toque una bandera, pasando por una serie de peligrosos y mortales obstáculos. Turno a turno, cada jugador recibe una serie de cartas, las cuales contienen instrucciones básicas para poder mover al robot, entre las que se encuentran: avanzar una casilla, girar a la izquierda, girar a la derecha o dar una vuelta en u. Empezando por el jugador inicial y en sentido de las manecillas del reloj, cada jugador elige una instrucción de su mano y la coloca boca abajo en una pila común. Cuando ningún jugador tenga cartas en la mano, se voltea la pila común y el robot empieza a realizar las instrucciones una a una. Si el robot no murió en su travesía o no llegó a tocar la bandera, entonces se repite el proceso. Si en algún momento, el robot toca la bandera, inmediatamente termina el juego y todos ganan, aunque aún haya cartas en la pila.

Como puede observarse, la complejidad del juego radica en que los jugadores no pueden ponerse de acuerdo en las instrucciones que van a jugar, por lo que el pobre robot siempre anda deambulando por todo el tablero. Entre los posibles obstáculos a los que se enfrenta el robot, son: bandas transportadoras, paredes y hoyos en el piso. Cuando el robot llega a una pared, ésta bloquea su camino, por lo que no será posible seguir avanzando. Si llega a una casilla con un hoyo, el robot se muere, se termina el juego y todos pierden. Finalmente, si después de ejecutar una instrucción, el robot queda sobre una banda transportadora, entonces se moverá automáticamente una casilla en la dirección de la banda (siempre que sea posible). Como es de esperarse, si el robot sale del tablero, también se considera muerto y cada casilla solo puede tener un tipo de obstáculo.

Debido a que es difícil coincidir en donde quedó el robot después de ejecutar las instrucciones en la pila, los invitados te han pedido ayuda con un programa, que dada la configuración inicial del tablero, la posición inicial del robot y la pila de cartas con instrucciones, decidas si el robot murió, tocó la bandera o menciones su posición final.

Entrada

La entrada comienza con dos números enteros N y M separados por un espacio, con $1 \leq N, M \leq 10^3$, que indican la anchura y longitud que tiene el tablero, luego le siguen N líneas, cada una con M enteros $0 \leq m_i \leq 20$ separados entre sí por un espacio, en donde el valor de cada m_i toma significado de acuerdo a la Figura 2.

Posteriormente se dan cuatro enteros r_1, r_2, b_1, b_2 con $1 \leq r_1, b_1 \leq N$ y $1 \leq r_2, b_2 \leq M$,

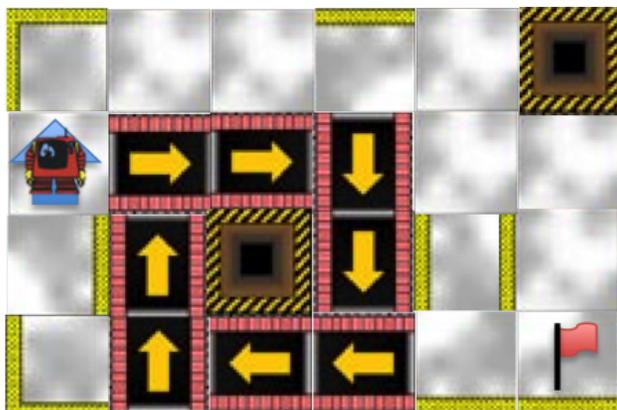


Figura 1: Posible tablero para el juego *Robo Race*. Este ejemplo se presenta en el primer caso de prueba.

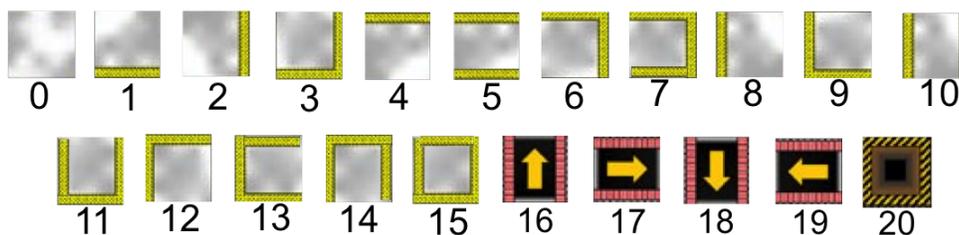


Figura 2: Significado de cada número en el tablero.

donde (r_1, r_2) indica la posición en donde inicia el robot y (b_1, b_2) la posición de la bandera. En la siguiente línea se proporciona un entero K con $0 \leq K \leq 10^3$ que representa el número de cartas que tiene la pila y finalmente le siguen K enteros k_i donde cada k_i tiene el siguiente significado:

- 1 - Avanzar una casilla
- 2 - Girar a la izquierda
- 3 - Girar a la derecha
- 4 - Dar vuelta en u

Dar por sentado que en ningún caso de prueba, la posición inicial de la bandera se encuentra sobre una banda o sobre un hoyo. El robot siempre inicia mirando hacia el norte.

