

# Programame

Ctrl Alt Supr

Concurso Regional Programame  
Zaragoza

16 de Marzo de 2016

## Cuaderno de Problemas

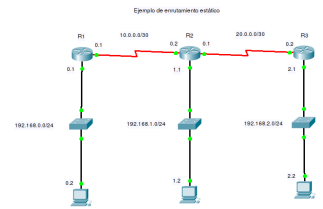


## Índice de problemas

<b>A.</b> Enrutando	<b>1</b>
<b>B.</b> El acuario de Zaragoza	<b>2</b>
<b>C.</b> Enigma II	<b>3</b>
<b>D.</b> Triple Town	<b>4</b>
<b>E.</b> El voto electrónico	<b>5</b>
<b>F.</b> El supercomputador	<b>6</b>
<b>G.</b> La Primitiva 2.0	<b>7</b>
<b>H.</b> Detector de plagios	<b>8</b>

Ejercicios realizados por:

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



## A. Enrutando

Los routers son dispositivos que interconectan diferentes redes entre sí para permitir la comunicación del tráfico de las mismas. Se conoce como tabla de enrutamiento a la memoria interna donde estos dispositivos almacenan la información de cómo redirigir ese tráfico. Básicamente indica por qué puerto o boca del router debe salir el paquete que se dirige a una red determinada.

En este caso, nos disponemos a programar uno de estos routers para que éste sepa qué hacer con cada uno de los paquetes que llegan a cada una de sus bocas de forma que pueda, enrutarlos correctamente, descartarlos si son para la red de donde proceden o bien generar un error si el paquete tiene como destino una red desconocida para él.

### Entrada

Como entrada se recibe la tabla de enrutamiento de un router indicando, para cada boca, el tráfico de qué red debe ser enrutado allí. A continuación se indican las ips destino (precedidas de un número que indica la cantidad de paquetes a procesar) de un número determinado de paquetes junto con la boca por la que vienen. Hay que tener en cuenta que el número de bocas de un router no es fijo y puede variar entre diferentes modelos.

### Salida

Como salida, para cada paquete, se debe indicar, si procede, el host y la boca por la que debe ser enrutado dicho paquete.

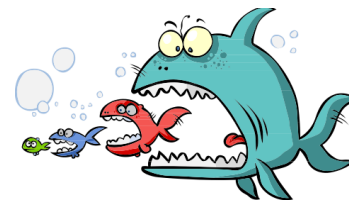
En caso de que la boca por la que el paquete deba salir sea la misma que por la que viene se descartará dicho paquete indicando 'paquete descartado'. Si la red para la que va dirigido un paquete no existe para el router actual, se indicará 'error'

### Entrada de ejemplo

```
1 192.168.201.0 2 192.168.202.0 3 192.168.203.0
6
1 192.168.202.23
1 192.168.201.34
2 192.168.203.34
3 192.168.201.123
2 192.168.202.212
3 192.168.210.212
```

### Salida de ejemplo

```
paquete para host 23 por boca 2
paquete descartado
paquete para host 34 por boca 3
paquete para host 123 por boca 1
paquete descartado
error
```



## B. El acuario de Zaragoza

El acuario de Zaragoza quiere introducir nuevas especies a sus instalaciones. Para evitar perder unidades de los nuevos peces que se incorporen, se quiere hacer un estudio acerca de cómo se comportan estos animales en convivencia.

De siempre se ha sabido que el pez grande se come al pez pequeño por lo que se trata de crear una simulación que permita ver cuántos peces sobrevivirán de diferentes muestras a evaluar y así sopesar que distribución conviene más para cada acuario.

Concretamente, la nueva especie introducida se sabe que come una vez al día y que lo hace alimentándose con peces de su misma especie de menor tamaño que él. El tamaño del pez influye en su velocidad, por lo que se sabe que, cada día, acaban comiendo antes los peces más pequeños y que ningún pez atacará a otro que ese día haya podido comer, puesto que éstos se esconden en la roca para devorar tranquilamente a su presa. Además, se ha comprobado que el pez incrementa su tamaño (siempre en cm) en la misma cantidad que el pez que acaba de comerse.

