LABORATORIO DE INVESTIGACIÓN DE SOFTWARE

8º Competencia de programación
2016

DEPARTAMENTO DE INGENIERÍA
EN SISTEMAS DE INFORMACIÓN

FACULTAD REGIONAL CÓRDOBA

Reglas de la Competencia

- Los integrantes de los equipos no pueden utilizar ningún medio de almacenamiento (pendrives, USB, etc.).
- No está permitido el uso de celulares, notebook, ni ningún otro dispositivo móvil.
- No está permitido el uso de internet.
- El equipo que viole cualquiera de estas condiciones quedará automáticamente descalificado.
- No se responderán consultas sobre los problemas o la programación de las soluciones en forma oral. El único canal para dichas consultas es mediante el envío de una solicitud de clarificación desde el software PC^2. Sólo se responderán las clarificaciones enviadas hasta las 21:00.

Problema 1: Función exponencial

Se conoce que la función exponencial puede ser definida de múltiples maneras, entre ellas, una definición posible es:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Esta función es de gran importancia en el estudio del plano complejo, y en el campo del análisis, ya que es la única función que es siempre igual a su derivada. Para su referencia se sabe que:

$$e^1 \approx 2,708333333 \text{ aproximado con 5 sumandos}$$
$$e^2 \approx 7,380952381 \text{ aproximado con 8 sumandos}$$

Usted debe determinar la suma de los valores de los decimales en las posiciones comprendidas entre a y b, ambos inclusive, del número e^x (calculado como una aproximación de los primeros k términos según la definición anterior). Con a, b, x, k enteros, y $a \leq b$. Por ejemplo para $x = 1$, $k = 5$, $a = 1$ y $b = 3$, el resultado será 15 pues es la suma de los tres primeros decimales para el cálculo de la función exponencial con los 5 primeros sumandos.

Entrada

La entrada comienza con un entero C que indica la cantidad de casos de prueba. Por cada caso de prueba se reciben cuatro números enteros x, k, a y b, siempre en ese orden. Donde x indica que se desea e^x, k indica que se lo va a calcular con la aproximación de k sumandos, a indica la posición del primer decimal que se tendrá en cuenta para la suma, y b es la posición del segundo decimal. Se sabe que siempre $a \leq b$.

Salida

Por cada caso de prueba se debe imprimir un número entero con el valor de la suma de los decimales ubicados entre los lugares a-ésimo y b-ésimo de la función e^x aproximada con los k primeros sumandos.

