

Problema 1: Palabras secretas

Se solicita implementar un programa que codifique y decodifique palabras. El programa debe aceptar los siguientes parámetros:

- modo: puede tener dos posibles valores “c” para codificar o “d” para decodificar
- entrada: palabra a procesar (codificar o decodificar).
- semilla: cualquier número entero mayor que 0

El proceso de codificación/decodificación funciona de la siguiente manera:

1. A cada letra del alfabeto (26 letras) se le asigna un número de manera consecutiva (A → 1, B → 2, ... Z → 26). NOTA: el alfabeto completo es “A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z”
2. La palabra de entrada, se codifica o decodifica carácter a carácter.
3. Para codificar una letra, se avanza tantas posiciones como indique el número del parámetro de entrada semilla, obteniendo la nueva letra. Por ejemplo, si la semilla es 3 y tenemos que codificar la letra Q, el resultado es T. Si se llega a la letra final, se empieza de nuevo por el principio. Si la semilla es 3 y tenemos que codificar la letra Y, el resultado es B.
4. Para decodificar el proceso es a la inversa (es decir, se retroceden posiciones).
5. Se procesarán todos los caracteres de entrada pero sólo se codificarán o decodificarán letras del alfabeto. Cualquier otro carácter se copiará a la salida estándar directamente.
6. Ejemplos:
 - a. Al codificar la entrada HOLA!!! con una semilla de 5, el resultado es MTQF!!!
 - b. Al decodificar la entrada FINTX con una semilla de 5 el resultado es ADIOS

Problema 2: Dibujar rectángulo

Debes implementar un programa que visualice por pantalla un rectángulo con caracteres de la siguiente manera.

En primer lugar el programa debe preguntar al usuario el carácter con el que desea dibujar el rectángulo. A continuación, debe preguntar dos números enteros, el primero indicará el número de filas, y el segundo, el número de columnas del rectángulo. Y, por último, debe preguntar si se desea dibujar todo el rectángulo o sólo el contorno.

A continuación el programa mostrará un rectángulo como se muestra en los siguientes ejemplos:

```
Introduce un carácter: *  
Número de filas: 4  
Número de columnas: 3  
Contorno o Relleno (C o R): R
```

```
***  
***  
***  
***
```

```
Introduce un carácter: +  
Número de filas: 5  
Número de columnas: 6  
Contorno o Relleno (C o R): C
```

```
++++++  
+      +  
+      +  
+      +  
++++++
```

Problema 3: Recuento de borda

El recuento de Borda es un proceso electoral en el cual los candidatos son ordenados en la papeleta de voto según las preferencias de cada elector; en el recuento, a cada posición de la ordenación le es atribuida una puntuación igual al número de candidatos que quedan por debajo. Si 'n' es el número total de candidatos, el candidato en la posición i-ésima de una papeleta recibe n-i puntos de esa papeleta. Así, si hay cinco candidatos, al candidato en segunda posición en una papeleta de voto se le asignan tres puntos y al último, cero puntos.

Ejemplo de papeleta para cinco candidatos:

Posición	Candidato	Puntuación obtenida
1º	Pablo Rivera	4
2º	Albert Sánchez	3
3º	Pedro Iglesias	2
4º	Mariano Garzón	1
5º	Alberto Rajoy	0

Las elecciones se pueden realizar para un solo puesto o varios puestos. Asumiremos que el número de puestos 'p' a ocupar es siempre menor que el número de candidatos 'n' a elegir. Cuando se han contabilizado todos los votos, y sumado todos los puntos obtenidos por cada candidato, los 'p' candidatos con mayor puntuación son asignados a los puestos.

Se pide implementar un programa que realice el recuento de Borda.

Datos de entrada

Los datos de entrada, que se leerán desde la entrada estándar, tienen este formato:

<identificador de recuento>

<p: numero de puestos>

<n: número de candidatos>

<candidato 1>

...

<candidato n>

<papeleta 1>

<papeleta 2>

...

<EOF>

El identificador de recuento es una palabra que indica el método de recuento de votos. El número de puestos es $p \leq 10$ y el número de candidatos es $n \leq 20$ y siempre se cumplirá que $p < n$. Todas las líneas van seguidas de un salto de línea, incluida la última. <EOF> indica el final de la entrada.

