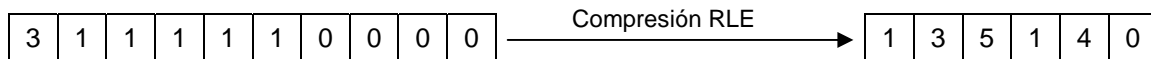


Ejercicio 1

[3 puntos]

La **compresión RLE** o *Run-Length Encoding* es una técnica simple de compresión de ficheros que consiste en sustituir cada secuencia de datos formada por un mismo valor repetido, por el número de veces se repite el valor y dicho valor. Como es lógico, esta técnica sólo interesa si hay muchos valores repetidos consecutivos. A continuación se muestra un ejemplo de compresión de una secuencia de enteros usando esta técnica:



El resultado de la compresión significa que “el fichero original contenía 1 tres, 5 unos y 4 ceros”. La descompresión consiste en el proceso inverso; es decir, en obtener la secuencia original a partir de los datos comprimidos.

Suponiendo que nuestro programa hace uso del tipo de datos `tpFichero` que se corresponde con ficheros secuenciales de números enteros, **se pide:**

- Desarrollar el procedimiento:

```
procedure comprimir(??? fOriginal: tpFichero; ??? fComprimido: tpFichero)
{ comprime, usando RLE, la secuencia de enteros contenida en fOriginal y la guarda en fComprimido }
```
- Desarrollar el procedimiento:

```
procedure descomprimir(??? fComprimido: tpFichero; ??? fOriginal: tpFichero)
{ devuelve en fOriginal la secuencia de datos que hay comprimida mediante RLE en fComprimido }
```

Nota: Se desconoce la longitud de la secuencia, y ésta puede ser muy larga.

Ejercicio 2

[3,5 puntos]

Una productora de cine y televisión necesita de una aplicación informática para filtrar los guiones de sus programas y largometrajes a público infantil, censurando todas aquellas palabras malsonantes o inadecuadas para dicho público.

Se pide:

- Definir el tipo `tpPalabra` que permita representar una palabra.
- Desarrollar un programa Pascal que, dado un fichero de texto que contenga las palabras a censurar (una por línea) lea un texto completo de otro fichero y lo guarde en un tercero, sustituyendo cada palabra censurada por “xxx” (véase el ejemplo). Los nombres de los tres ficheros de texto involucrados (el de censura, el de entrada y el de salida) serán ‘censura.txt’, ‘entrada.txt’ y ‘salida.txt’, respectivamente.

Notas:

- Se podrá considerar que el número máximo de palabras censurables que hay en el fichero de censura es de 500, y que el número máximo de caracteres de una palabra es 30.
- Las palabras del fichero de texto de entrada estarán separadas mediante uno o más caracteres no alfabéticos (espacios, signos de puntuación...) o saltos de línea, que deberán preservarse tal cual en el fichero de destino.
- No es obligatorio tener en cuenta caracteres específicos del castellano (como la ñ o las vocales acentuadas).
- Se deberá tener en cuenta que dos palabras son idénticas independientemente de si están en mayúsculas o en minúsculas.
- No se permite utilizar el tipo de dato `string` para representar las palabras.

Ejemplo de ejecución:

Contenido del fichero censura.txt:	Contenido del fichero entrada.txt:	Contenido del fichero salida.txt tras la ejecución:
recorcholis carajo caracoles jolines	Cuando el grajo vuela bajo hace un frio del carajo. Jolines. Caracoles, pues es verdad.	Cuando el grajo vuelva bajo hace un frio del XXX. XXX. XXX, pues es verdad.

(continúa en página siguiente →)

Ejercicio 3

[3,5 puntos]

Un satélite geoestacionario se encarga de la vigilancia de una base ultrasecreta americana en el desierto de Mojave, en el sudoeste de EEUU. Para ello, el satélite toma fotografías del área en la que se encuentra la base cada cinco minutos. Además, también obtiene un mapa de alturas del terreno con la misma frecuencia, de forma simultánea a cada fotografía.

La **fotografía**, que es en escala de grises, se representa en un computador como una matriz de 1024x1024 elementos denominados píxeles. Cada uno de estos píxeles es un número entero entre 0 y 255, con 0 representando negro y 255 blanco, y los valores intermedios distintas gradaciones de gris. El **mapa de alturas** también se representa como una matriz, en este caso de número reales, de 1024x1024 elementos. Los números representan altura en km sobre el nivel del mar. La matriz que constituye la fotografía y la que muestra el mapa de alturas están **perfectamente alineadas**, es decir, el elemento (1,1) de la fotografía y el (1,1) del mapa de alturas corresponden al mismo punto del terreno.

Se pide: Desarrollar un procedimiento en Pascal `encontrarVehiculos` que, a partir de la fotografía y el mapa de alturas correspondientes a un cierto instante, haga lo siguiente:

- Detecte y muestre por pantalla la posición de la base. Para ello, se sabe que el centro de la base está en el punto más elevado del terreno, esto es, el de mayor valor en el mapa de alturas. Si hubiese varios máximos, tomará el primero que encuentre.
- Busque la existencia de posibles vehículos y su posición. Los vehículos se pueden identificar en la fotografía porque son conjuntos de 2x2 píxeles muy oscuros (esto es, con valores de píxel menores que 100). Los píxeles pertenecientes a la base y al desierto tienen valores de píxel superiores a 100. Nota: no puede haber dos vehículos pegados (es decir, en píxeles contiguos de la matriz).
- Para cada vehículo encontrado, calcule su distancia a la base y la muestre por pantalla. La distancia entre la base y un vehículo se calculará en píxeles (ver ejemplo). De cara a calcular la distancia, la posición de la base viene dada por su centro, y la del vehículo por el píxel superior izquierdo de los cuatro que lo componen.
- Se calculará y mostrará por pantalla, además, el número total de vehículos encontrados.

La fotografía será de tipo `tpImagen`, y el mapa de alturas de tipo `tpAlturas`. Ambos tipos se definen a continuación, junto con la cabecera del procedimiento a desarrollar.

```
const  SIZE = 1024;
type   tpImagen  = array[1..SIZE,1..SIZE] of integer;
        tpAlturas = array[1..SIZE,1..SIZE] of real;
procedure encontrarVehiculos(??? alturas: tpAlturas; ??? foto: tpImagen);
```

Un ejemplo (para un tamaño de 5x5) junto con la salida correspondiente es el que sigue:

Entrada:

Base					Vehículo				
1,561	1,559	1,558	1,555	1,553	150	255	255	10	10
1,559	1,559	1,557	1,554	1,553	255	250	255	11	11
1,452	1,452	1,451	1,450	1,449	148	6	9	150	149
1,451	1,450	1,449	1,447	1,448	136	5	7	148	125
1,450	1,449	1,448	1,447	1,446	133	150	149	115	120

← alturas

← foto

Vehículo

Salida:

Encontrada base en (1,1) - altura = 1.561
Encontrado vehículo en (1,4) - distancia a la base = 3.00
Encontrado vehículo en (3,2) - distancia a la base = 2.24
Se han encontrado un total de 2 vehículos.