Por ejemplo, si soltamos 4 peces de tamaños 2, 3, 5 y 6 cms, el primer día el pez de tamaño 3 se comerá al de tamaño 2 y el de tamaño 6 comerá al de tamaño 5. El segundo día, el pez más grande (ahora de 11 cm) se acabará comiendo al más pequeño (de 5 cm), sobreviviendo éste con un tamaño total de 16 cm. Hay que tener en cuenta que ningún pez se comerá a otro que sea mayor que 10 cms.

### Entrada

Como entrada se indicará primero el número de muestras a evaluar. Para cada muestra se indica el número de peces de dicha muestra y a continuación, en la misma línea, el tamaño (en cm) de cada uno de ellos. La lista de peces y tamaños no necesariamente deberá estar ordenada.

### Salida

Como salida, y teniendo en cuenta el comportamiento de esta especie, se debe indicar, para cada muestra, un número por cada vez que haya sobrevivido y que a su vez indique su tamaño (en cm). La salida se devolverá siempre ordenada de menor a mayor tamaño

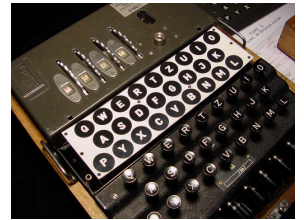
### Entrada de ejemplo

```
3
8
2 2 6 5 8 8 9 8
10
4 6 1 2 12 14 9 8 7 5
4
2 3 5 6
```

### Salida de ejemplo

```
23 25
12 12 13 14 17
16
```

## C. La máquina ENIGMA II



La máquina enigma era el instrumento con el que el ejército alemán cifraba sus mensajes durante la Segunda Guerra Mundial. Durante un tiempo sus mensajes fueron ilegibles pero, finalmente, se logró descifrar su mecanismo, lo que contribuyó a acelerar el fin de la guerra.

Ahora, pasado un tiempo, se quiere rescatar la idea para fabricar una segunda versión de esa máquina con el fin de utilizarla en conflictos armados donde los ordenadores no sean una solución. Así, será posible enviar mensajes de texto claro de forma que sólo el destinatario pueda leerlos.

La nueva máquina cifrará los mensajes utilizando una clave (compuesta de dos caracteres) que irá cambiando a medida que se cifra el mensaje, haciendo así más robusto el algoritmo. La clave se procesa sumando los dígitos de sus caracteres ASCII de forma recurrente para conseguir un solo dígito que se aplicará sobre el código ASCII del carácter que toque cifrar para desplazar el carácter tantas letras (según el abecedario) como indique. Por ejemplo, si la clave es 'ab' (en ASCII 9798), el número que se obtiene sería 6 ( $9 + 7 + 9 + 8 = 33 = 3 + 3 = 6$ ). Si toca cifrar la letra 'h', el resultado sería 'n'.

Hay que tener en cuenta que tras cifrar cada carácter la clave cambia adelantando el segundo carácter en una letra (de 'ab' a 'ac'). Además, después de cada palabra se incrementará también el primero de los caracteres (de 'ab' a 'bc', cambio por letra y palabra). En ambos casos, si la letra es una 'z', volverá a empezar en la 'a'.

### Entrada

Como entrada se recibe una frase para cifrar y la clave inicial en la siguiente línea

### Salida

Como salida se obtiene únicamente el mensaje cifrado

### Entrada de ejemplo

```
hola concursantes de programame zaragoza  
aa
```

### Salida de ejemplo

```
musi dqggzxziwugv ik xbpiergun cdvfmvij
```



## D. Triple Town

El juego del Triple Town es un juego de estrategia muy conocido por su uso en redes sociales y móviles. Consiste en construir un mundo imaginario utilizando un número limitado de elementos, que combinándolos pueden dar lugar a otros y así ir mejorándolo. En la versión que vamos a desarrollar hoy el usuario sólo dispone de 3 elementos: G (grass), B (bush ) y T (tree) que son césped, arbusto y árbol. Cuando se combinan 3 unidades de cualquiera de estos tres elementos se genera un nuevo elemento siguiendo unas reglas:

- 3 unidades de césped (G) generán un arbusto (B)
- 3 unidades de arbusto (B) generar un árbol (T)
- 3 unidades de árbol (T) generan una casa (que será identificada en el mapa con una H)

Hay que tener en cuenta que estas reglas sólo se pueden aplicar a elementos que se encuentran alineados horizontal y verticalmente, nunca en diagonal.

