



**Olimpiada Chilena de Informática
2015**

Final Nacional

12 de Noviembre, 2015

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 `.c`, `.cpp` o `.pas` dependiendo de si la solución está escrita en C, C++ o Pascal respectivamente.

Casos de Prueba y Subtareas

1. La solución enviada por los participantes será ejecutada varias veces con distintos casos de prueba.
2. Cada problema define diferentes subtareas que restringen el problema y se asignará puntaje de acuerdo a la cantidad de subtareas que logre solucionar 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

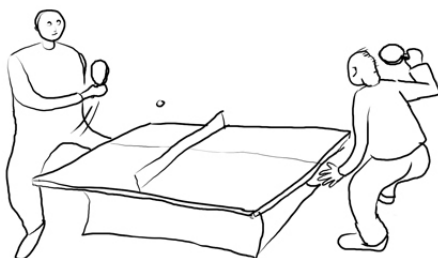
1. Toda lectura debe ser hecha desde la **entrada estándar** usando, por ejemplo, las funciones `scanf`, `std::cin` o `readln` dependiendo del lenguaje escogido.
2. La entrada corresponde a un solo caso de prueba el cual está descrito en varias líneas dependiendo del problema.
3. **Se debe asumir que 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`, `std::cout` o `writeln` dependiendo del lenguaje escogido.
2. El resultado del caso de prueba debe ser escrito siguiendo el formato descrito para cada problema.
3. **El formato de salida debe ser seguido de manera estricta** considerando los espacios especificados.
4. Toda línea, incluyendo la última, debe terminar con un salto de línea.

Problema A

Partido de ping-pong



Jota Pe y Nelman son fanáticos del ping-pong. Cada vez que se juntan aprovechan de jugar un partido. Sus partidos no son profesionales y siguen las siguientes reglas que ellos mismos han inventado:

- El partido se juega a N puntos y el primero que llega a los N puntos gana.
- La cantidad de puntos, es decir, el valor de N , puede ser distinto en cada partido.
- En cada partido uno de los dos comienza sacando.
- Cada jugador conserva su saque hasta perder un punto.

Lo que no saben Jota Pe y Nelman es que sus partidos son muy predecibles. Resulta que cada vez que a Jota Pe le toca sacar tiene una racha de A puntos, es decir, gana A puntos seguidos y luego pierde el saque. Lo mismo pasa con Nelman, cada vez que a él le toca sacar tiene una racha de B puntos y luego pierde su saque. El largo de las rachas, es decir, el valor de A y B , depende del día en que jueguen. Notar que cuando un jugador pierde el saque el oponente gana un punto.

Jota Pe y Nelman quedarían muy sorprendidos si alguien fuera capaz de adivinar quién de los dos va a ganar un partido antes de que lo jueguen. ¿Qué tal si haces un programa para sorprenderlos?

Entrada

La entrada está compuesta de dos líneas.

La primera línea contiene dos enteros N y P . N corresponde a la cantidad de puntos a los que se jugará el partido. P es un entero que indica quién comenzará sacando (Jota Pe = 1 y Nelman = 2).

La siguiente línea contiene dos enteros A y B , que representan respectivamente el largo de las rachas de Jota Pe y de Nelman.

Salida

Tu programa debe imprimir un 1 si Jota Pe es quién ganará el partido y un 2 si Nelman es quién ganará el partido.

Subtareas y Puntaje

10 puntos Se probarán varios casos donde $1 \leq N \leq 100$, $P = 1$, y $A = B = 0$.

10 puntos Se probarán varios casos donde $1 \leq N \leq 100$, $P = 1$, y $A = B = 1$.

20 puntos Se probarán varios casos donde $1 \leq N \leq 100$, $A = B$, y $0 \leq A, B \leq N$.

60 puntos Se probarán varios casos donde $1 \leq N \leq 100$, y $1 \leq A, B \leq N$.

