(ADN) CSI, en busca del ADN perdido

Descripción del problema

El ADN guarda la información genética de cada individuo. Podemos representarlo como una secuencia de bases. Estas bases son A (adenina), T (timina), C (citosina), G (guarina). De esta forma, una cadena de ADN se puede ver como una secuencia de caracteres.

Dicha cadena se puede representar como un vector de caracteres:

A	T	G	C	G	T	A	T

Hay que implementar un programa que compare dos secuencias de ADN e indique si son de la misma persona.

Debemos tener en cuenta que la secuencia de ADN es cíclica, por lo que puede comenzar en cualquier posición. Las cadenas se leen de izquierda a derecha y cuando terminamos volvemos a empezar a leer por el principio. Por ejemplo las dos secuencias siguientes coinciden, es decir, son de la misma persona.

A	T	g	C	G	t	a	T

A	T	a	T	G	C	G	T

Las dos bases sombreadas corresponden al inicio de la secuencia. Pueden venir en mayúsculas o minúsculas.

Es posible que las secuencias de ADN estén contaminadas, con otros elementos o con las mismas bases. Lo importante para decidir si dos secuencias son de la misma persona es que la secuencia de una cadena de ADN contenga a la otra y que la longitud sea como mínimo la indicada. Por ejemplo:

Para una longitud de 4 podríamos tener las siguientes secuencias de ADN

aTCg	BGABtCB	tCga	atcgw	aattatcgw	cgaaTchhAT

Todas las secuencias anteriores se consideran de la misma persona, es decir, son iguales, aunque hay algunas que están contaminadas. Si nos fijamos, la parte resaltada aparece para todas las secuencias de forma consecutiva-cíclica.

Las siguientes secuencias con una longitud de 4 son todas de diferentes personas.

aTCg	GAtzCC	Cgxax	Atc	aattacgw	cgvT

Tened en cuenta que para una longitud menor a la indicada, directamente se descarta la secuencia.

También hay que tener en cuenta la forma en la que está contaminada una secuencia. Imaginemos que tenemos la secuencia de 5 elementos ACTGC

Contaminaciones:

ACTG→ menor número de elementos→INCORRECTO

ACTGBCB→ otros elementos distintos a las 4 bases, en este caso B. La secuencia sería igual a la original ACTGC ya que podemos quitar la contaminación.

CCACTGCTTT→ contaminación con bases que no rompen la secuencia. Estaría correcto y SI sería igual que la original ACTGC.

ACTGACA→ contaminación con bases que rompen la secuencia. Esta secuencia NO sería igual que la original ACTGC.

Entrada

Para comparar pares de secuencias de ADN necesitaremos saber tres cosas que leeremos en tres líneas diferentes:

- Una primera línea indicando la longitud de las cadenas ADN a comparar. Las cadenas tendrán una longitud de entre 0 y 36 elementos.
- Dos líneas con las secuencias de ADN con valores de bases *A* (adenina), *T* (timina), *C* (citosina), *G* (guarina). Los valores podrán venir dados en mayúsculas o en minúsculas, es decir, por ejemplo la base adenina puede venir dada por la letra ***a*** o ***A***. Igual para el resto de bases.

Analizaremos secuencias de ADN hasta que leamos alguna que tenga una longitud de *0*.

Salida

Se indicará:

- **SI** si las dos secuencias son iguales
- **NO** si las dos secuencias no son iguales
- **INCORRECTO** en el caso de que cualquiera de las dos cadenas tenga una longitud menor a la indicada

Entrada de ejemplo

```
8
ATGCGTAT
ATATGCGT
10
TGCGTTCAAT
ATATCGCAAT
8
ATGCGTT
ATATGCGT
9
ASTACGTAGC
CATACGTAG
0
```

Salida de ejemplo

```
SI
NO
INCORRECTO
SI
```

(CAJERO) Cajero Automático

Descripción del problema

Los cajeros disponen de una cantidad de dinero que ponen a disposición de sus clientes distribuidos en una serie de billetes diferentes. Nuestro cajero va a disponer billetes de 500, 100, 50, 20 y 10. El problema tratará de dar solución a las peticiones de los clientes usando el menor número de billetes posible.

