



Olimpiada Chilena de Informática 2018

Regional

6 de Octubre, 2018

Este conjunto de problemas está siendo utilizado simultáneamente en las siguientes sedes:

Iquique (UTA), Viña (UTFSM), Santiago (UTFSM), Talca (UCM),
Chillán (UBB), Concepción (UCSC) y Temuco (UCT)

Las siguientes personas participaron en la elaboración de este conjunto de problemas:

Benjamín Aguilar, Rodrigo Alonso, Sebastián Barbieri,
Blaz Korecic, Nicolás Lehmann, Pablo Messina,
Martin Molina, Javier Oliva y Sebastián Zagal

Información General

Esta página muestra información general que se aplica a todos los problemas.

Envío de una solución

1. Los participantes deben enviar **un solo archivo** con el código fuente de su solución.
2. El nombre del archivo debe tener la extensión `.cpp` o `.java` dependiendo de si la solución está escrita en **C++** o **Java**, respectivamente. Para enviar una solución en Java hay que seguir algunos pasos adicionales. Ver detalles más abajo.

Casos de prueba y subtareas

1. La solución enviada por los participantes será ejecutada varias veces con distintos casos de prueba.
2. A menos de especificar explícitamente lo contrario, cada problema define diferentes subtareas que restringen el problema y se asignará puntaje de acuerdo a la cantidad de subtareas que logre solucionar completamente de manera correcta.
3. Una solución puede resolver al mismo tiempo más de una subtarea.
4. La solución es ejecutada con cada caso de prueba de manera independiente y por tanto puede fallar en algunas subtareas sin influir en la ejecución de otras.

Entrada

1. Toda lectura debe ser hecha desde la **entrada estándar** usando, por ejemplo, las funciones `scanf` o `std::cin` en C++ o la clase `BufferedReader` en Java.
2. La entrada corresponde a un solo caso de prueba el cual está descrito en varias líneas dependiendo del problema.
3. **Se garantiza que la entrada sigue el formato descrito** en el enunciado de cada problema.

Salida

1. Toda escritura debe ser hecha hacia la **salida estándar** usando, por ejemplo, las funciones `printf` o `std::cout` en C++ o `System.out.println` en Java.
2. El formato de salida es explicado en el enunciado de cada problema.
3. **El formato de salida debe ser seguido de manera estricta** considerando los espacios, las mayúsculas y minúsculas.
4. Toda línea, incluyendo la última, debe terminar con un salto de línea.

Envío de una solución en Java

1. Cada problema tiene un *nombre clave* que será especificado en el enunciado. Este nombre clave será también utilizado en el sistema de evaluación para identificar el problema.
2. Para enviar correctamente una solución en Java el archivo debe contener una clase llamada igual que el nombre clave del problema. Esta clase debe contener también el método `main`. Por ejemplo, si el nombre clave es `marraqueta` el archivo con la solución debiese llamarse `marraqueta.java` y tener la siguiente estructura:

```
public class marraqueta {  
    public static void main (String[] args) {  
        // tu solución va aquí  
    }  
}
```

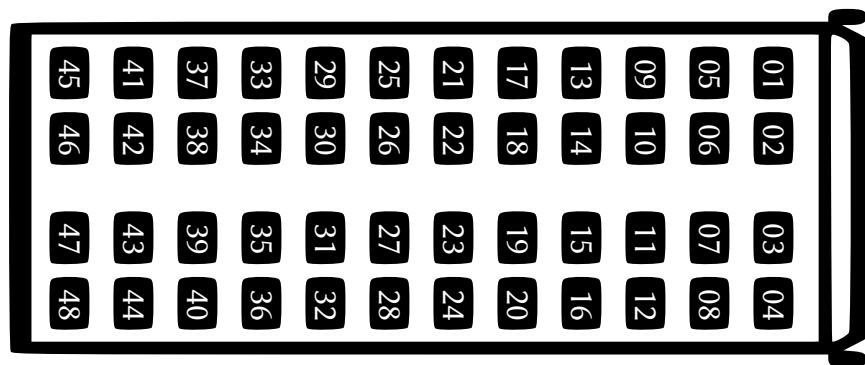
3. Si el archivo no contiene la clase con el nombre correcto el sistema de evaluación reportará un error de compilación.
4. La clase no debe estar contenida dentro de un *package*. Hay que tener cuidado pues algunos entornos de desarrollo como Eclipse incluyen las clases en un *package* por defecto.
5. Si la clase está contenida dentro de un *package* el sistema reportará un error de compilación.

