

Ejercicio 1

[3 puntos]

Sabiendo que un número es divisible por 11 si el valor absoluto de la suma de las cifras en posición par menos la suma de las cifras en posición impar es cero o múltiplo de 11, y que un número es divisible por 5 si su dígito menos significativo es 0 o 5, es posible calcular de un modo simple, y sin utilizar el operador MOD, si un número es divisible por 11 y por 5, incluso para números naturales con un número de cifras tan grande como se quiera.

Se pide: Escribir un programa en Pascal que lea de la entrada estándar una secuencia no vacía de dígitos, terminada con "#", que representa un número natural y, haciendo uso de las reglas anteriores, muestre por la salida estándar si es múltiplo de 5 y/o de 11. Debe tenerse en cuenta que el número puede tener tantas cifras como se quiera y, por tanto, no es representable en memoria.

Ejemplo de ejecución:

escriba un número natural: 292307565#
es múltiplo de: 5 11

Ejercicio 2

[3,5 puntos]

El alfabeto Morse fue desarrollado por Alfred Vail y Samuel Morse en 1830 en la invención del telégrafo eléctrico. En esencia es un sistema de representación de letras y números mediante códigos consistentes en rayas y puntos; es decir, señales telegráficas intermitentes que se diferencian en el tiempo de duración de la señal (larga y corta).

Para codificar mensajes en Morse se dispone del fichero de registros 'codMorse.dat' en el que cada registro contiene la codificación Morse de un carácter utilizando dos campos: el campo `ch` con el carácter y el campo `cod` con el código Morse correspondiente.

Se pide:

- 1) Definir las estructuras de datos necesarias para resolver el problema.
- 2) Desarrollar el procedimiento `cargarCodigo` que lea del fichero 'codMorse.dat' los códigos Morse correspondientes a cada letra y dígito decimal, y los guarde en la estructura de datos en memoria suministrada como parámetro.
- 3) Desarrollar un programa que lea de la entrada estándar una línea con el mensaje a codificar y muestre por la salida estándar el código Morse correspondiente a dicho mensaje (véase el ejemplo a continuación).

Notas: Para facilitar la posterior lectura del mensaje codificado, se añadirá "/" tras el código de cada carácter, y "/" en los finales de palabra. Se supondrá que no hay errores en los datos y que se utilizan exclusivamente mayúsculas y dígitos.

Ejemplo:

mensaje	mensaje codificado en Morse
HOLA MUNDO / --- / / . - / - - / . - / - . / - . . / - - / /



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

Fig. 1 Telégrafo comercial y tabla con los códigos Morse correspondientes a las letras y dígitos decimales.

Ejercicio 3

[3,5 puntos]

Una forma común de representar objetos sólidos tridimensionales en un computador es mediante el uso de vóxeles. En esta representación, un vóxel es la unidad cúbica elemental que compone el objeto tridimensional, y que puede estar ocupado o vacío. De este modo, un volumen tridimensional de vóxeles modela un objeto sólido, representado por los vóxeles ocupados de dicho volumen.

Para capturar un objeto físico y representarlo mediante vóxeles se utiliza un *escáner láser*, que genera un fichero de texto con las coordenadas de los vóxeles ocupados del objeto. Se supondrá que el fichero contiene al menos un vóxel válido.

Del mismo modo que se accede a un píxel de una imagen a través de dos números enteros (sus coordenadas) a un vóxel dentro de un volumen se accede mediante tres números enteros (coordenadas en el espacio). Podremos suponer que el tamaño máximo admitido de un volumen es de 100x100x100 vóxeles.

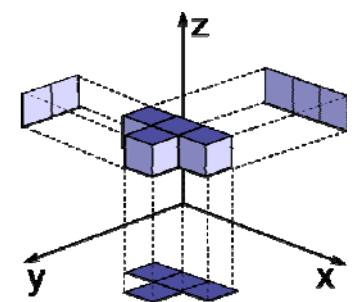
Debido a la naturaleza del proceso de escaneado mediante láser, puede haber:

- **Vóxeles repetidos:** que ya han aparecido antes y, por tanto, no aportan información (ya estaban marcados como ocupados).
- **Vóxeles espurios:** son aquellos que tienen alguna de sus coordenadas fuera del rango permitido (de 1 a 100 en nuestro caso).

Tanto los espurios como los repetidos se consideran datos erróneos y serán ignorados.

```
3 5 9
4 5 9
5 5 9
4 6 9
3 5 105
4 5 9
```

Ejemplo de fichero generado por el escáner.



Representación isométrica del objeto definido por los vóxeles del fichero

La parte izquierda de la figura muestra el contenido de un fichero, con un total de 4 vóxeles válidos, uno espurio (3 5 105) y otro repetido (4 5 9).

Se pide:

- Definir la estructura de datos `tpObjeto` que permita representar todos los vóxeles de un objeto de hasta el tamaño máximo admitido, junto con los rangos de valores de las coordenadas que delimitan el objeto en cada de una de las dimensiones: `minX`, `maxX`, `minY`, `maxY`, `minZ`, `maxZ`.

- Desarrollar el siguiente procedimiento:

```
procedure cargarObjeto(??? fEsc: text; ??? obj: tpObjeto)
```

{Pre: “fEsc” es un fichero de texto que contiene los datos de un objeto, de acuerdo al formato descrito

Post: Almacena en “obj” un volumen con los vóxeles descritos en “fEsc” marcados como ocupados y el resto como vacíos, así como los rangos de valores de las coordenadas que delimitan el objeto en cada una de las dimensiones}

- Desarrollar la siguiente función:

```
function porcentajeOcupacion(??? obj: tpObjeto): real;
```

{Pre: “obj” contiene un objeto

Post: devuelve el porcentaje de vóxeles ocupados en “obj” sobre el total de vóxeles del prisma definido por los rangos de coordenadas, asumiendo que el prisma es rectangular de caras paralelas a los planos XY, YZ, XZ}

La parte derecha de la figura muestra el volumen del objeto con rangos 3-5, 5-6 y 9-9 en X, Y y Z, respectivamente. La ocupación es del 66.66%, pues ocupa 4 vóxeles del total de 6 del prisma.