Ejemplo

Entrada	Salida
2 k a b	15
1 5 1 3	17
2 8 2 4	

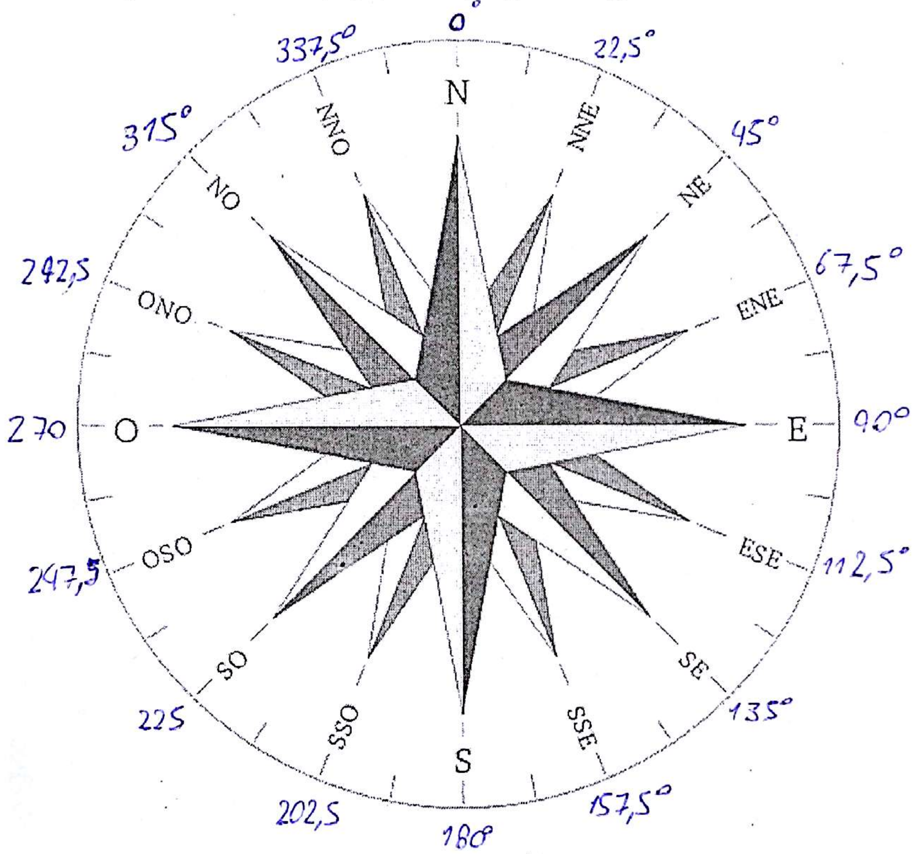
exponente, precisión, rango

Problema 2: Barco a la deriva

Un barco tiene rota su brújula, por lo tanto luego de realizar cada giro se desconoce hacia donde está avanzando.

Sin embargo el capitán de este barco sabe correctamente cuántos grados gira cada vez que lo hace y está seguro de que el puerto desde donde salió le permite iniciar su recorrido exactamente hacia el norte.

Se requiere un programa que reciba la dirección y ángulo de cada giro e informe el punto cardinal hacia el que está dirigido. Para ello el programa debe informar el punto cardinal con una precisión de 22,5°, es decir, según la siguiente rosa de los vientos:



En caso de que el barco no finalice apuntando exactamente a uno de los 16 puntos cardinales indicados, se debe informar el más cercano.

Entrada

La entrada inicia con una línea con la cantidad de casos de prueba C. Luego se presentan C casos, cada uno de los cuales presenta una línea conteniendo un número entero G indicando la cantidad de giros y luego G líneas. Por cada giro se ingresa una letra B o E, indicando si el giro fue a babor (izquierda, sentido antihorario) o estribor (derecha, sentido horario) respectivamente y luego un número entero A con la cantidad de grados que giró.

$$0 < C < 10^4$$

$$0 < G < 10^8$$

$$0 \leq A < 360$$

Salida

Por cada caso de prueba se debe informar una cadena con la sigla del punto cardinal hacia el que el barco está más orientado (una de {N, NNE, NE, ENE, E, ESE, SE, SSE, S, SSO, SO, OSO, O, ONO, NO, NNO}).

Ejemplo

Entrada

C ← 2
G1 ← 3
B 45
dato1 B 45
B 0
G2 ← 5
E 15
E 15
dato2 B 10
B 10
E 5

Salida

```
O
NNE
```

Problema 3: Computadora primitiva

En un laboratorio de Investigación de una Universidad unos investigadores encontraron en un rincón una computadora primitiva usada por los militares para comunicarse durante la guerra con señales.

Cuando la prendieron a la computadora se dan cuenta que solo es capaz de almacenar un número entero que está compuesto por 32 bits. (Cada bit es representado por un 0 o un 1)

Al lado de la computadora se encontró también una caja que almacenaba tarjetas de memoria (bastante primitivas) que se insertaban en la computadora y tenían cada una de ellas ese número.

Se demostró que un mensaje estaba compuesto por una o más tarjetas que contenían estos números ya que cada tarjeta tenía un código de mensaje.

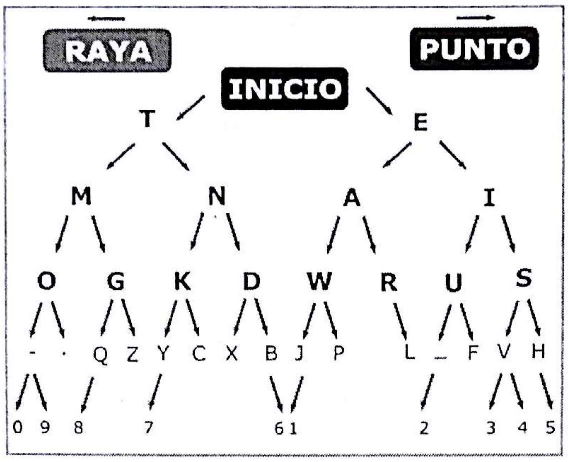
Luego de muchos estudios se dan con la situación de que en la época los militares utilizaban muy frecuentemente el Código Morse para comunicarse por lo cual empezaron a interpretar que los bits que estaban en cada memoria eran letras en Morse. Todos los mensajes consisten únicamente de letras mayúsculas. Además se identificó que los mensajes tenían todos los bits invertidos, es decir que la interpretación debía realizarse desde el bit menos significativo hacia el más significativo.