Salida

Para cada caso de prueba, imprimir el mensaje: `Toco la bandera, todos ganan!` , `Murio en el intento, todos pierden!` o `(x,y) mirando hacia el D`, si después de ejecutar la pila de instrucciones sobre el tablero, el robot llegó a tocar la bandera, en algún momento se murió o llega a la posición (x, y) respectivamente. D es la dirección hacia don-



de quedó mirando el robot la cual puede tomar el valor de Norte, Sur, Este u Oeste.

Entrada Ejemplo

```
4 6
12 0 0 4 0 20
0 17 17 18 0 0
2 16 20 18 10 0
9 16 19 19 1 0
2 1 4 6
16
2 4 1 2 1 3 1 3 1 1 1 1 1 2 1 1
```

Salida Ejemplo

Toco la bandera, todos ganan!

Entrada Ejemplo

```
4 6
12 0 0 4 0 20
0 17 17 18 0 0
2 16 20 18 10 0
9 16 19 19 1 0
2 1 4 6
12
1 1 2 1 4 1 3 1 1 1 4 2
```

Salida Ejemplo

Murio en el intento, todos pierden!

Entrada Ejemplo

```
4 6
12 0 0 4 0 20
0 17 17 18 0 0
2 16 20 18 10 0
9 16 19 19 1 0
1 3 1 5
4
2 4 3 1
```

Salida Ejemplo

(2,4) mirando hacia el Sur



E - Vamos a Platicar

Tiempo Límite: 3 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

Después de haber pasado horas de diversión en la fiesta, Norman se dio cuenta de que ya era tiempo de regresar a su casa, ya que debido al programa *Un día sin carro* que ha impulsado el gobierno, tiene que regresar en transporte público. Pero no todo es tan malo como parece, ya que el transporte público en *Nlognoría* es muy eficiente, seguro y tiene camiones de alta tecnología, que recorren en los dos sentidos rutas únicas entre estaciones. Cada ruta tiene un costo de pasaje y siempre es posible llegar a una estación de cualquier otra.

Pero creo que ya me entretuve hablando de lo eficiente que es el transporte público, así que regresando al tema principal, Norman, antes de irse a su casa, convenció a Luis de que se fueran juntos, con el propósito de platicar algunos proyectos que tenía en mente. Como eran muchas cosas las que Norman quería hablar con Luis, pensó en que sería una buena idea compartir la mayor cantidad de rutas posibles durante el viaje, obviamente sin que ninguno de los dos gaste más de lo necesario, ya que ambos tienen exactamente lo justo para llegar cada quien a su casa.

De esta manera, Norman sacó un mapa en donde se indican las posibles rutas, así como el costo del pasaje asociado. Sin embargo, ninguno de los dos ha podido encontrar cuál es la mayor cantidad de rutas que pueden compartir, sin que incurra en un costo extra para ninguno de los dos. Por lo de nueva cuenta te han pedido ayuda con un programa que les ayude a encontrar la respuesta.

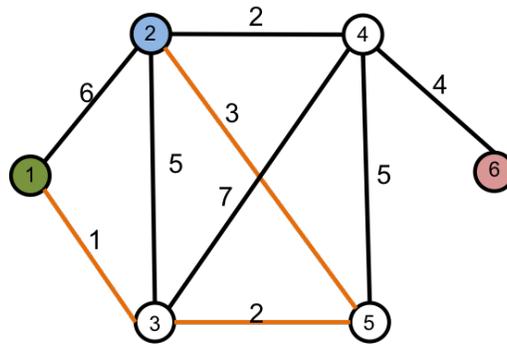


Figura 1: Mapa para el primer caso de ejemplo. El nodo verde, azul y rojo, simbolizan la parada del camión más cercana al lugar de la fiesta, a la casa de Luis y a la casa de Norman respectivamente. Las líneas naranjas representan la mayor cantidad de rutas que Norman y Luis pueden compartir, sin incurrir en un gasto extra para ninguno de los dos.