Entrada

El problema en cuestión va a recibir una línea de entrada en la que se recogerán, separado por espacios, las peticiones de dinero de los clientes, terminado un 0.

Salida

Se mostrará, para cada petición de dinero, una línea en la que se mostrarán 5 números indicando la cantidad de billetes de 500, 100, 50, 20, 10, (en ese mismo orden), necesarios para atender la petición. En caso de no poder calcular la cantidad de dinero solicitada con los billetes disponibles, se mostrará un cero.

Entrada de ejemplo

```
250 420 580 120 235 633 1890 10 660 0
```

Salida de ejemplo

```
0 2 1 0 0
0 4 0 1 0
1 0 1 1 1
0 1 0 1 0
0
0
3 3 1 2 0
0 0 0 0 1
1 1 1 0 1
```

(CENTRONU) Centro Numérico

Descripción del problema

Un **centro numérico** es un número que separa una lista de números enteros (comenzando en 1) en dos grupos de números, cuyas sumas son iguales.

El primer centro numérico es el 6, que separa la lista (1,8) en dos grupos: (1,2,3,4,5) y (7,8), cuyas sumas son iguales a 15.

El segundo centro numérico es el 35, que separa la lista (1, 49) en dos grupos: (1 a 34) y (36 a 49), cuyas sumas son iguales a 595.

Se pide un programa que dado un número entero positivo, determine la lista de centros numéricos existentes entre 1 y el número dado.

Entrada

El programa recibirá una cantidad determinada de números enteros positivos, iguales o menores a 50000. Termina cuando la entrada sea -1.

Salida

Para cada entrada, el programa mostrará la lista de números céntricos existentes entre 1 y el número dado. Cuando haya más de un número céntrico, se separarán por guiones (sin espacios). Cuando no haya ningún número céntrico en el intervalo dado, se mostrará el mensaje `NO HAY NINGUN NUMERO CENTRICO`.

Entrada de ejemplo

```
1
13
5
150
1326
824
-1
```

Salida de ejemplo

```
NO HAY NINGUN NUMERO CENTRICO
6
NO HAY NINGUN NUMERO CENTRICO
6-35
6-35-204-1189
6-35-204
```

(CICLISTA) Club ciclista

Descripción del problema

Un club ciclista anota semanalmente los km de entrenamiento por cada día de salida de su grupo de corredores. Si un día el club ciclista no sale se apunta 0 km, esa jornada se considerará una jornada de descanso:

a. Mostrar si en todas las salidas se han sobrepasado los 100 km.
b. Media de km diarios, sin computar los días de descanso. El resultado se trunca al entero más bajo, es decir, **4.9 = 4**
c. Diferencia máxima de km entre todas las salidas (sin contar los días de descanso).
d. ¿Cuál fue el intervalo de días con más jornadas de descanso consecutivos? Hay que mostrar el número de días.
e. Comprobar si existe una progresión ascendente (en km) de todas los días de la semana, incluido los días de descanso. Se considera que existe progresión si el día siguiente se hacen **más** kilómetros que el anterior. Y así para toda la semana.

Entrada

Un listado de 7 números que indican los Km realizados cada día de una semana. En una misma línea con una separación de un espacio en blanco entre número y número, con valor mínimo de 0.

Tendremos tantas líneas como semanas a analizar. Dejaremos analizar semanas cuando leamos la palabra ***fin***

Salida

En una misma línea hay que anotar los resultados de los apartados separados por espacios en blanco.

En el apartado a) si es correcto mostramos ***1***, si es falso ***0***

En el apartado b) un valor entero con la media aritmética. Truncado al entero menor.

En el apartado c) un entero con la diferencia entre la salida con más kilómetros y la que menos (sin contar los días de descanso).

