

Programame

Ctrl Alt Supr

Concurso Regional Programame
Zaragoza

15 de Marzo de 2017

Cuaderno de Problemas



Índice de problemas

A. Filtro anti-spam	3
B. Crisis energética	4
C. Bomberman	5
D. El DNS	6
E. El organizador de paquetes	7
F. La calculadora romana	8
G. Partido de tenis	9
H. La media maratón de Zaragoza	10
I. Póker para dummies	11

Ejercicios realizados por:

- José Manuel Guallar (Colegio Montessori)
- Fernando Valdeón (Colegio Montessori)
- Santiago Faci (Colegio Montessori)



A. Filtro anti-spam

Con el aumento del SPAM en nuestros buzones de correo electrónico, se hacen necesarias medidas más severas para luchar contra todos los correos no deseados que nos llegan cada día.

Así, debes diseñar un filtro anti-spam que asigne una puntuación a los correos que llegan al servidor de forma que cuantos más puntos tenga un correo electrónico más probable es que sea SPAM. Así, otros sistemas se encargarán de clasificarlo como tal.

En tu caso, eres el responsable de fijarte en tres parámetros y valorarlos como sigue:

- Hay que comprobar si la dirección del remitente (*from*) no está en la lista negra
- Hay que comprobar que el remitente (*from*) sea una dirección de correo electrónico válida
- Hay que comprobar que el texto del correo no contenga ninguna palabra no permitida

Una dirección de correo se considera válida si contiene un nombre de buzón (un texto cualquiera), seguido del carácter @ y termina con un dominio (.com, .es, . . .)

Entrada

Como entrada se recibe una línea con los dominios que están en una lista negra seguido de otra línea con las palabras no permitidas en los textos de los correos electrónicos. Finalmente se reciben un número que indica el número de correos a procesar seguido de los propios correos.

Cada correo tiene 4 partes:

- *mail*: Es el código que se utiliza para identificar cada correo
- *from*: Sirve para indicar el remitente de dicho correo
- *to*: Indica el destinatario del correo
- La última parte contiene el texto del correo

Salida

Como salida se obtiene, para cada correo electrónico su puntuación total:

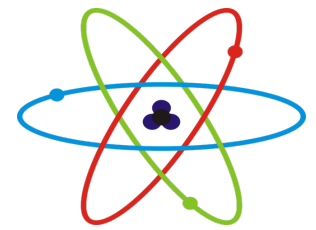
- Se asignan 3 puntos si la dirección de correo no es válida
- Si la dirección de correo electrónico del remitente es válida se comprueba si está en lista negra. En caso de que esté se le asignan 10 puntos
- Si el correo contiene, como texto, alguna palabra no permitida se añade 1 punto a su puntuación

Entrada de ejemplo

```
hotmail.com,loquesea.com,dominioconvirus.com
spam,soyspam,estoesspam,virus
4
mail,from:josemanuel@hotmail.com,to:santi@midominio.com,hola santi soy yo
mail,from:fernando@midominio.es,to:josemanuel@hotmail.com,hola que tal
mail,from:micorreo@gmail,to:josemanuel@hotmail.com,yo soy un virus
mail,from:soyunvirus@hotmail.com,to:josemanuel@hotmail.com,hola virus
```

Salida de ejemplo

```
mail 1 spam score = 10
mail 2 spam score = 0
mail 3 spam score = 4
mail 4 spam score = 11
```



B. Crisis energética

Con la subida del precio de la energía se ha modificado el sistema de asignación de energía y ahora será el cliente quién podrá decidir a quién la compra directamente, siempre respetando el orden asignado por la administración.

Así, un cliente, cuando llega su turno, puede decidir comprar toda la energía disponible a la empresa suministradora más barata y sucesivamente a las menos baratas hasta satisfacer toda su demanda.

El sistema que debes diseñar busca también conocer cuál será el estado de las empresas suministradoras después de hacer el reparto, puesto que necesitan saber cuál será el excedente de energía al término de cada período para saber si deben generar más o menos y así maximizar su beneficio. Hay que tener en cuenta que las empresas suministradoras siempre tendrán, al menos, suficiente energía como para abastecer a todos los clientes que la soliciten.

Entrada

Como entrada se recibe un número que indica el número de empresas que ofrecen energía y a continuación cada empresa indica su nombre, su precio (Kw/h) y de cuanta energía dispone. Luego se recibe el número de clientes que demandan energía y a continuación, para cada uno de ellos, se indica su nombre y su necesidad de energía. Hay que tener en cuenta que el listado de empresas estará siempre ordenado por el precio de forma ascendente.