Puede haber hasta 1000 papeletas. Cada papeleta es una línea que contiene los números de 1 a n, sin repetir, separados por espacios, que se refieren a los n candidatos. He aquí un ejemplo para cuatro candidatos:

```
Borda
1
4
Pablo Rivera
Albert Sanchez
Pedro Iglesias
Mariano Garzon
1 4 2 3
2 3 1 4
3 2 1 4
4 2 3 1
4 3 2 1
1 4 2 3
```

No debes tener en cuenta problemas de formato en la entrada, de momento, como que los números en las papeletas no vayan de 1 a n, no haya n números, etc...

Salida

La salida será el nombre de los 'p' candidatos ganadores. En caso de empates y siempre que haya puestos disponibles, se emitirán los nombres de los ganadores en el mismo orden en el que aparecen en la lista de candidatos de entrada. En caso de que persistan los empates para el último puesto disponible, se asignará el puesto al candidato que aparezca en primer lugar en la lista de candidatos. Para el ejemplo de entrada anterior, la salida del programa debe ser:

```
Albert Sanchez
```

con un salto de línea al final de cada línea. Si se hubieran elegido candidatos para 2 puestos, la salida sería:

```
Albert Sanchez
Mariano Garzon
```

Y para tres puestos sería:

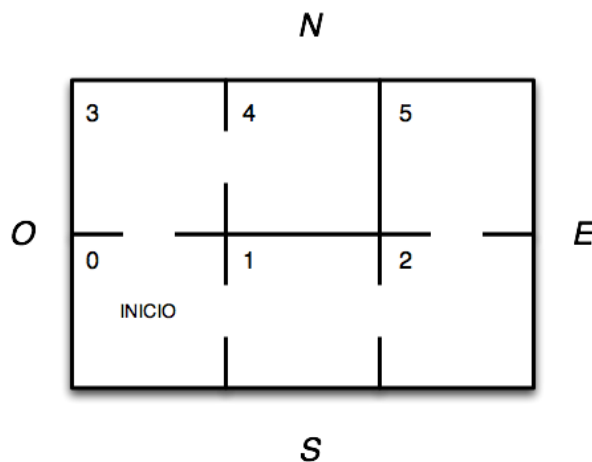
```
Albert Sanchez
Mariano Garzon
Pablo Rivera
```

En caso de que no haya papeletas, el programa debe imprimir
VOTACION ANULADA

(también con salto de línea al final)

Problema 4: La búsqueda del tesoro

Debes programar una aventura en la que el usuario se mueve por un mapa escribiendo la dirección del movimiento deseado (N: norte, E: este, S: sur, O: oeste). La posición inicial del



usuario es la habitación 0. El mapa debe ser el que se muestra en la figura.

Al comienzo del juego se elige aleatoriamente la posición de un vampiro y la de un tesoro. Deben ser números distintos entre 1 y 5 (no pueden estar en la posición inicial del usuario, ni ambos en la misma habitación). El juego termina cuando el usuario llega a la habitación en la que se encuentra el vampiro o el tesoro. Debe aparecer el mensaje indicando si el usuario ha sido mordido por el vampiro o ha encontrado el tesoro.

A continuación puedes ver un ejemplo de ejecución del programa:

```
Estás en la habitación: 0
¿A dónde quieres ir? ["N", "E"] > E
Has escogido la dirección: Este
Y te mueves a la habitación: 1

Estás en la habitación: 1
¿A dónde quieres ir? ["E", "O"] > Blablabla
¿A dónde quieres ir? ["E", "O"] > S
¿A dónde quieres ir? ["E", "O"] > O
```

Has escogido la dirección: Oeste
Y te mueves a la habitación: 0

Estás en la habitación: 0
¿A dónde quieres ir? ["N", "E"] > N
Has escogido la dirección: Norte
Y te mueves a la habitación: 3

Estás en la habitación: 3
¿A dónde quieres ir? ["S", "E"] > E
Has escogido la dirección: Este
Y te mueves a la habitación: 4
Has encontrado el tesoro!!

Observaciones:

- Se disponen de 3 horas para realizar los ejercicios.
- Los resultados se harán públicos en la página web de la Escuela Politécnica Superior de la Universidad de Alicante antes del día 17 de marzo.