### Entrada

El tablero con el estado inicial del mundo seguido de las jugadas que el usuario realice, sin límite. El usuario deberá indicar, para cada jugada, el elemento que coloca seguido de las coordenadas de la casilla donde lo ubicará en el formato *fila,columna* teniendo en cuenta que, tanto filas como columnas, empiezan a numerarse con el número 1

### Salida

La salida será el estado final del mapa, aplicando las reglas descritas en el juego

### Entrada de ejemplo

```
X X X B X X
X X X X X X
X G X B X X
X G X X X X
X X X G X X
H X X X X X
G, 5, 2
B, 2, 4
```

### Salida de ejemplo

```
X X X X X X
X X X T X X
X X X X X X
X X X X X X
X B X G X X
H X X X X X
```

## E. El voto electrónico



Para la próximas elecciones autonómicas se está pensando en implantar un sistema de voto electrónico para que podamos votar desde casa más cómodamente. Una vez que hayamos votado a través de Internet, será un supercomputador quién contará los votos y marcará qué partido político ha ganado en cada Comunidad Autónoma.

Este supercomputador recibirá los votos asignados a cada partido en cada localidad de toda España y realizará el cómputo para indicar finalmente qué partido ha ganado en cada Comunidad.

### Entrada

Como entrada se envía, por ciudad, la siguiente información: El nombre de la ciudad, su comunidad autónoma y una secuencia de números que indica el número de votos asignados a cada partido, que por seguridad, tendrá asignado un número (empezando en 1) según su posición en dicha secuencia. Esta posición será la misma para todo el listado de ciudades que se introduzca al sistema.

### Salida

Como salida se obtendrá, ordenado alfabéticamente por Comunidad Autónoma, que partido político ha ganado en cada una de las mismas, indicando el número que tenía asignado según la secuencia de número de votos de la entrada

### Entrada de ejemplo

```
Zaragoza, Aragon, 23, 43, 45, 67, 23
Logrono, La Rioja, 22, 33, 41, 0, 1
Pamplona, Navarra, 23, 45, 67, 0, 1
Teruel, Aragon, 12, 23, 45, 67, 10
Castellon, Comunidad Valenciana, 23, 45, 67, 89, 10
Huesca, Aragon, 23, 12, 12, 12, 10
Valencia, Comunidad Valenciana, 23, 12, 45, 67, 10
Alicante, Comunidad Valenciana, 23, 45, 67, 89, 10
```

### Salida de ejemplo

```
Aragon partido 4
Comunidad Valenciana partido 4
La Rioja partido 3
Navarra partido 3
```



## F. El supercomputador

El nuevo supercomputador de Zaragoza, HAL9000, dispone de una gran potencia de cálculo para realizar cualquier tipo de tarea pero requiere de un software especial que sepa decidir en qué momento cuál es la tarea que debe ser procesada.

Dicho software debe decidir, en cada instante de tiempo, de entre un listado de tareas que esperan ser ejecutadas, cuál de ellas será procesada antes en función a unos criterios que serán establecidos por los Ingenieros de Software, entre los que se encuentra vuestro equipo de Programame. Por ello, te encuentras diseñando el simulador que determinará cómo debe comportarse HAL9000.

Para construir el simulador se ha creado un sistema al que se le pasa un listado de tareas con el instante en que llegan al supercomputador, su duración y su prioridad. Hay que tener en cuenta que una tarea nunca podrá ejecutarse hasta que llegue su momento. Y también que, en un instante determinado, puede haber varias tareas encoladas y listas para ser ejecutadas. En ese momento, se ejecutará la tarea de mayor prioridad. El sistema está preparado para no encolar nunca dos tareas de prioridad alta por lo que ese caso no es necesario contemplarlo. En caso de que varias tareas de baja prioridad estén listas, se ejecutará primero la que esté antes en la lista.

El tiempo siempre se indica en la misma unidad, en ms.

### Entrada

Como entrada para la simulación, se introduce un listado con todas la tareas que HAL9000 va a tener que ejecutar. Cada tarea viene identificada por la palabra "tarea" y su número, y a cotinuación cierta información: el primer número indica el instante en que llega al supercomputador, el segundo número indica el tiempo que costará ejecutarla y por último, se indica su prioridad.