Problema A

Bus de turismo

nombre clave: bus

La empresa de turismo *Al Sur del Mundo* tiene una flota de buses que utiliza para transportar a sus pasajeros en cada uno de los recorridos que ofrece a lo largo del país. Todos los buses en la flota tienen 48 asientos distribuidos en 12 filas y con un pasillo de separación, tal como se muestra en la siguiente figura:



A menudo la gente hace reservas para grupos de más de una persona. Como es de esperar, a las personas siempre les gusta sentarse junto a alguien de su mismo grupo. Se considera que dos personas están juntas si están sentadas en la misma fila y no están separadas por el pasillo. A una persona que viaja sola nunca le importará dónde sentarse.

Durante el último tiempo han surgido algunas inquietudes dentro de los directivos de la empresa, pues creen que mucha gente está quedando separada de su grupo, lo que estaría provocando mucha incomodidad entre los clientes. Los guías que trabajan directamente dentro de los buses creen que esto no es cierto pues ellos ayudan a los pasajeros a sentarse de forma óptima de modo que la máxima cantidad de pasajeros quede junto a alguien de su grupo.

Dado el número de gente en cada grupo, y suponiendo que todos se sientan de forma óptima, tu tarea es determinar la cantidad de gente que no queda sentada junto a alguien de su grupo.

Entrada

La primera línea de la entrada contiene un entero N correspondiente a la cantidad de grupos que han hecho reserva. Cada una de las siguientes N líneas contiene un entero mayor que cero correspondiente a la cantidad de gente en cada uno de los grupos. La cantidad total de gente nunca será mayor que 48.

Salida

La salida debe contener un único entero correspondiente a la cantidad de gente que **no** queda sentada junto a alguien de su grupo. A las personas que viajan solas no les importa dónde sentarse y por lo tanto no deben ser consideradas en esta cuenta.

Subtareas y puntaje

Este problema no contiene subtareas. Se probarán varios casos de prueba y se otorgará puntaje de acuerdo a la cantidad de casos de prueba correctos.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
4	2
5	
6	
23	
14	

Entrada de ejemplo	Salida de ejemplo
5	0
2	
1	
14	
4	
20	

Problema B

El juego del OClo

nombre clave: ocio

Los organizadores de la Olimpiada Chilena de Informática están aburridos mientras los participantes tratan de resolver los problemas. Para matar el tiempo deciden jugar al juego del OClo.

El juego del OClo consiste en encontrar los números de una secuencia. La secuencia comienza siempre con el número 1. Luego alguien lee en voz alta lo que ve, “un uno”, y lo escribe como el siguiente número en la secuencia, es decir, escribe 11. Luego otro jugador lee en voz alta el resultado del paso anterior, “dos unos”, y lo escribe: 21. Posteriormente el jugador siguiente lee “un dos, un uno” y lo escribe: 1211. Los OClosos jugadores continúan de esta forma, en cada paso leyendo en voz alta el resultado del paso anterior y luego escribiéndolo. Los primeros 9 números de la secuencia en el juego del OClo son los siguientes:

1, 11, 21, 1211, 111221, 312211, 13112221, 1113213211, 31131211131221

En esta ocasión los organizadores juegan al OClo escribiendo todos los números en un computador muy viejo con una memoria limitada de tamaño M . Cada dígito ocupa un espacio de memoria, y cada par de números consecutivos debe estar separado por un espacio, el cual también ocupa un espacio de memoria.

Los OClosos organizadores se preguntan cuantos números en la secuencia serán capaces de escribir **completamente** antes de que se les acabe la memoria. Por ejemplo, si $M = 19$, se pueden escribir completamente 5 números.

1		1	1		2	1		1	2	1	1		1	1	1	2	2	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

pero si $M = 18$, tan sólo se podrían escribir 4 números.

