

Desencriptando Cadenas

NOTA: Si usted está leyendo este documento sin haber extraído el compactado que se le entregó, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios no se guarden. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

Fito ha comenzado a trabajar en la Agencia de Seguridad de Cadenas e Intercepción de Información (ASCII). Su tarea allí es muy importante: es parte del equipo encargado de desencriptar los mensajes del enemigo que han sido interceptados.

La tarea de desencriptar un mensaje es un proceso complejo, por eso el equipo de Fito solicita ayuda a otros departamentos para poder conseguirlo. Cada vez que un mensaje encriptado llega al equipo de Fito, este es enviado a ser procesado por otros departamentos de inteligencia para determinar las características del lenguaje empleado y el método de encriptación usado.

Un **lenguaje** no es más que el conjunto de todas las cadenas que se pueden formar a partir de un alfabeto definido. Un **alfabeto** es un conjunto finito de símbolos o caracteres que son combinados para crear las cadenas o palabras del lenguaje. Un ejemplo de alfabeto puede ser {'A', 'B', 'T', 'E'}. "BATE", "TE", "TTTEB" y "A" son **cadenas** posibles dentro del lenguaje generado por este alfabeto. "BEX" es una cadena que **no pertenece** al lenguaje generado pues contiene el carácter 'X' que no pertenece al alfabeto. Un **mensaje siempre es una cadena no vacía** generada a partir de un alfabeto definido.

Recientemente se están interceptando mensajes empleando un conocido método de encriptación. El método consiste en, dado el alfabeto del lenguaje en que se escribe el mensaje y un entero no negativo k previamente seleccionado, transformar cada carácter en el que se encuentra a k posiciones a la derecha, **considerando el alfabeto como circular**.

Por **ejemplo**, sea el alfabeto {'A', 'B', 'T', 'E'} y $k = 2$ entonces las 'A' se cambian por 'T', las 'B' por 'E', las 'T' por 'A' y finalmente las 'E' por 'B'. Por ende si se tuviera la palabra "BATE" escrita en este lenguaje quedaría encriptada como "ETAB" utilizando este método.

El equipo de Fito se enfrenta ahora a un problema más difícil: tienen la cadena encriptada y conocen el alfabeto del lenguaje al cual pertenece pero desconocen cuál fue el k empleado en el proceso de encriptación.

Utilizando la información ofrecida por los agentes de campo desplegados por todo el mundo, se ha podido conocer con alto nivel de confianza, cuál es el carácter del alfabeto empleado que más se repite en el mensaje original. Con esta nueva información el equipo de Fito ya puede desencriptar el mensaje ya que solo debe conocer en qué carácter se ha transformado el carácter que más se repite para encontrar el k empleado.

Por **ejemplo**, sea el alfabeto {'A', 'B', 'T', 'E'}, el mensaje encriptado "EAEBT" y la información de que el carácter 'A' es el que más se repite, el equipo de Fito puede concluir que el carácter 'A' se ha transformado en 'E' por lo que $k = 3$ y la cadena desencriptada es "ABATE".

Tú como importante integrante del equipo de Fito, tienes la tarea de implementar el algoritmo de descryptación de mensajes teniendo en cuenta todo lo antes planteado.

Usted debe haber recibido junto a este documento una solución de Visual Studio con dos proyectos: una biblioteca de clases (Class Library) y una aplicación de consola (Console Application). Usted debe implementar el método *Descryptar* que se encuentra en la clase *Criptografía* en el *namespace Weboo.Examen*. En la biblioteca de clases encontrará la siguiente definición:

```
namespace Weboo.Examen
{
    public class Criptografía
    {
        public static string Descryptar(char[] alfabeto, char caracter, string mensaje)
        {
            //Borre la siguiente línea y escriba su código
            throw new NotImplementedException();
        }
    }
}
```

Este método recibe como primer parámetro un array de **char** con el alfabeto con el que se escribió el mensaje, como segundo parámetro un **char** que es el caracter que más se repite en el **mensaje original** y como último parámetro un **string** con el mensaje encriptado. Ud. debe implementar este método de tal manera que devuelva un **string** con el mensaje descryptado sabiendo que el método de encriptación usado fue el anteriormente descrito.