Código Morse					
A	• —	N	— •	1	• — — — —
B	— • • •	O	— — —	2	• • — — —
C	— • — •	P	• — — •	3	• • • — —
D	— • •	Q	— — • —	4	• • • • —
E	•	R	• — •	5	• • • • •
F	• • — •	S	• • •	6	— • • • •
G	— — •	T	—	7	— — • • •
H	• • • •	U	• • —	8	— — — • •
I	• •	V	• • • —	9	— — — — •
J	• — — —	W	• — —	0	— — — — —
K	— • —	X	— • • —		
L	• — • •	Y	— • — —		
M	— —	Z	— — • •		

Para poder interpretar correctamente el mensaje en morse, se utilizó en esta computadora una codificación bastante particular. Por cada caracter a transmitir se utilizaban entre tres y seis bits. Los dos primeros bits indican la longitud del caracter, según la siguiente tabla:

Bits	Longitud
00	1
01	2
10	3
11	4

A continuación de los bits de longitud aparecen tantos bits como indica dicha longitud. Esos bits representan una letra codificada en morse donde el punto es un 1 y la raya es un 0. Para representar un espacio utiliza la secuencia 1100. Dado que en un número de 32 bits se pueden transmitir varias letras, toda tarjeta comienza con dos bits que indican la cantidad de letras incluidas. Dicha cantidad es representada con la misma codificación anterior.



Con dicha codificación, una tarjeta con un número 26 representa el texto "A", ya que en binario es 011010. Los dos bits menos significativos son 00 indicando que almacena una única letra, los dos bits siguientes son 01 indicando que la letra se codifica con dos símbolos y los dos bits siguientes son 10 indicando punto-raya, que es la codificación de la letra A. Con la misma codificación el número 3596799 (en binario 1101101110000111111111) representa al texto "HOLA".

Entrada

La entrada comienza con la cantidad de casos de prueba C a evaluar. Por cada caso de prueba se ingresa un número T que indica la cantidad de tarjetas a leer. A continuación se reciben T números enteros menores a 2^{31}.

Salida

Por cada caso de prueba se debe mostrar una línea con el mensaje original decodificado. En los casos que contengan más de un número se deben mostrar todos sus caracteres concatenados