1		1	1		2	1		1	2	1	1		1	1	1	2	2	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	

¡Ayuda a los OClosos jugadores a encontrar la respuesta que buscan!

Entrada

La entrada consiste en un entero M que describe el tamaño de la memoria.

Salida

La salida debe contener un único entero correspondiente a la cantidad de números de la secuencia que pueden escribirse completamente en la memoria.

Subtareas y puntaje

30 puntos $0 < M \leq 50$

70 puntos $0 < M \leq 10^5$

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
19	5

Entrada de ejemplo	Salida de ejemplo
18	4

Problema C

Mega Woman

nombre clave: megawoman

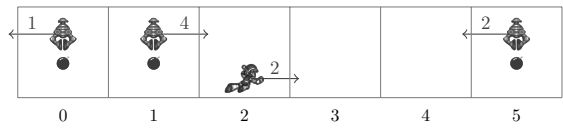
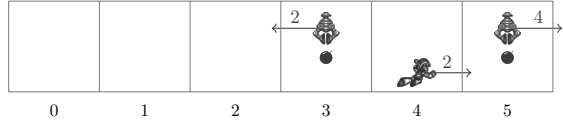
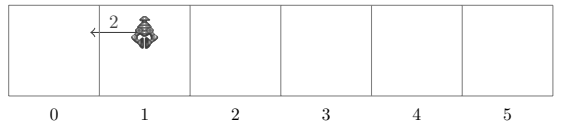
La reconocida empresa de videojuegos Jockeycom está desarrollando una nueva entrega de su aclamada franquicia Megawoman. En esta nueva versión, la heroína, Mega Woman, debe atravesar un escenario plagado de robots bombarderos y así poder salvar a su amigo Uno.

El escenario está compuesto de N casillas numeradas de 0 a $N - 1$ y contiene R robots numerados de 0 a $R - 1$. Mega Woman corre siempre hacia la derecha a una velocidad constante de v casillas por segundo. Cada robot vuela por el escenario en una sola dirección, ya sea izquierda o derecha, a una velocidad constante u_i . Tanto los robots como Mega Woman deben moverse todo el tiempo sin tener la posibilidad de detenerse.

Al comienzo de cada segundo los robots lanzan una bomba en la casilla en la que están. Si en ese momento Mega Woman se encuentra dentro de la misma casilla será alcanzada por la bomba y perderá el juego. Los robots solo lanzan una bomba al inicio de cada segundo y, por lo tanto, Mega Woman puede pasar por debajo de ellos si lo hace en el momento indicado.

El objetivo de Mega Woman es escapar del escenario, lográndolo cuando su posición está más a la derecha que la última casilla. Dada la posición inicial de Mega Woman y la de cada uno de los robots, a los desarrolladores de Jockeycom les gustaría determinar si es posible para Mega Woman escapar del escenario sin ser alcanzada por una bomba.

La siguiente tabla muestra segundo a segundo un ejemplo de un escenario donde Mega Woman puede escapar del escenario sin ser alcanzada por una bomba. La flechas indican la velocidad y dirección de cada uno de los personajes.

En el momento inicial los robots lanzan sus bombas, pero Mega Woman no es alcanzada por ninguna.	
Transcurrido un segundo, tanto Mega Woman como los robots se han movido la cantidad de casillas correspondientes a su velocidad. Adicionalmente todos los robots lanzan su bomba, pero Mega Woman no es alcanzada por ninguna.	
Al tercer segundo Mega Woman logra escapar del escenario y por lo tanto gana el juego. En este momento no importa las bombas que lancen los robots pues Mega Woman ya se encuentra fuera del escenario.	

Entrada

La entrada está descrita en varias líneas. La primera contiene un entero N ($N > 0$), correspondiente a la cantidad de casillas en el escenario. La siguiente línea contiene dos enteros w y v ($0 \leq w < N, 0 < v \leq 10^9$) correspondientes a la casilla donde se encuentra inicialmente Mega Woman y a la velocidad con la que se mueve. Posteriormente viene una línea con un entero R ($R > 0$) indicando la cantidad de robots en el escenario. Finalmente, siguen R filas describiendo cada uno de los robots. Cada fila contiene dos enteros r_i y u_i ($0 \leq r_i < N, |u_i| < 10^9$), correspondientes a la posición inicial y a la velocidad del robot, respectivamente. La velocidad será positiva si el robot se mueve hacia la derecha y negativa si se mueve hacia la izquierda.