Entrada

La entrada consiste en varios casos de prueba (no más de 30), en donde cada uno de ellos comienza con 5 enteros n, m, s, t_1, t_2 con $1 \leq n \leq 2^{10}$, $(n-1) \leq m \leq \frac{n*(n-1)}{2}$ y $1 \leq s, t_1, t_2 \leq n$. En donde n es el número de paradas en el mapa, m es el número de posibles rutas y s, t_1, t_2 simbolizan la parada del camión más cercana al lugar de la fiesta, a la casa de Luis y a la casa de Norman respectivamente. Las siguientes m líneas consisten en 3 enteros u, v, p con $1 \leq u, v \leq n$ y $1 \leq p \leq 10^5$, que representan una posible ruta de la parada u a la parada v cuyo costo es p .

Salida

Para cada caso de prueba, imprimir tres enteros r, a, b en donde r indica la mayor cantidad de rutas que Norman y Luis pueden compartir, mientras que a representa el mínimo costo que Luis gasta para llegar a su casa. Análogo el entero b para Norman.



Entrada Ejemplo

6 9 1 2 6
1 2 6
1 3 1
2 3 5
2 4 2
2 5 3
3 4 7
3 5 2
5 4 5
4 6 4

Salida Ejemplo

3 6 12



F - Intercambio de Regalos

Tiempo Límite: 3 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

Tiempo después de la fiesta de la pareja de Norman, su grupo de amigos ha decidido llevar a cabo un intercambio de regalos como excusa para organizar un convivio, en donde puedan pasar un rato agradable lejos del estrés y los deberes de la escuela. De manera aleatoria, se repartió a cada integrante un papel con el nombre de la persona a quien se le tiene que dar el regalo y puede suponerse que bajo ninguna circunstancia, una persona se da regalo a sí misma. Para que el intercambio sea más entretenido y duradero, Norman propuso la siguiente forma de realizarlo:

1. Empezará cualquier persona que no haya repartido su regalo.
2. Aquella persona que lo reciba:
 - a) Si aún no ha repartido su regalo, entonces procede a entregarlo.
 - b) Si ya entregó su regalo, pero aún faltan personas por participar en el intercambio, entonces se realiza una nueva ronda comenzando en el paso 1, hasta que todos los participantes hayan repartido y recibido su regalo.

Norman tiene la idea de que los *intercambios completos*, en donde la persona que inicia el intercambio es la última en recibir su regalo, son los más entretenidos. Por lo cual, te ha pedido ayuda para crear un programa que dada la relación de los regalos, menciones si existe alguna persona con la cual iniciar, de tal manera que el intercambio sea completo. En caso de que sea imposible esta situación, Norman quiere saber cuál es el mínimo número de rondas necesarias para terminar el intercambio.

Entrada

La primera línea contiene un entero T con $1 \leq T \leq 10^2$ que representa el número de casos de prueba a seguir. Cada caso de prueba comienza con un entero N , con $2 \leq N \leq 10^4$ indicando el número de personas que participan en el intercambio. Posteriormente se dan N líneas, cada una de ellas con dos enteros A y B separados por un espacio, con $1 \leq A, B \leq N$ y $A \neq B$, los cuales refieren al hecho de que A le va a dar su regalo a B .

Salida

Imprimir la palabra **Completo**, si existe una persona que pueda iniciar el intercambio de tal manera que sea la última en recibir su regalo. En caso contrario, imprimir un entero con el mínimo número de rondas necesarias para completar el intercambio.