Puede asumir que:

- El parámetro alfabeto **nunca** será **null** y será un array **no vacío** de **char** diferentes. El array alfabeto puede contener **cualquier char**.
- El parámetro caracter **siempre** pertenecerá al alfabeto y a la cadena descryptada.
- El parámetro mensaje **nunca** será **null** ni vacío y **siempre** estará compuesto por caracteres del alfabeto.
- El problema **siempre** tiene solución, o sea, **siempre** existirá un **único** $k \geq 0$ que cumpla con todo lo planteado y permita descryptar el mensaje.
- El mensaje descryptado **no** tiene por qué tener sentido en nuestro lenguaje. Es tarea de la Agencia decidir que hacer con el texto descryptado.

NOTA: Todo el código de la solución debe estar en este proyecto (biblioteca de clases), pues es el único código que será evaluado. Usted puede adicionar todo el código que considere necesario, pero no puede cambiar los nombres del namespace, clase o método mostrados. De lo contrario, el probador automático fallará. En particular, es imprescindible que usted no cambie los parámetros del método *Descryptar*, ni su orden. Por supuesto, usted puede (y debe) adicionar todo el código que necesite.

Ejemplos

NOTA: Los casos de prueba que aparecen en este proyecto son solamente de ejemplo. Que usted obtenga resultados correctos con estos casos no es garantía de que su solución sea correcta y de buenos resultados con otros ejemplos. De modo que usted debe probar con todos los casos que considere convenientes para comprobar la validez de su implementación.

```
////////////////////////////////////  
////////////////////////////////////CASO DE PRUEBA 1////////////////////////////////////  
////////////////////////////////////
```

```
char[] alfabeto1 = { 'A', 'B', 'T', 'E' };  
char caracter1 = 'B';  
string msjEncriptado1 = "ATAE";  
  
string msj1 = Criptografía.Desencriptar(alfabeto1, caracter1, msjEncriptado1);  
Console.WriteLine(msj1); //msj1 = "BEBA"
```

```
////////////////////////////////////  
////////////////////////////////////CASO DE PRUEBA 2////////////////////////////////////  
////////////////////////////////////
```

```
char[] alfabeto2 = { 'A', 'B', 'T', 'E' };  
char caracter2 = 'A';  
string msjEncriptado2 = "EAEBT";  
  
string msj2 = Criptografía.Desencriptar(alfabeto2, caracter2, msjEncriptado2);  
Console.WriteLine(msj2); //msj2 = "ABATE"
```

```
////////////////////////////////////  
////////////////////////////////////CASO DE PRUEBA 3////////////////////////////////////  
////////////////////////////////////
```

```
char[] alfabeto3 = { 'S', ' ', 'T', 'E', 'H', 'U', 'R', 'Y', 'O' };  
char caracter3 = 'E';  
string msjEncriptado3 = "EORSURHYT "; //termina con un caracter espacio  
  
string msj3 = Criptografía.Desencriptar(alfabeto3, caracter3, msjEncriptado3);  
Console.WriteLine(msj3); //msj3 = "SUERTE HOY"
```

```
////////////////////////////////////  
////////////////////////////////////CASO DE PRUEBA 4////////////////////////////////////  
////////////////////////////////////
```

```
char[] alfabeto4 = { 'x', '-', 'Q', '1', '*', '0', 'c', 'L', '[' };  
char caracter4 = '1';  
string msjEncriptado4 = ".*[xx1-*cL";  
  
string msj4 = Criptografía.Desencriptar(alfabeto4, caracter4, msjEncriptado4);  
Console.WriteLine(msj4); //msj4 = "1L1L[[Qx10c"
```