Salida

Como salida se muestra el excedente de energía que queda a cada empresa tras ser asignada toda la energía a los clientes que la demandaban.

Entrada de ejemplo

```
3
empresa1,10,1000
empresa2,20,1800
empresa3,30,10000
4
cliente1,500
cliente2,1000
cliente3,4000
cliente4,200
```

Salida de ejemplo

```
empresa1 0
empresa2 0
empresa3 7100
```

C. Bomberman



Llega el clásico juego de Bomberman a Programame. Se trata de una versión para que los alumnos puedan jugar en clase cuando se aburran. Se puede jugar con lápiz y papel aunque en este caso se trata de hacer un programa que permita comprobar que las partidas jugadas en clase son correctas.

Así, el juego consiste en dibujar un tablero (que será siempre de 6 filas x 6 columnas) de forma que se colocan aleatoriamente un número variable de enemigos (identificados como M), al jugador (identificado como J) y un número variable de bombas (identificadas como B). A continuación, hay que calcular cuántos enemigos serán alcanzados por la onda expansiva de todas las bombas, teniendo en cuenta que ésta alcanza a las casillas que se encuentran justo a continuación de donde se encuentra situada la bomba (en las 4 direcciones posibles, arriba, abajo, izquierda y derecha).

Hay que tener cuidado porque puedes perder la partida si la bomba alcanza a tu jugador.

Entrada

Como entrada se recibe un tablero de 6x6 con las casillas vacías marcadas como X y el resto de elementos del juego tal y como se indica en el enunciado.

Salida

Como salida se debe indicar un mensaje indicando cuántos enemigos se han eliminado en total haciendo explotar todas las bombas. En caso de que alguna bomba alcance a tu personaje se mostrará únicamente el mensaje *Has explotado. Has perdido la partida*

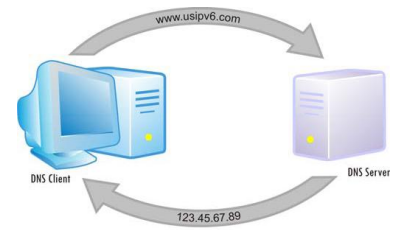
Entrada de ejemplo

```
X , X , X , M , X , X
X , X , X , X , X , X
B , X , X , X , X , X
M , J , X , B , X , X
X , X , X , M , X , X
X , X , X , M , X , X
```

Salida de ejemplo

```
Tus bombas han eliminado a 2 enemigos
```

D. El DNS



Un servidor DNS es un sistema que almacena pares dominio-IP de forma que nosotros podamos memorizar fácilmente nombres de dominio y sea este sistema quién tenga luego que traducirlo por la IP real del servidor al que queremos acceder. El sistema gestiona una gran base de datos donde se guarda la IP que corresponde a cada dominio existente en Internet.

En este caso también será responsable de notificar al usuario cuando éste esté accediendo a un dominio que no se encuentra registrado en el sistema.

Conviene saber que el usuario no está obligado a escribir, delante del dominio, la combinación 'www' para que la URL se considere correcta. Será el sistema quién tendrá que hacer que esto sea posible.

Entrada

Como entrada se obtiene un número que indica el número de pares dominio-IP que gestiona el servidor DNS de nuestro sistema. A continuación un número que indicará cuántas URLs se van a procesar seguido de las propias URLs.

Salida

Como salida se debe obtener, para cada URL procesada, la misma habiendo cambiado el dominio por la IP que corresponda según la tabla de conversión del sistema. En caso de que se emplee algún dominio que no reconozca nuestro servidor DNS se mostrará un mensaje de error como el que se muestra en la salida de ejemplo.

Entrada de ejemplo

```
3
programame.com,80.34.223.21
colegiomontessori.com,78.23.45.123
misuperpaginaweb.com,90.123.234.12
4
http://www.programame.com/regionales/zaragoza
http://www.colegiomontessori.com/
http://programame.com/equipos.php
http://www.misitioweb.com/index.html
```

Salida de ejemplo

```
http://80.34.223.21/regionales/zaragoza
http://78.23.45.123/
http://80.34.223.21/equipos.php
No se puede resolver el dominio misitioweb.com
```

E. Organizador de paquetes



Con la revolución del comercio electrónico, las empresas de mensajería se están poniendo las pilas con la creación de nuevos algoritmos que les permitan optimizar sus rutas de reparto para llevar a cabo más repartos en menos tiempo y así incrementar su beneficio y el buen servicio a sus clientes.