Ejemplos de Entrada y Salida

Entrada de ejemplo	Salida de ejemplo
11 1 0 0	2

Entrada de ejemplo	Salida de ejemplo
11 1 1 1	1

Entrada de ejemplo	Salida de ejemplo
7 2 2 2	2

Entrada de ejemplo	Salida de ejemplo
6 1 2 3	2

Problema B

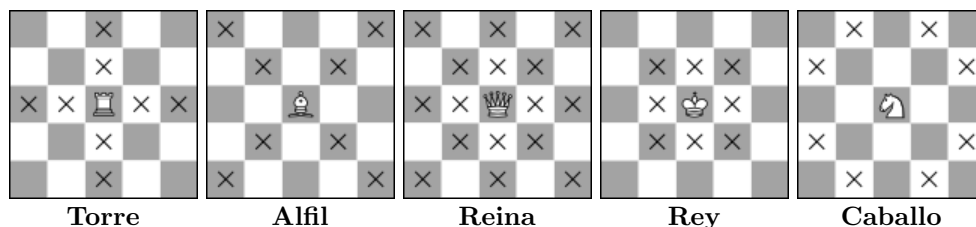
Ajedrez

El ajedrez es un juego de tablero conocido por su dificultad. Los mejores jugadores del mundo destacan por su capacidad de anticiparse a todas las posibles jugadas. El juego tiene ya casi 1500 años, pero continúa siendo muy popular y en la actualidad siguen desarrollándose nuevas técnicas y estrategias.

El ajedrez se juega entre dos personas sobre un tablero de 8×8 casillas. Cada jugador tiene a su disposición un conjunto de piezas que debe mover estratégicamente para lograr vencer a su oponente. En cada turno un jugador debe mover una de sus piezas desde una casilla a otra. En el ajedrez, cada tipo de pieza tiene distintas reglas para poder moverse. En este problema solo nos interesa la forma en que puede moverse cada tipo de pieza sobre un tablero.

Consideraremos el movimiento de 5 tipos de piezas: la torre, el alfil, la reina, el rey y el caballo. A continuación se describen las reglas de movimiento de cada pieza. Adicionalmente, la figura de más abajo muestra las posibles casillas en un tablero de 5×5 a las que podría moverse cada pieza en un solo movimiento.

- **Torre:** La torre puede moverse en línea recta de forma horizontal o vertical cuantas casillas quiera en un movimiento.
- **Alfil:** El alfil puede moverse a lo largo de una diagonal cuantas casillas quiera en un movimiento.
- **Reina:** La reina es una combinación de la torre y el alfil, ya que puede moverse en diagonal, horizontal o verticalmente cuantas casillas quiera en un movimiento.
- **Rey:** El rey, al igual que la reina, puede moverse en cualquier dirección pero solo una casilla por movimiento.
- **Caballo:** El caballo es la pieza con el movimiento más complicado. Su movimiento es en forma de “L”, es decir, siempre se mueve 2 casillas en una dirección (horizontal o vertical), y luego una casilla en la otra dirección.



Tu tarea consiste en crear un programa que determine la mínima cantidad de movimientos en los que es posible llevar una pieza desde una casilla a otra en un tablero de 100×100

Entrada

La entrada consiste en 3 líneas.

La primera línea contiene un único entero P , que representa el tipo de pieza que se desea mover (torre=1, alfil=2, reina=3, rey=4, caballo=5).

La segunda línea contiene dos enteros X_i y Y_i , correspondientes a las coordenadas de la posición inicial de la pieza. El valor X_i corresponde a la columna de la casilla en el tablero, mientras que Y_i corresponde a la fila. Las columnas y filas del tablero son numeradas con valores entre 0 y 99, de izquierda a derecha y de abajo hacia arriba. La columna de más a la izquierda corresponde a la columna 0 y la de más a la derecha a la columna 99. La fila de más abajo corresponde a la fila 0 y la de más arriba a la fila 99.

La tercera línea de la entrada contiene dos enteros X_f y Y_f correspondientes a las coordenadas de la posición donde se quiere mover la pieza.

Los valores de X_i , Y_i , X_f y Y_f siempre serán mayores o iguales que 0 y menores o iguales que 99. Además P siempre representará una pieza válida, es decir, $1 \leq P \leq 5$.

Salida

Tu programa debe imprimir una única línea con un entero: el número mínimo de movimientos en que es posible llevar la pieza desde la casilla inicial a la final. Si no es posible mover la pieza desde la casilla inicial a la final tu programa debe imprimir -1.