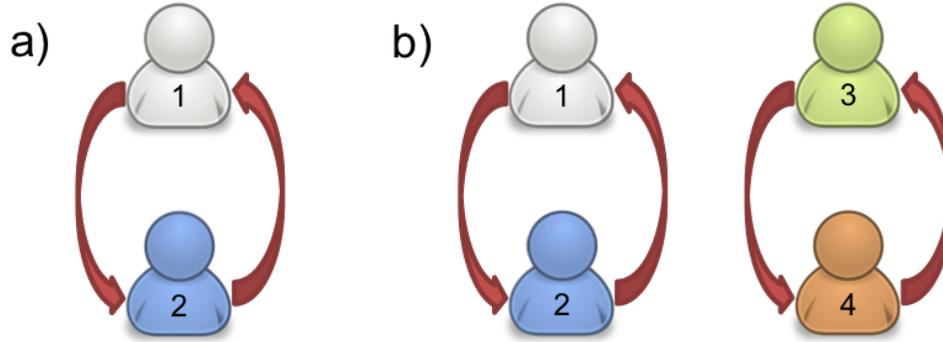


Figura 1: Imagen que representa el intercambio realizado en los casos de ejemplo

Entrada Ejemplo	Salida Ejemplo
2	Completo
2	2
1 2	
2 1	
4	
1 2	
2 1	
3 4	
4 3	



G - Trabajo Vacacional

Tiempo Límite: 2 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

Debido al periodo vacacional, Norman ha decidido trabajar en una agencia de viajes como guía turístico, dando recorridos por los lugares más emblemáticos de la ciudad. El sueldo que percibe por un día de trabajo es variable, y depende tanto del lugar que se visite como del número de personas en la reservación. A Norman le gustaría trabajar todos los días, ya que quiere juntar suficiente dinero para sorprender a su pareja, llevándola a cenar a uno de los mejores restaurantes, sin embargo, la ley prohíbe que un empleado labore más de 6 días consecutivos. Para poder planear las salidas, la agencia de viajes le ha proporcionado a Norman el calendario con las reservaciones de los siguientes N días, así como el sueldo correspondiente, por lo cual, te ha pedido ayuda, para que le menciones cuál es la mayor cantidad de dinero que podría conseguir.



Entrada

La primera línea es un entero N , con $1 \leq N \leq 10^5$ que representa las siguientes salidas a realizar, posteriormente, le siguen N enteros x_i con $0 \leq x_i \leq 10^3$, que simbolizan la cantidad de dinero que la agencia pagará por trabajar en el i -ésimo día.

Salida

Imprimir el símbolo \$ seguido de un entero que representa el número máximo de dinero que Norman puede conseguir durante la temporada vacacional, respetando la regla de no laborar más de 6 días consecutivos.



Entrada Ejemplo

9
10 10 10 10 10 5 20 10 2

Salida Ejemplo

\$82



H - El Juego del Tren

Tiempo Límite: 2 Segundos

Límite de la Pila: 10MB

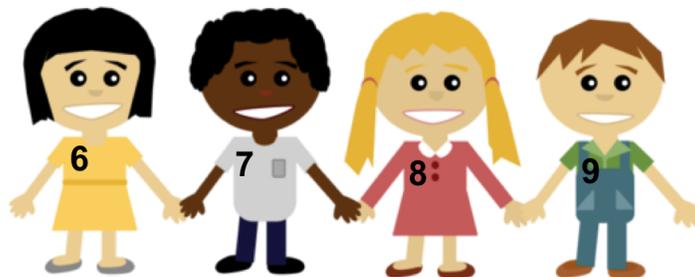
Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

La próxima semana, Norman será guía turístico en el museo de trenes de la ciudad, como parte de una excursión en una escuela. Para que la visita en el museo sea una experiencia inolvidable, a Norman se le ocurrió un juego fantástico basado en la temática de los trenes. Las reglas son bastante sencillas y se explican a continuación: Al principio del juego, se le reparte a cada niño un número, que simboliza un tipo de vagón (Carro tanque, torva granelera, Furgón, cabús, etc.).

Posteriormente, éstos tienen que ir tomándose de las manos, para tratar de formar el tren más largo posible. La única restricción que existe, es que un vagón sólo puede unirse por su lado derecho con su predecesor y por el otro con su sucesor. Por ejemplo, si a un niño le tocó el número 7, entonces solamente puede tomar con su mano derecha aquel que tenga el número 6 y con su mano izquierda al que tenga el número 8. En cualquier momento un niño puede soltarse de las manos para unirse a algún otro vagón, si considera que con ello puede aumentar el tamaño del tren más largo. Como Norman quiere que haya mucha interacción, te ha pedido ayuda con un programa, que dado los números que escogió para el juego, diga cuántos niños se quedarán fuera del tren más largo.



Entrada

La primera línea consiste en un entero T , con $1 \leq T \leq 15$ que representa los casos de prueba a seguir. Cada caso de prueba, es una línea con a lo más 10^4 enteros a_i con $0 \leq a_i < 2^{31}$ separados por espacio, que representan los números que Norman va a repartir entre los niños.

Salida

Para cada caso de prueba, imprimir el número de niños que se van a quedar fuera de el



tren más largo posible, utilizando el formato $P(\#Caso\ d) = r$, donde d el caso de prueba y r el resultado.

Entrada Ejemplo

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

Salida Ejemplo

```
P(#Caso 1) = 3
P(#Caso 2) = 2
```



I - Los Patos de la Feria

Tiempo Límite: 2 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

Norman decidió descansar después de una larga jornada de trabajo y aprovechó el día para ir a la feria de su colonia. Después de haberse subido en varios juegos mecánicos, le llamo mucho la atención un local donde la gente le disparaba a unos patos de metal. Lo que más le intrigó, es que no era el típico juego de tiro al blanco, sino una especie de acertijo en donde se podía ganar un gran premio. La mecánica es bastante sencilla y se explica a continuación: Una vez que se paga el respectivo precio por jugar, el encargado toma n patos de metal numerados del 1 al n , los revuelve y coloca alrededor de un círculo giratorio.