En el apartado d) hay que mostrar el máximo número de días consecutivos de descanso. ***0*** si hay salidas todos los días.

En el apartado e) se muestra un valor ***1*** si existe una progresión ascendente de km en todos los días de la semana, si no es así se mostrará un ***0***.

Entrada de ejemplo

```
0 0 0 0 0 0 0
0 0 10 10 0 0 0
23 34 56 77 99 100 111
110 115 122 129 131 132 133
99 99 100 100 110 110 123
55 59 0 0 0 99 44
44 32 21 0 0 99 0
100 101 102 103 104 105 106
96 45 112 41 82 94 42
75 245 164 186 43 19 62
fin
```

Salida de ejemplo

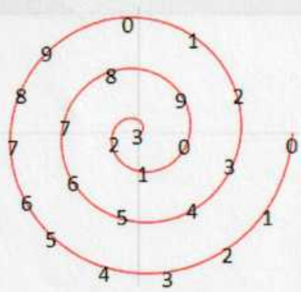
```
0 0 0 7 0
0 10 0 3 0
0 71 88 0 1
1 124 23 0 1
0 105 24 0 0
0 64 55 3 0
0 49 78 2 0
1 103 6 0 1
0 73 71 0 0
0 113 226 0 0
```

(ESPIRAL) Espiral numérica

Descripción del problema

Según el diccionario de la Real Academia de la Lengua Española una ***espiral*** es *"Curva plana que da indefinidamente vueltas alrededor de un punto, alejándose de él más en cada una de ellas."*

Podríamos introducir en la línea de la espiral una secuencia repetitiva de números entre 0 y 9 comenzando desde fuera de la espiral, de tal manera que la espiral quedase de la siguiente forma:



Podríamos representar una espiral numerada similar a la anterior en un cuadrado de la siguiente forma:

0	1	2	3	4
5	6	7	8	5
4	3	4	9	6
3	2	1	0	7
2	1	0	9	8

Llamaremos a esta espiral, **Espiral Numérica Cuadrada (ENC)**.

Entrada

Recibiremos pares de números representando el número de lados del cuadrado, (entre ***1*** y ***200***), y el número en el que comienza la espiral, (entre ***0*** y ***9***). La espiral siempre comenzará en la casilla superior izquierda de la ***ENC***. En el ejemplo visto en la descripción del problema el cuadrado es de ***5*** lados y la espiral comienza en el número ***0***.

Dejaremos de analizar espirales cuando recibamos el par ***0 0***

Salida

La salida mostrará los números de cada línea de la *ENC* separados por espacios en blanco. Cada línea de la *ENC* se mostrará en líneas diferentes.

Entrada de ejemplo

```
5 0
3 7
4 5
0 0
```

Salida de ejemplo

```
0 1 2 3 4
5 6 7 8 5
4 3 4 9 6
3 2 1 0 7
2 1 0 9 8
7 8 9
4 5 0
3 2 1
5 6 7 8
6 7 8 9
5 0 9 0
4 3 2 1
```

(LABERINT) Laberinto

Descripción del problema

Encontrar la salida del laberinto.

Un laberinto se puede representar por una matriz en la que los 1's representan que existe camino y los 0's que no existe.

	0	1	2	3	4	5	6	7	8
0	1	1	0	0	0	1	1	1	1
1	0	1	1	1	1	1	0	0	0
2	0	1	0	0	0	1	0	0	0
3	1	1	0	0	0	1	0	1	0
4	0	1	0	0	0	1	0	1	0
5	1	1	0	0	0	1	0	1	0
6	1	0	0	0	1	1	1	1	1
7	1	1	1	1	1	0	0	0	0

Entrada

Salida

Nuestro laberinto será un laberinto especial:

Especificaremos un punto de ***entrada*** y un punto de ***salida***, sólo se podrá entrar y salir por esos puntos. En el ejemplo, la entrada sería el punto ***(3,0)*** y la salida sería la ***(6,8)***. Se nos garantiza que únicamente existe un camino posible y que el siguiente punto en el camino **NUNCA** estará en una columna posterior, es decir, podemos ir de izquierda a derecha, de arriba abajo y de abajo arriba, pero **NUNCA** de derecha a izquierda. Como vemos el camino marcado en **rojo** no es posible pues en un momento dado va hacia atrás, es decir, de derecha a izquierda. El resto de caminos marcados en **azules**, o bien no tienen el punto marcado como entrada correcta o no tienen el punto marcado como salida correcta.