Así, se ha demostrado que se perdía mucho tiempo en la organización de pedidos de diferentes clientes que viven en la misma calle. En muchas ocasiones, al ser pedidos diferentes se entregaban a diferentes repartidores y se desperdiciaba tiempo puesto que dos de ellos acababan encontrándose en la misma calle haciendo un reparto.

Así, se quiere crear un nuevo sistema que evalúe en cada momento los pedidos que están listos para organizar y los agrupe por ciudad y calle indicando la cantidad de pedidos que hay para cada combinación.

Entrada

Como entrada recibe un número que indica el número de pedidos que se van a procesar. A continuación, para cada pedido se indica ciudad y calle.

Salida

Como salida se debe obtener, para cada ciudad y calle, cuántos pedidos pendientes de entrega ha encontrado el sistema.

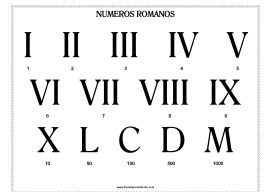
Hay que tener en cuenta que el orden de salida será el mismo en el que se ha encontrado un pedido para una ciudad y calle en la entrada del sistema.

Entrada de ejemplo

```
8
Zaragoza,Lagasca
Zaragoza,Miraflores
Zaragoza,Lagasca
Zaragoza,Miraflores
Pamplona,Plaza Mayor
Zaragoza,Miraflores
Pamplona,Plaza Mayor
Soria,Calle sin numero
```

Salida de ejemplo

```
Zaragoza,Lagasca,2
Zaragoza,Miraflores,3
Pamplona,Plaza Mayor,2
Soria,Calle sin numero,1
```



F. La calculadora romana

Recientemente se ha descubierto que los romanos ya disponían de calculadoras que les permitían hacer operaciones muy básicas. Estaban hechas de madera y permitían operar de una manera muy básica con algunos de sus números.

No eran muy completas puesto que su sistema de numeración resulta un tanto complejo pero permitían realizar casi la totalidad de los cálculos que necesitaban para el día a día.

Se ha descubierto que para llevar a cabo estas calculadoras se eliminaron todos los números que se componen utilizando la resta de símbolos como pueden ser el 9 (IX) o el 90 (XC). Por tanto, sólo cabía esperar poder operar con números que no necesiten ser compuestos de esta manera.

Se espera que seas capaz de simular correctamente el comportamiento de una de estas calculadoras romanas, aunque para este caso bastará con que lo hagas para la operación de sumar.

Entrada

Como entrada se obtiene una serie de parejas de número romanos.

Salida

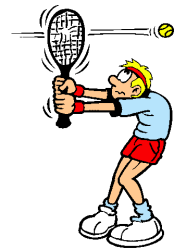
Como salida se obtendrá el resultado de la suma de los dos números de cada pareja en el sistema de numeración decimal.

Entrada de ejemplo

```
MMC, CCLIII
C, C
L, LVIII
II, III
```

Salida de ejemplo

```
2353
200
108
5
```

G. Partido de tenis

Tras la aparición de uno de los mejores avances en tecnología deportiva conocido como el ojo de halcón para sacar de dudas a los jueces a la hora de verificar el punto de un tenista, llega una nueva tecnología que podrá decidir el resultado de un partido sin necesidad de jueces humanos.

Se trata de un sistema que, cada vez que un tenista gana un juego, almacena su nombre. Es capaz de almacenar esta información para cada uno de los sets que dure un partido. Al final, el sistema es capaz de mostrar automáticamente el resultado del partido indicando quién de los dos tenistas lo ha ganado.

Hay que tener en cuenta que la tecnología está todavía en pruebas y sólo se están poniendo en marcha en partidos a 3 sets. El sistema funciona conforme a las reglas del tenis, que dicen que gana un set el primer tenista que llega a 6 juegos siempre y cuando aventaje a su adversario en más de un juego. En caso que no sea así se siguen disputando juegos hasta que uno de ellos aventaje al otro en 2.

Entrada

Como entrada se reciben 3 líneas que corresponden a los 3 sets en los que se compone un partido. En cada línea se indica quién ha ganado un juego, en el orden en que se ganó, mostrando su nombre.

Salida

Como salida se debe indicar el nombre del tenista que gana el partido.