Ejemplo

Entrada

```
3
1
26
1
3596799
2
16813
3401607
```

Salida

```
A
HOLA
COMO VA
```

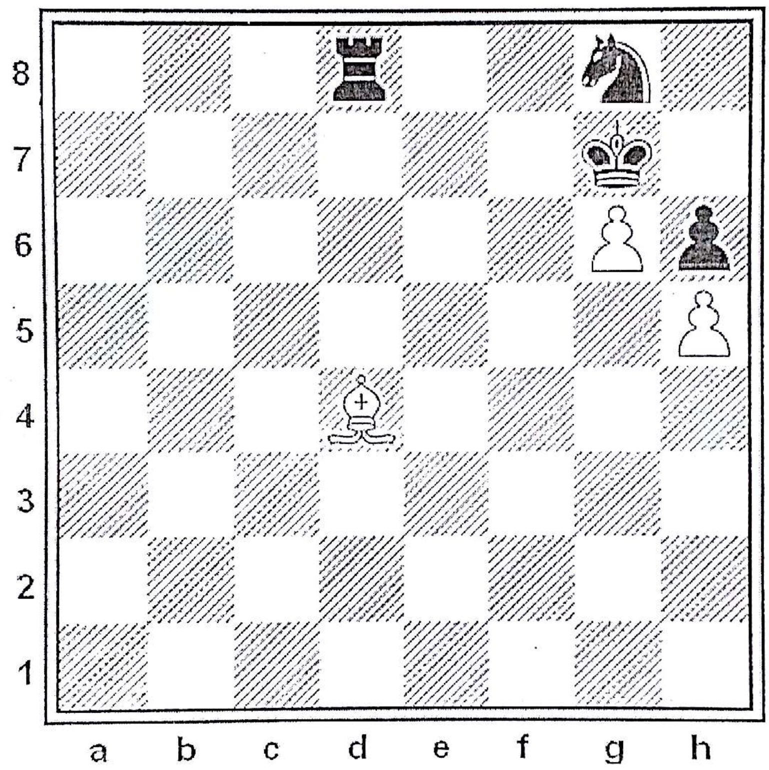
11010 → 01|0110|000

MAL

Problema 4: Jaque al Rey

El problema consiste en determinar si en una partida de Ajedrez, en un determinado instante de juego, el rey de las Blancas o el Rey de las Negras se encuentran en Jaque o bien determinar si no se dá ninguna de las dos condiciones anteriores. Se asume que en el tablero es de 8×8 casillas como es el juego en realidad y que las piezas se mueven como de costumbre.

El tablero será informado como entrada al problema, indicando primero las posiciones de las fichas Blancas y luego las posiciones de las fichas Negras. Las piezas se indicarán como P para el Peón, A para Alfil, D para Dama, T para Torre, C para Caballo y R para Rey. Las coordenadas en el tablero de ajedrez se corresponden con la siguiente figura:



Entrada

La entrada inicia con un valor entero indicando la cantidad de tableros a procesar.

A continuación, vendrá una letra B o N para indicar si las siguientes líneas son posiciones de las fichas Blancas o Negras, respectivamente.

Luego en cada línea vendrá la posición de una pieza de la forma X LN, donde X es la inicial del nombre de la pieza, L es la posición (A-H) y N es en número (1-8).

Cada uno de los tableros de entrada a procesar vendrán separados por una línea en blanco.

Salida

Por cada caso de prueba se debe indicar “JAQUE DE BLANCAS”, “JAQUE DE NEGRAS” o “SIN JAQUE”.

Ejemplo

Entrada

```
1
B
A D4
P G6
P H5
N
T D8
C G8
R G7
P H6
```

Salida

```
JAQUE DE BLANCAS
```

Problema 5: El colocador de cerámicos

Cuando se instalan cerámicos en pisos y paredes durante la construcción de una casa, debe comprarse una cantidad precisa de cajas considerando la superficie a cubrir y estimando el desperdicio que se genera por los recortes y las roturas.

Un colocador de cerámicos necesita un software que le estime la cantidad de cajas a comprar dadas las dimensiones del ambiente y de cada pieza a instalar. Como normalmente las longitudes de las paredes no son múltiplos exactos de cada cerámico, en los bordes deben realizarse cortes para cubrir toda la superficie. Para ahorrar piezas, el instalador puede dividir una pieza para colocarla en dos lugares diferentes cuando la superficie a cubrir sea menor al 50 % de la superficie de la pieza. Se puede suponer que aunque sea materialmente posible, no utiliza una misma pieza para cubrir más de dos lugares y que sólo utiliza piezas cuadradas.

Entrada

La entrada inicia con una línea con la cantidad de casos de prueba C. Luego se presentan C casos, cada uno correspondiendo a un ambiente a construir. Por cada caso se reciben cuatro líneas conteniendo en orden:

- Ancho del piso (A), expresado en centímetros, $0 < A < 10000$.
- Largo del piso (B), expresado en centímetros, $0 < B < 10000$.
- Lado de cada pieza (L), expresado en centímetros, $15 < L < 60$.
- Cantidad de piezas por caja (Q), $6 < Q < 30$.

Salida

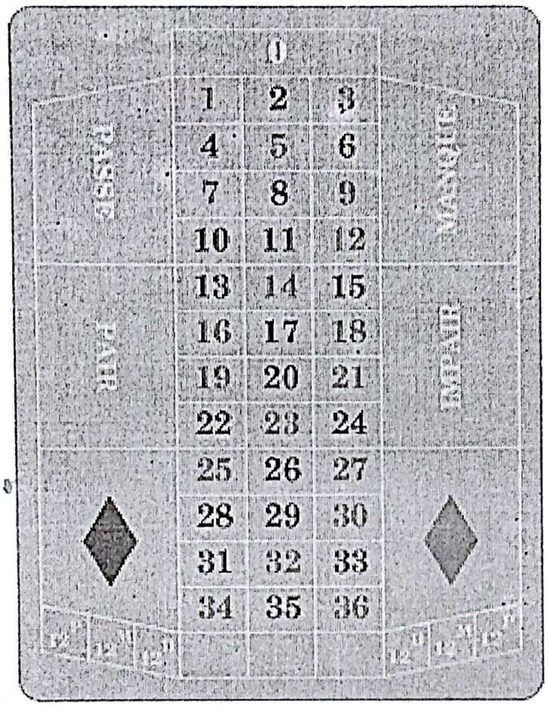
Por cada caso se debe presentar una línea con un entero que indique la cantidad de cajas a comprar.