No se permitirá avanzar en diagonal.

Entrada

Recibiremos una primera línea con puntos de entrada y salida. Para el ejemplo anterior vendría dado de la siguiente forma: **3,0-6,8** es decir el **3,0** sería el punto de entrada y el **6,8** sería el de salida con una coma entre ambos números. Estos dos puntos estarán separados por un guión medio.

Una segunda línea en la que nos indicarán el tamaño de la matriz, en el ejemplo anterior sería **8-9** es decir 8 filas y 9 columnas. Ambos números separados por un guión medio.

En las siguientes líneas se mostrarán los 1's y 0's que conforman la matriz.

Dejaremos de analizar caminos cuando nos lleguen los puntos **0,0-0,0**

Salida

Una línea en la que, para cada matriz a analizar mostraremos la ruta seguida para ir desde la entrada hasta la salida. Para el ejemplo anterior se mostraría:

3,0--3,1--2,1--1,1--1,2--1,3--1,4--1,5--2,5--3,5--4,5--5,5--6,5--6,6--6,7--6,8

Los dos números de cada punto separados por comas y los distintos puntos separados por dos guiones medios.8

Existe la posibilidad de que en una matriz dada, y dados también los puntos de entrada y salida no exista un camino posible. En esos casos se mostrará la salida ***NO EXISTE CAMINO***

Entrada de ejemplo

```
3,0-6,8
8-9
1 1 0 0 0 1 1 1 1
0 1 1 1 1 1 0 0 0
0 1 0 0 0 1 0 0 0
1 1 0 0 0 1 0 1 0
0 1 0 0 0 1 0 1 0
1 1 0 0 0 1 0 1 0
1 0 0 0 1 1 1 1 1
1 1 1 1 1 0 0 0 0
2,0-0,2
3-3
0 1 1
0 1 0
1 1 0
0,0-4,4
5-5
1 1 1 1 0
0 1 0 1 0
1 1 0 1 0
1 0 0 1 0
1 1 1 1 0
0,0-4,5
6-6
1 1 1 1 0 0
0 1 0 1 0 0
1 1 0 1 1 1
1 1 0 0 0 0
1 1 0 1 1 1
0 1 1 1 0 0
0,0-0,0
```

Salida de ejemplo

```
3,0--3,1--2,1--1,1--1,2--1,3--1,4--1,5--2,5--3,5--4,5--5,5--6,5--6,6--6,7--6,8
2,0--2,1--1,1--0,1--0,2
NO EXISTE CAMINO
0,0--0,1--1,1--2,1--3,1--4,1--5,1--5,2--5,3--4,3--4,4--4,5
```

Salida

Una línea en la que, para cada matriz a analizar mostraremos la ruta seguida para ir desde la entrada hasta la salida. Para el ejemplo anterior se mostraría:

3,0--3,1--2,1--1,1--1,2--1,3--1,4--1,5--2,5--3,5--4,5--5,5--6,5--6,6--6,7--6,8

Los dos números de cada punto separados por comas y los distintos puntos separados por dos guiones medios.8

Existe la posibilidad de que en una matriz dada, y dados también los puntos de entrada y salida no exista un camino posible. En esos casos se mostrará la salida ***NO EXISTE CAMINO***