Entrada de ejemplo

Nadal , Federer , Federer , Federer , Nadal , Federer , Nadal , Federer , Federer Nadal , Nadal , Nadal , Nadal , Nadal , Federer , Federer , Federer , Federer , Federer , Federer , Federer , Federer Nadal , Federer , Federer , Federer , Nadal , Federer , Nadal , Federer , Federer

Salida de ejemplo

Federer

H. La media maratón de Zaragoza



El otro día se organizó la XX Media Maratón de Zaragoza en el que se apuntaron más de 3000 corredores.

Tanto en la salida como en la llegada hay un control de tiempo de forma que se entrega a cada corredor un chip para controlar cuánto tiempo tarda en recorrer los poco más de 20 Kms que tiene la prueba. Se debe tener en cuenta que, debido a la cantidad de corredores que participan, habrá algunos de ellos que cruzarán la salida pasados algunos minutos con respecto a los que están en posiciones más adelantadas. Por ello, el sistema informático asociado al chip debe empezar a contar el tiempo en el momento en que un corredor cruza la línea de salida y no cuando el cronómetro se pone en marcha.

Pero este año ha habido un problema con el sistema informático y debes arreglarlo. Resulta que una vez terminada la prueba los encargados de preparar la clasificación se han encontrado con que el sistema ha guardado los tiempos de los corredores de forma totalmente aleatoria y además con respecto a la puesta en marcha del cronómetro. Por suerte han conseguido sacar también cuanto tardó cada corredor en pasar por la salida.

Hay que tener en cuenta que los tiempos de los corredores son siempre superiores a 1 hora e inferiores a 3 horas y que los chips que controlan el tiempo miden solamente horas y minutos.

Entrada

Como entrada se tiene un número que indica el número de participantes y a continuación, por cada uno de ellos, tendremos su nombre, su tiempo con respecto a la puesta en marcha del cronómetro seguido del tiempo (siempre en minutos) que tardó en cruzar la salida cuando empezó a correr.

Salida

Como salida debes mostrar el listado completo de corredores con su nombre y tiempo total, ordenado por éste.

Entrada de ejemplo

```
6
Jose Manuel:2,09:3
Fernando:2,10:5
Eduardo:2,00:1
Antonio:1,50:0
Sergio:1,10:0
Maria:1,20:0
```

Salida de ejemplo

```
Sergio:1,10
Maria:1,20
Antonio:1,50
Eduardo:1,59
Fernando:2,05
Jose Manuel:2,06
```



I. Póker para dummies

El otro día un grupo de amigos se disponía a jugar al póker por primera vez. Ninguno de ellos tenía ni idea y las reglas eran un tanto confusas así que uno de ellos se decidió a crear un programa que les ayudará a decidir cuál era la mejor jugada posible que podían formar con sus cartas y las que se colocan encima de la mesa.

Hay que tener en cuenta que este grupo de amigos jugaba a la modalidad Texas Hold'em por la que cada jugador tiene solamente dos cartas y se colocan encima de la mesa, boca arriba, otras 5, de forma que cada jugador puede combinar dichas cartas de la forma que más le convenga para formar la mejor jugada de 5 cartas que le sea posible.

Como este grupo de amigos no entendió muy bien las reglas se limitaron sólo a 5 jugadas (que se enumeran en orden de mejor a peor):

- COLOR: Cuando el jugador tenga 5 cartas de un mismo palo
- POKER: Cuando el jugador tenga 4 cartas iguales (mismo número, sin importa el palo)
- TRIO: Cuando el jugador tenga 3 cartas iguales (mismo número, sin importa el palo)
- PAREJA: Cuando el jugador tenga 2 cartas iguales (mismo número, sin importa el palo)
- UNA: El jugador no tiene nada. Jugará a su carta más alta

El palo de las cartas se indicarán mediante P(picas), T(tréboles), C(Corazones) y D(Diamantes) y hay que tener en cuenta se juega con baraja francesa (A,2,3,4,5,6,7,8,9,10,J,Q,K)

Entrada

Como entrada se recibe el número de jugadores, a continuación una línea con las cartas que hay encima de la mesa y, a continuación, la mano de cada uno de ellos (sus 2 cartas).

Salida

Como salida se tiene que obtener, para cada jugador, cuál es su mejor jugada posible de las que se contemplan en el enunciado.

Entrada de ejemplo

```
3
AD,AP,3P,10P,10T
10C,10T
3C,3P
2P,9P
KP,QT
```

Salida de ejemplo

```
Mejor jugada: POKER
Mejor jugada: TRIO
Mejor jugada: COLOR
Mejor jugada: PAREJA
```