Subtareas y Puntaje

20 puntos Se probarán varios casos solo con torres ($P=1$)

20 puntos Se probarán varios casos solo con alfiles ($P=2$)

20 puntos Se probarán varios casos solo con reinas ($P=3$)

20 puntos Se probarán varios casos solo con reyes ($P=4$)

20 puntos Se probarán varios casos solo con caballos ($P=5$)

Ejemplos de Entrada y Salida

Entrada de ejemplo	Salida de ejemplo
1 10 10 45 30	2

Entrada de ejemplo

2
2 2
5 4

Salida de ejemplo

-1

Entrada de ejemplo

3
99 98
1 0

Salida de ejemplo

1

Entrada de ejemplo

4
50 50
51 52

Salida de ejemplo

2

Entrada de ejemplo

5
50 50
50 49

Salida de ejemplo

3

Entrada de ejemplo

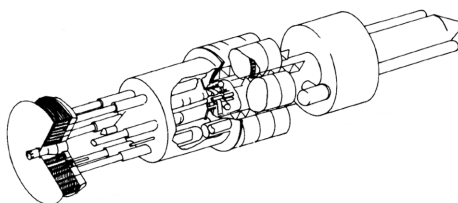
5
0 0
99 99

Salida de ejemplo

66

Problema C

Sistemas interestelares



La Oficina de Comunicaciones Interestelares (OCI) es una organización formada por distinguidos científicos de todo el mundo y encargada del estudio de comunicaciones a gran distancia.

La división de señales de alta potencia de la OCI ha descubierto recientemente un método para enviar señales con un alcance de cientos de años luz. Para probarlo, se iniciará un programa de emisión de estas ondas de largo alcance. Sin embargo, el programa es muy costoso, y en esta primera fase el presupuesto solo alcanza para emitir señales a 5 sistemas interestelares dentro de la Vía Láctea. Para apoyar el trabajo de la división de Búsqueda de Vida Extraterrestre (BVET), el directorio de la OCI ha decidido que el destino de estas señales sea determinado por los científicos de la BVET.

La principal actividad de la BVET consiste en el análisis de ondas de radio provenientes de sistemas interestelares dentro de nuestra galaxia. Esporádicamente, aparecen pulsaciones que podrían indicar la presencia de vida extraterrestre. Sus equipos monitorean estas pulsaciones registrando para cada evento el sistema de donde provino.

La BVET decidió emitir las señales a los cinco sistemas que hayan registrado más eventos con pulsaciones, pues esto indicaría una mayor probabilidad de existencia de vida extraterrestre. Sin embargo, sus equipos han registrado muchos eventos y están teniendo problemas para determinar cuáles son los 5 sistemas con más eventos. Tú, que has sido recientemente contratado por la OCI, debes ayudar a los científicos de la BVET a tomar la decisión.

Entrada

La entrada consiste en varias líneas. La primera línea contiene dos enteros N y S separados por un espacio. N corresponde a la cantidad de eventos que han sido registrados por los equipos de la BVET, y S corresponde a la cantidad de sistemas monitoreados. Cada sistema es representado con un entero entre 1 y S .

Las siguientes N líneas corresponden a cada uno de los eventos con pulsaciones. Cada línea contiene un único entero i ($1 \leq i \leq S$) correspondiente al sistema en el cual se detectó el evento. Se garantiza que al menos habrán cinco sistemas para los cuales existen eventos.

Salida

Tu respuesta debe consistir en una única línea con cinco enteros separados por espacios. Estos deben corresponder a los cinco sistemas que registraron más eventos. Puedes imprimir estos enteros en

cualquier orden. Si existe más de una solución puedes reportar cualquiera. (Mira la explicación luego de los casos de ejemplo.)

Subtareas y Puntaje

5 puntos Se probarán varios casos donde $N = S = 5$.

10 puntos Se probarán varios casos donde $N = 5$, $S = 6$.

20 puntos Se probarán varios casos donde $5 \leq N \leq 100$ y $5 \leq S \leq 100$.

30 puntos Se probarán varios casos donde $100 < N \leq 10^5$ y $100 < S \leq 10^5$.

35 puntos Se probarán varios casos donde $100 < N \leq 10^5$ y $10^8 < S \leq 10^9$.