Entrada de ejemplo

```
3,0-6,8
8-9
1 1 0 0 0 1 1 1 1
0 1 1 1 1 1 0 0 0
0 1 0 0 0 1 0 0 0
1 1 0 0 0 1 0 1 0
0 1 0 0 0 1 0 1 0
1 1 0 0 0 1 0 1 0
1 0 0 0 1 1 1 1 1
1 1 1 1 1 0 0 0 0
2,0-0,2
3-3
0 1 1
0 1 0
1 1 0
0,0-4,4
5-5
1 1 1 1 0
0 1 0 1 0
1 1 0 1 0
1 0 0 1 0
1 1 1 1 0
0,0-4,5
6-6
1 1 1 1 0 0
0 1 0 1 0 0
1 1 0 1 1 1
1 1 0 0 0 0
1 1 0 1 1 1
0 1 1 1 0 0
0,0-0,0
```

Salida de ejemplo

```
3,0--3,1--2,1--1,1--1,2--1,3--1,4--1,5--2,5--3,5--4,5--5,5--6,5--6,6--6,7--6,8
2,0--2,1--1,1--0,1--0,2
NO EXISTE CAMINO
0,0--0,1--1,1--2,1--3,1--4,1--5,1--5,2--5,3--4,3--4,4--4,5
```

(LOTERIA) Ordenación por apariciones

Descripción del problema

En un sorteo de lotería se realizan 100 extracciones de bolas cuyos valores están comprendidos entre el 0 y el 9. Después de realizar una extracción la bola se reintegra de nuevo al bombo de extracciones. Debemos hacer una ordenación (de mayor a menor) por número de repeticiones. Además se visualizará el número de repeticiones. En caso de igualdad se tiene preferencia el valor más alto.

Entrada

Líneas con los resultados las 100 extracciones. Analizaremos cada sorteo y dejaremos de analizarlos cuando encontremos la línea con la palabra ***fin***

Salida

Listado ordenado de mayor a menor, según el número de repeticiones, de los 100 valores resultantes, acompañado del número de repeticiones. En caso de empate se mostrarán los números extraídos de mayor a menor.

Entrada de ejemplo

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
5 3 5 0 7 5 5 9 7 1 6 8 3 4 9 2 7 7 9 0 8 7 8 2 3 0 6 3 9 6 1 1 9 7 1 5 9 9 2 2 8 9 7 6
2 2 8 9 6 4 8 1 9 3 4 7 0 4 5 4 1 0 2 2 1 4 4 7 6 3 3 9 0 4 7 6 6 4 9 3 1 6 2 8 2 2 4 6
3 0 6 3 8 6 0 1 2 9 9 9
fin
```

Salida de ejemplo

```
0 100
9 0
8 0
7 0
6 0
5 0
4 0
3 0
2 0
1 0
9 15
6 12
2 12
7 10
4 10
3 10
1 9
8 8
0 8
5 6
```

(LUCHA) La lucha japonesa

Descripción del problema

Una academia dedicada a las difíciles artes de lucha oriental, está organizando un torneo de luchadores de sumo. A los efectos de conformar la parrilla de combates, es necesario armar parejas de luchadores que puedan ofrecer un espectáculo atractivo. Como es sabido que los deportistas que practican esta disciplina son hombres de estructura física imponente, los organizadores habitualmente arman duplas de combatientes estableciendo comparaciones relativas a sus alturas y pesos. Por experiencia, los organizadores saben que un combatiente domina a otro si lo supera en ambas medidas, o bien si lo iguala en peso y lo supera en altura, o viceversa. En cualquier otro caso, los luchadores no son comparables lo que hace el resultado imprevisible y por lo tanto más atractivo para el público. Sabiendo que no hay luchadores que coinciden en ambas medidas, los organizadores quieren saber a cuantos posibles contrincantes domina cada uno de ellos, y por esta razón se pide que se confeccione un programa que efectúe este recuento.

Entrada

La entrada constará de:

- Una primera línea que indica la cantidad L de luchadores. El valor estará comprendido entre 2 y 100000
- Una serie de L líneas con 2 números P y H que indican el peso en gramos y la altura en centímetros de cada atleta.

Salida

La salida reflejara una sucesión de L líneas con la cantidad de luchadores a quienes domina cada participante, en el mismo orden en el que los participantes entraron.

Entrada de ejemplo