Ejemplo

Entrada	Salida
2 500 600 36 18 500 600 26 16	14 30

Problema 6: Martingala

La martingala es una técnica nacida en Francia alrededor del s XVIII, la cual busca obtener de forma ingenua (creyendo que esta técnica de juego los hará millonarios) beneficios jugando a la ruleta.



La estrategia de esta técnica consiste en, creyendo falsamente que hay aproximadamente 50 % probabilidad de ganar, al momento de incurrir en una pérdida, volver a apostar por el doble del total perdido a la misma jugada. En la nueva apuesta, el jugador tiene la posibilidad de recobrar todas sus pérdidas, más una ganancia.

Con esta estrategia podría parecer que a largo plazo existe esperanza de ganancia, creyendo falsamente que la probabilidad de obtener ganancias se mantienen constantes y a favor del jugador.

Esto no es cierto, debido a dos factores:

- El jugador cuenta con recursos finitos, es decir no posee el suficiente dinero para apostar continuamente.
- La ruleta define una apuesta máxima por jugada, es decir que si se contara con recursos infinitos, llegaríamos a un techo en la apuesta, por ende nunca recuperar lo apostado.

Por ello, es necesario que usted genere un simulador de apuestas el cual permita verificar que la martingala lleva a la quiebra rápidamente a los jugadores.

Sistema de Juego

- El jugador solo puede apostar a color (rojo, negro), o paridad (par o impar), pero siempre a lo mismo.
- El jugador inicia con la apuesta mínima.
- Si incurre en pérdida, siempre dobla la apuesta.
- Si el jugador gana, la siguiente apuesta la realiza por el mismo monto que la última.
- Si sale 0 (cero) el jugador pierde la apuesta, ya que este valor no pertenece a ningún color, y a ningúna paridad.
- La casa de juego tiene definida una apuesta máxima.
- El jugador cuenta con un presupuesto finito.
- Si el jugador no dispone de dinero suficiente para la siguiente apuesta, realiza una con el saldo que disponga.
- El jugador se considera quebrado cuando su dinero se hace igual a 0.

Se quiere saber cual es el dinero con el que contará el jugador dada una cierta cantidad de tiradas de ruleta y si el mismo termina quebrado.

Su programa debe basarse en la siguiente tabla de números de 50 números aleatorios para realizar la simulación de los lanzamientos de la ruleta:

[4 24 12 25 8 14 15 19 12 17 29 17 21 21 23 0 34 26 10 8 29 4 29 29 17 27 33 16 26 7 29 28 21 21 17 25 21 20 13 30 4 0 8 16 22 15 14 22 31 13]

Entrada

La entrada comienza con un número entero indicando la cantidad de casos de prueba que se desean procesar.

Luego, por cada caso de prueba se indica un número entero entre 0 y 49 indicando el primer número de la tabla de jugadas que deberá usarse para comenzar la simulación. Por ejemplo si se indica 2, esto significa que el primer número a usarse debe ser 12, luego 25, y así sucesivamente. Luego de la última jugada (valor de 13) asuma que la próxima jugada de la ruleta es nuevamente 4, 24, y así sucesivamente.

A continuación, y en una nueva línea, se indica la apuesta (paridad o color), y luego se indica la estrategia elegida para esa apuesta. Seguidamente se indica el presupuesto (dinero que trajo para apostar), y el valor de la apuesta máxima de la mesa. La apuesta siempre inicia con $1.

Salida

Se debe indicar la cantidad de jugadas que ha podido realizar el apostador antes de terminar quebrado.

Ejemplo

Entrada	Salida
2 0 paridad impar 1000 64000 2 color rojo 5000 20000	41 35