Ejemplos de Entrada y Salida

Entrada de ejemplo

5 5
5
3
1
2
4

Salida de ejemplo

1 2 3 4 5

Entrada de ejemplo

5 6
5
1
6
3
4

Salida de ejemplo

5 1 6 3 4

Entrada de ejemplo	Salida de ejemplo
14 10 2 6 6 7 7 2 1 4 4 1 6 5 7 5	6 7 1 2 4

Nota que dado que el orden de los valores en la respuesta no importa, en el primer caso de ejemplo otra posible respuesta válida puede ser

5 4 3 2 1

Por su parte, en el segundo caso de ejemplo otra posible respuesta válida puede ser

1 6 3 5 4

Más aun, si analizas el tercer caso de ejemplo, en los sistemas 6 y 7 se detectaron 3 pulsaciones, mientras que en los sistemas 1, 2, 4 y 5 se detectaron exactamente 2 pulsaciones, y en los demás sistemas no se detectaron pulsaciones. Por esto, las respuestas

6 7 1 4 5

y

6 7 2 4 5

también serían válidas.

Recuerda que independiente de la cantidad de respuestas válidas que existan tu programa solo debe generar una de ellas.

Problema D

Gollum vs Golomb



Nuestro querido amigo Gollum se encuentra en medio de un debate de acertijos contra el hábil Golomb por la disputa del anillo único. Luego de una infructuosa ronda de acertijos clásicos, Golomb propone el siguiente desafío a Gollum:

“Considera una secuencia de enteros positivos $G(1), G(2), G(3), \dots$ con la propiedad de que para todo valor $n \geq 1$ se tiene que $G(n) \leq G(n+1)$ y además el número n aparece exactamente $G(n)$ veces en la secuencia. ¿Cuál es el valor de $G(10^{10})$?”

Gollum se estremece de emoción al escuchar tan interesante problema. Luego de pensar bastante, y de emitir sonidos de naturaleza diversa, Gollum llega a varias conclusiones sobre la secuencia. En primer lugar llega a la maravillosa conclusión de que $G(1)$ debe ser 1, lo que implica que el 1 aparece una única vez y al inicio de la secuencia. Esto lo lleva a concluir que el segundo número en la secuencia debe ser 2, es decir $G(2) = 2$. De acá concluye que el 2 aparece exactamente 2 veces en la secuencia y por lo tanto $G(2)$ y $G(3)$ deben ser 2. De hecho, siguiendo con estas deducciones Gollum se da cuenta que los primeros valores de la secuencia son los siguientes:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$G(n)$	1	2	2	3	3	4	4	4	5	5	5	6	6	6	6

Por ejemplo, de la tabla de arriba obtenemos que $G(8) = 4$ y $G(15) = 6$. Gollum logra calcular mentalmente algunos valores más, pero rápidamente se da cuenta de que necesitará ser mucho más astuto si desea calcular $G(10^{10})$. Luego de ver su frustración, Golomb se apiada de Gollum y le permite utilizar el comodín telefónico. Gollum recuerda que participaría en la OCI y espera que puedas ayudarlo utilizando el computador. Tu tarea consiste en hacer un programa que dado un entero n determine cuál es el valor de $G(n)$.

Entrada

La entrada consiste en una única línea con un entero positivo n .

Salida

Tu programa debe imprimir solo un entero con el valor de $G(n)$.

Subtareas y Puntaje

10 puntos Se probarán varios casos donde $0 < n \leq 15$.

25 puntos Se probarán varios casos donde $15 < n \leq 25$.

30 puntos Se probarán varios casos donde $25 < n \leq 10^5$.

35 puntos Se probarán varios casos donde $10^8 < n \leq 10^{10}$.

Nota: En la subtarea 3 el valor de n puede ser muy grande, tan grande que no pueda almacenarse en una variable de tipo `int`. Debes asegurarte de guardar el valor en una variable de tipo `long long`.

Ejemplos de Entrada y Salida

Entrada de ejemplo	Salida de ejemplo
10	5

Entrada de ejemplo	Salida de ejemplo
1000	86

Entrada de ejemplo	Salida de ejemplo
9000000000	1705830