```
10
3000 1500
3200 1500
2990 1580
3300 1690
3300 1540
3390 1500
2980 1700
3440 1500
2760 1678
2890 1499
```

-cout ("%d%d", &H &P)

Salida de ejemplo

```
1
2
1
6
3
3
2
4
0
0
```

(MULTIPLO) Múltiplos

Descripción del problema

Un número ***a*** es múltiplo de otro ***b*** cuando es el resultado de multiplicarlo por otro número ***c***.

18 es múltiplo de ***2***, ya que resulta de multiplicar ***2*** por ***9***.

$$18 = 2 \cdot 9$$

18 es múltiplo de ***3***, ya que resulta de multiplicar ***3*** por ***6***.

$$18 = 3 \cdot 6$$

Por el ejemplo anterior podemos determinar que un número **n** puede ser múltiplo de varios números. El 18 es múltiplo de 1, 2, 3, 6 y 9

Dadas dos listas de números mostraremos qué números de la primera lista tienen, al menos, 20 múltiplos en la segunda lista.

Este proceso lo repetiremos hasta leer dos listas que sólo tengan un cero cada una.

Entrada

Recibiremos pares de listas entre 1 y 10000 números enteros positivos separados por un espacio en blanco.

Al final tendremos 2 listas en las que en cada una sólo hay un cero. Esta será la condición para dejar de analizar listas.

Salida

Mostrará, para cada par de listas los números distintos de la primera lista que cumplen tener, al menos, 20 múltiplos en la segunda lista.

En caso de no haber ninguno que cumpla la condición, se mostrará la palabra **NINGUNO**.

Entrada de ejemplo

```
15 10 2 20 37 1 5 1 5 10
2 30 20 5 6 8 15 25 15 10 20 50 80 90 5 2 15 1 25 55 65 10 10 10 10 15 15 40 60 80
1 5 2 30 58 15
2 5 5 5 15 20 35 52
1 5 2 30 58 15 1
2 5 5 5 15 20 35 52 2 327 15 7 9 15 1 2 3 8 9 20
0
0
```

Salida de ejemplo

```
1 5
NINGUNO
1
```

(TRAGAPER) Máquina tragaperras

Descripción del problema

Una máquina tragaperras genera combinaciones de 6 números de forma aleatoria en cada partida. En vez de mostrar iconos que representan imágenes, por ejemplo fruta es lo más habitual, van a ser números comprendidos entre el 0 yel 9. Los premios pueden ser de 5 tipos:

a. Que salgan todos iguales.

b. Escalera ascendente. Siempre el siguiente será mayor que el anterior. De izquierda a derecha.

c. Simétrico. Da lo mismo leerlos de izquierda a derecha que de derecha a izquierda

d. Tres parejas de números diferentes. No tienen que estar consecutivas.

e. Todos diferentes

Entrada

Seis números, comprendidos entre el ***0*** y el ***9***, mostrados en una misma línea separados por un espacio en blanco.

Dejaremos de analizar entradas cuando leamos la palabra ***fin***.

Salida

Si se cumple uno de los apartados se muestra ***1***, si no es así ***0***. En una misma línea con un espacio en blanco de separación entre cada número.

Entrada de ejemplo

```
1 2 3 4 5 6
5 5 5 5 5 5
9 4 5 5 4 9
2 2 4 4 6 6
3 4 5 4 3 5
1 2 3 4 5 6
2 2 2 2 2 2
2 3 3 2 3 2
1 2 3 4 5 6
4 4 4 4 4 4
fin
```

Salida de ejemplo

```
0 1 0 0 1
1 0 1 0 0
0 0 1 1 0
0 0 0 1 0
0 0 0 1 0
0 1 0 0 1
1 0 1 0 0
0 0 0 0 0
0 1 0 0 1
1 0 1 0 0
```