Salida

La salida debe corresponder a una única línea conteniendo la palabra **SI** en el caso que sea posible para Mega Woman escapar del escenario o **NO** en caso contrario.

Subtareas y puntaje

20 puntos Se probarán varios casos donde la velocidad de Mega Woman es 1, todos los robots se mueven hacia la derecha con velocidad 2 y tanto la cantidad de casillas como de robots es menor o igual que 10^5 ($u_i = 2, v = 1, R \leq 10^5, N \leq 10^5$).

20 puntos Se probarán varios casos donde la velocidad de Mega Woman es 1, todos los robots se mueven hacia la izquierda con velocidad -1 y tanto la cantidad de casillas como de robots es menor que 10^5 ($u_i = -1, v = 1, R \leq 10^5, N \leq 10^5$).

25 puntos Se probarán varios casos en que el número de robots es menor que 50, el número de casillas es menor o igual que 10^5 y sin restricciones sobre las velocidades ($R \leq 50, N \leq 10^5$).

35 puntos Se probarán varios donde el número de robots es menor o igual que 10^5 , la cantidad de casillas es menor o igual que 10^{12} y sin restricciones sobre las velocidades ($R \leq 10^5, N \leq 10^{12}$).

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
6	SI
2 2	
3	
1 4	
0 -1	
5 -2	

Entrada de ejemplo

6
2 3
3
1 4
0 -1
5 -2

Salida de ejemplo

NO

Entrada de ejemplo

5
2 3
3
1 4
0 -1
5 -2

Salida de ejemplo

SI

Problema D

Reloj inteligente

nombre clave: reloj

Nicolás acaba de comprarse un *reloj inteligente*. Ahora, cada vez que realiza una actividad deportiva, Nicolás lleva su nuevo reloj, pues este le entrega información muy útil como su velocidad o ritmo cardíaco. Por otro lado, su reloj también se conecta con una aplicación en su celular y lleva un registro de todas las actividades que realiza. La parte entretenida es que la aplicación propone ciertos desafíos. En uno de los desafíos, Nicolás puede escoger un entero M y, entre todas sus actividades, la aplicación le dirá el tiempo mínimo en que ha podido recorrer M metros.

Después de un tiempo ocupando la aplicación, Nicolás se ha dado cuenta de que el desafío no funciona como él esperaba. Cada vez que empieza una actividad, el reloj guarda el tiempo cada M metros recorridos. Por ejemplo, si $M = 3$, la aplicación guardará el tiempo que tardó en recorrer los primero 3 metros, luego el tiempo entre el metro 3 y el 6, entre el 6 y el 9, etc. A Nicolás le gustaría que la aplicación considerara todos los intervalos en los que recorrió M metros. Por ejemplo, para $M = 3$, otros intervalos posibles podrían ser entre el metro 2 y 5 o entre el 7 y el 10.

Para suerte de Nicolás, el reloj es realmente inteligente y puede programarse para añadir nuevas funcionalidades. Lamentablemente, Nicolás es realmente malo programando y necesita de tu ayuda.

Internamente, el reloj toma una *muestra* cada segundo y registra la distancia en metros que fue recorrida durante ese segundo. Las muestras son guardadas en un arreglo de enteros de tamaño N . Un intervalo corresponde a un conjunto consecutivo de muestras y su distancia, a la suma total de todas las muestras en él. Como es posible que la distancia de los intervalos no sea exactamente M , a Nicolás le interesan los intervalos que sean lo más cercanos posible a M . Un *intervalo válido* es uno cuya distancia es mayor o igual que M , tal que al quitarle la muestra de más a la derecha, la distancia queda menor que M . Dada la descripción de las muestras, tu tarea es determinar el menor tiempo en que Nicolás recorrió un intervalo válido.