### Salida

Como salida se mostrará el orden real de ejecución de las tareas y, finalmente, el tiempo total que ha estado el ordenador en funcionamiento (sin procesar o procesando tareas) seguido del tiempo total de proceso, que es el tiempo duarante el cual el supercomputador ha estado procesando tareas

### Entrada de ejemplo

```
tarea1 1 3 baja
tarea2 1 3 baja
tarea3 1 5 alta
tarea4 8 4 baja
tarea5 9 5 alta
```

### Salida de ejemplo

```
Ejecutando tarea3
Ejecutando tarea1
Ejecutando tarea5
Ejecutando tarea2
Ejecutando tarea4
Tiempo de funcionamiento 21 ms
Tiempo total de proceso 20 ms
```





## G. La Primitiva 2.0

El nuevo sistema para comprobar boletos premiados de la primitiva permite, enviado un email, comprobar un número ilimitado de boletos de forma automática. Sólo hay que enviar un email con la información de los boletos que tengas y la combinación de números del sorteo de un día determinado. El sistema, automáticamente, procesa tus los boletos e informa del premio que corresponde a cada uno, siempre en función del gasto que hayas realizado. Para calcular los premios hay que tener en cuenta las siguiente reglas:

- Sólo se reparte premio en los boleos que aciertan 3 ó más números
- En el caso de que se acierten 3 números se reparten 100 euros (el coste del boleto) + 10 euros por número premiado
- Si se aciertan 4 números se reparten 100 euros + 20 euros por cada número premiado
- En el caso de acertar los 5 números se reparte el triple del coste del boleto

### Entrada

El usuario debe facilitar el número de boletos con los que participa, seguido de las combinaciones de números de cada uno de ellos (uno por línea) y finalizar con la combinación de números para cuyo sorteo quiere comprobar sus ganancias

### Salida

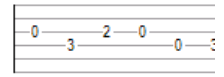
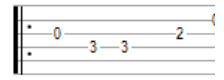
Como salida se debe generar, la cantidad en euros con los que está premiado cada uno de los boletos que se han pasado como entrada, en el mismo orden

### Entrada de ejemplo

```
4
78 5 4 23 55
45 33 32 22 56
5 78 14 9 15
14 5 4 78 15
5 4 14 15 78
```

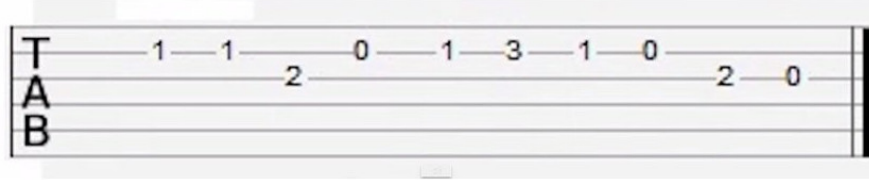
### Salida de ejemplo

```
130
0
180
300
```



## H. Detector de plagios

La tablatura se utiliza como forma de notación para varios instrumentos musicales, que indican como tocar una determinada pieza musical. En el caso de las guitarras, en las tablaturas se indican la cuerda y el traste sobre el que se debe actuar para emitir una nota determinada



En el caso de la imagen, cada línea representa una cuerda de la guitarra, y los números los trastes, de forma que siguiendo la secuencia se puede tocar una canción determinada.

Así, una forma eficaz de detectar plagios consiste en observar dos tablaturas y buscar similitudes entre ambas durante varias notas seguidas. En el caso que nos ocupa consideraremos que una canción es plagio de otra cuando se encuentren, seguidas, tres o más notas "parecidas", teniendo en cuenta que una nota se considera "parecida" a otra cuando, utilizando el mismo traste, se utiliza la misma cuerda o cualquiera de las de al lado (hacia arriba o hacia abajo)

Las cuerdas de la guitarra se numeran de 1 a 6, empezando por arriba

### Entrada

Como entrada se envía la canción original junto con su posible plagio. Se pueden enviar tantas comprobaciones origen-plagio como se deseen siempre y cuando se separen por el número 0. Sólo se comprobará el plagio entre las dos canciones que se envían como pareja. Hay que tener en cuenta que cada pareja de números indica una nota musical compuesta por su cuerda y el traste (cuerda-traste)

### Salida

Como salida se indicará qué canción es un plagio (indicando el número) y entre qué notas se ha detectado. Para cada canción se detectaran tantos plagios como se encuentren. También hay que tener en cuenta que puede no haber plagio entre dos canciones dadas

### Entrada de ejemplo

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

### Salida de ejemplo

```
cancion 2 plagio de la nota 4 a 7
cancion 4 plagio de la nota 1 a 3
```