Por su parte, el jugador sólo puede dispararle al pato que se encuentra señalado por una flecha (la cual nunca se mueve y empieza señalando al primer pato de la secuencia). El objetivo del juego, es derribar a todos los patos en forma ascendente, es decir, primero se necesita dispararle al pato con el número 1, luego el que tiene el número 2 y así sucesivamente. Durante el juego no esta permitido dejar pasar un pato si es a este al que se le tiene que disparar.

Antes de comenzar a girar la rueda, el encargado hace la siguiente pregunta: Considerando que nunca se falla un tiro ¿Cuál es la cantidad de patos que se dejarán pasar antes de derribar a todos en el orden descrito?. Si el jugador acierta, entonces ganará un gran premio. Por ejemplo, si la secuencia de patos es 3,5,1,2,4 entonces la respuesta es 4, ya que en la primera vuelta se dejan pasar aquellos con número 3,5,4. Posteriormente, solo se deja pasar al que tiene el número 1. Considera que antes de girar la rueda se puede dispararle al primer pato de la secuencia, es decir, si el orden de los patos es 1,2,3, entonces no es necesario dejar pasar a ninguno de ellos para derribarlos a todos.

Para Norman fue bastante sencillo ganar el premio, sin embargo, quiso complicarse la vida (algo que raramente hace) y se hizo la siguiente pregunta que ahora no puede resolver, causándole noches de insomnio: Dado un número N de patos, un entero K y considerando que nunca se falla un tiro ¿Cuál es el número de secuencias que se pueden generar de tal manera que se derriben todos los patos dejando pasar exactamente a K de ellos?

Entrada

La entrada consiste en múltiples casos de prueba (no más $6 * 10^4$), donde cada uno de ellos contiene dos enteros N, K , separados por un espacio, con $1 \leq N \leq 10^3$, $0 \leq K \leq 10^3$ que indican el número de patos en la secuencia y la cantidad de ellos que se necesitan dejar pasar para derribarlos a todos. El final se indica por un número cero el cual no debe de



ser procesado.

Salida

Para cada caso de prueba imprimir un entero que representa el número de secuencias de patos que pueden derribarse dejando pasar exactamente a K de ellos. Como el número es bastante grande reduce la respuesta módulo $10^9 + 7$.

Entrada Ejemplo	Salida Ejemplo
3 2	2
6 7	101
0	



J- Los Feriapesos

Tiempo Límite: 1 Segundos

Límite de la Pila: 10MB

Límite de la Memoria: 32MB

Redacción: Manuel Alcántara Juárez

Verificador: Luis Enrique Chacón Ochoa

El premio que recibió Norman por ganar el juego de tiro al pato fue un fajo de *Feriapesos*, los cuales son billetes que pueden utilizarse para pagar o reducir el precio de diversos alimentos en la sección de comida. Cada establecimiento que acepta *Feriapesos* los cambia siempre a razón de 10 a 1, es decir, 10 *Feriapesos* equivale a un peso real y sólo se podrán cambiar en múltiplos de 10, con la finalidad de evitar los centavos. Por ejemplo, 15 *Feriapesos* equivalen solo a 1 peso, sin embargo 20 *Feriapesos* se cambian por 2 pesos.

A Norman se le antojó un buñuelo que cuesta X pesos y quiere utilizar la mayor cantidad de *Feriapesos* para pagarlo, sin embargo, debido a la restricción en el cambio de los *Feriapesos*, no sabe bien cuál es la cantidad que aún debe pagar. ¿Podrías de nueva cuenta ayudar a Norman?

Entrada

La primera línea se compone de un entero X que representa el precio del buñuelo y la siguiente contiene un entero F que simboliza la cantidad de *Feriapesos* que Norman posee. Suponer que $0 \leq X, F \leq 10^6$.

Salida

Imprimir el símbolo \$ seguido de un entero que representa la cantidad de dinero que aún debe pagar Norman para recibir su buñuelo.

Entrada Ejemplo	Salida Ejemplo
8 29	\$6

Entrada Ejemplo	Salida Ejemplo
1 10	\$0

*No te pierdas de las nuevas aventuras y problemas a los que se enfrentará Norman el próximo año.