La siguiente figura contiene un ejemplo para un arreglo de muestras de tamaño 7. Si $M = 8$, algunos intervalos válidos son el contenido entre las posiciones 1 y 4 (distancia $3 + 2 + 1 + 2 = 8$) y el contenido entre las posiciones 2 y 5 (distancia $2 + 1 + 2 + 4 = 9$). Por otro lado, el intervalo contenido entre las posiciones 2 y 6, a pesar de tener una distancia mayor que 8, no es válido, pues al sacarle la muestra de más a la derecha su distancia todavía es mayor que 8. Finalmente, el intervalo válido de menor tiempo es el contenido entre las posiciones 4 y 6.

1	3	2	1	2	4	3
0	1	2	3	4	5	6

Entrada

La primera línea de la entrada contiene dos enteros M ($0 < M \leq 10^5$) y N ($N > 0$), correspondientes respectivamente al valor que Nicolás ha escogido para el desafío y la cantidad de muestras que registró

el reloj. La siguiente línea contiene N enteros mayores o iguales que 0 y menores o iguales que 100, correspondientes a la cantidad de metros recorridos en cada muestra.

Salida

La salida debe contener un único entero correspondiente al tiempo en segundos del intervalo válido de menor tiempo. Se garantiza que habrá al menos un intervalo válido, es decir, la distancia del intervalo formado por todas las muestras será mayor o igual que M .

Subtareas y puntaje

20 puntos Se probarán varios casos donde $N \leq 10^2$

30 puntos Se probarán varios casos donde $N \leq 2 \times 10^3$

50 puntos Se probarán varios casos donde $N \leq 10^6$

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
8 7 1 3 2 1 2 4 3	3

Entrada de ejemplo	Salida de ejemplo
1 2 1 1	1

Entrada de ejemplo	Salida de ejemplo
10 5 2 2 2 2 2	5

Problema E

Las sumas de Gabriela

nombre clave: sumas

Gabriela necesita ahorrar mucho dinero. Es por esto que pasa todo el día haciendo cálculos para saber cuánto dinero ha gastado. Los cálculos de Gabriela son simples: dada una lista de números a_1, a_2, \dots, a_n , debe sumarlos para obtener un valor S .

Gabriela mantiene en su libreta un registro de todas las sumas que ha hecho, pero ella no quiere que la gente sepa cuánto dinero ha gastado. Por esta razón, diseñó una estrategia para poder ocultarlo. Su estrategia consiste en anotar los números en sus sumas sin escribir ningún signo o espacio de separación. Por ejemplo, para la suma $1111 + 2222 + 3333 = 6666$ ella escribiría el siguiente registro en su libreta 1111222233336666.

Dado un registro, a Gabriela le interesa poder recuperar la suma original. Preocupada de no poder hacerlo, decidió agregar información extra incluyendo al principio de todos sus registros la cantidad de sumandos en la operación. Para el ejemplo anterior la cantidad de sumandos es 3 y por lo tanto el registro final que Gabriela escribiría en su libreta es 31111222233336666.

Por casualidad acabas de recibir en tus manos la libreta de Gabriela. ¿Crees ser capaz de descifrar sus cálculos?

Entrada

La entrada consiste en una línea que contiene una cadena de largo M formada únicamente por los siguientes caracteres: $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (no contiene el 0). Esta cadena corresponde a uno de los registros de Gabriela, es decir, para alguna lista de números a_1, a_2, \dots, a_n cuya suma es S , la línea tendrá el formato:

$$na_1a_2\dots a_nS$$

Se garantiza que la suma tiene al menos dos sumandos ($n \geq 2$) y que todos los números son mayores que 0 y menores o iguales que 10^9 .

Salida

La salida debe contener el registro de Gabriela agregando espacios entremedio de los números, es decir, debe tener el formato:

$$n _ a_1 _ a_2 _ \dots _ a_n _ S$$

Está garantizado que siempre existe una forma válida de separar el registro y que esta es única.

Subtareas y puntaje

40 puntos Se probarán varios casos donde $M \leq 15$.

60 puntos Se probarán varios casos donde $M \leq 28$.

Ejemplos de entrada y salida

Entrada de ejemplo

31111222233336666

Salida de ejemplo

3 1111 2222 3333 6666

Entrada de ejemplo

81234567836

Salida de ejemplo

8 1 2 3 4 5 6 7 8 36