

# Práctica 2 de Programación 1

Andrés Farías y Jesús Pérez

Diciembre, 2019

## 1 Introducción

La práctica consiste en programar un traductor de mensajes. Para ello se debe usar un conjunto de funciones y procedimientos que permitan resolver pequeñas partes para llevar a cabo las operaciones necesarias.

El Traductor de Mensajes requiere una cadena inicial que estará expresada en un formato **binario** y será convertida a un formato **alfanumérico**.

Un ejemplo de esto sería:

-Dada la cadena **01001000011011110110110001100001**, donde 01001000 es ‘H’, 01101111 es ‘o’, 01101100 es ‘l’ y 01100001 es ‘a’, la salida traducida deberá ser **“Hola”**.

## 2 Práctica

Para la realización de este trabajo, se deberá hacer uso de lo visto en clases, sin añadir otra biblioteca adicional a “pr1-ula.h” y sin añadir contenido adicional a lo explicado en el curso hasta los momentos, abarcando el manejo de los tipos de datos Entero, Caracter y Cadena, así como también las funciones, procedimientos, las estructuras de repetición y de decisión.

Casos de prueba:

Input: 01001000011011110110110001100001	Output: Hola
Input: 010100000111001000110001010101010110110001100001	Output: Pr1Ula
Input: 0100010101111000011000010110110110010101101110	Output: Examen

Estos casos de prueba le servirán como una ayuda para verificar que el programa funcione bien, pero los evaluados serán distintos a los aquí expresados.

### 3 Cosas a considerar antes de empezar

Es necesario comprender el funcionamiento de dos aspectos fundamentales en la programación:

a) **Representación en código ASCII:** Todos los caracteres del idioma inglés pueden ser representados mediante un valor dentro del código ASCII, es por esto que los caracteres pueden considerarse como números dentro de este código.

Ejemplo:

Character: A	Código en ASCII: 65
Character: W	Código en ASCII: 87
Character: a	Código en ASCII: 97
Character: j	Código en ASCII: 106
Character: l	Código en ASCII: 108

b) **Código binario:** Permite escribir números de base decimal (0,1,2,3,4....) en números de 8 dígitos en base binaria (00000000,00000001,00000010,00000011....), donde cada dígito del número binario es llamado bit.

0 1 0 0 0 0 0 1

c) **Lectura de números binarios:** Los números binarios a diferencia de los números de base decimal deben leerse de derecha a izquierda, es decir, empezando desde el final, donde cada bit posee una representación de una potencia de 2:

$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$

Donde la transformación de este número a binario es la suma de las potencias cuyos bits sean 1, por ejemplo:

$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$   
0 1 0 0 0 0 0 1

Este sería el número  $2^6 + 2^0 = 64 + 1 = 65$

d) **Representación de caracteres en formato binario:** Todos los números de la tabla ASCII normal pueden ser representados mediante un código binario de ocho bits denominado “byte”, es por esto que la tabla ASCII posee su propia interpretación en binario. De esta manera podemos comprender que cada caracter puede ser llevado a su representación en el código ASCII y de allí convertirse en su equivalente binario.

Ejemplo:

Character: A	Código en ASCII: 65	Representación en binario: 01000001
--------------	---------------------	-------------------------------------

## 4 Enunciado

La agencia espacial LaSDAI necesita un programa que le permita enviar instrucciones a sus astronautas a bordo de una nave espacial. Dadas las complicaciones para la transferencia de grandes paquetes de información, la agencia espacial ha determinado que la mejor manera de enviar datos es en forma binaria. A pesar de esto los astronautas requieren poder visualizar el mensaje en un formato de texto alfanumérico.

## 5 Rutinas

### 1. Calentamiento:

a) **Entero potencia(Entero,Entero):** Recibe un entero base y un entero exponente, cumple con la definición matemática de la potenciación, deberá retornar el valor de elevar la base a la potencia.

b) **Caracter decimalAASCII(Entero):** Recibe un dato de tipo entero y deberá retornar el caracter representado en la tabla ASCII . Esta rutina se deberá realizar en solo dos instrucciones de código.

c) **void imprimirCadena(Cadena):** Recibe una cadena y deberá imprimir su contenido por el terminal.

### 2. Subprogramas:

a) **Entero binarioADecimal(Cadena):** Recibe como parámetro una cadena, la cual contendrá un byte, es decir 8 bits, los cuales deberán ser convertidos y retornados en su interpretación formato decimal. Para la realización de esta rutina se recomienda utilizar la función potencia(Entero,Entero) previamente descrita.

b) **Cadena agregarCaracter(Cadena,Caracter):** Recibe como parámetros una cadena y un caracter; su objetivo es retornar la cadena recibida a la cual se le habrá adjuntado al final el caracter recibido. Se debe tener en cuenta que, al agregar un caracter a una cadena se debe actualizar su tamaño.

c) **Cadena extraerSubcadena(Cadena,Entero,Entero):** Recibe una cadena y dos datos de tipo entero, los cuales expresan el punto inicial y final desde donde se extraerá la subcadena de la original (estos puntos inclusive). Se debe tener en cuenta que la cadena a retornar tenga actualizado su tamaño.

La implementación del procedimiento **void traducirMensaje(Cadena)** es facilitada en el archivo *plantilla.c*, es este el procedimiento que hace el llamado a los demás procedimientos y funciones desarrolladas. Recibe una cadena cuya traducción será mostrada por terminal, **NO DEBE SER MODIFICADA**. Será esta la que permita corroborar el buen funcionamiento de las demás implementaciones. Este procedimiento debe ser agregado dentro del archivo *cadenas.c*.

## 6 Evaluación

La fecha de entrega de esta práctica es desde el 06/12/2019 hasta el 13/12/2019.

Para la evaluación se debe enviar el archivo *cadenas.c* a la dirección:

**jaf4rdev@gmail.com**

El asunto del correo debe ser exactamente el texto “**PR1-LAB-02**” sin las comillas. Si falla con el asunto del correo entonces la práctica no será evaluada, ya que el mecanismo automatizado de filtrado no la podrá detectar. NO comprima el archivo y no envíe nada adicional a éste. El único contenido que debe aparecer en el correo es el archivo *cadenas.c* en archivo adjunto, y en el mensaje, su sección, el primer apellido, el primer nombre, como se muestra en el siguiente ejemplo:

**S1,Perez,Jesus**

Durante el periodo de entrega tiene dos intentos para la corrección de la práctica, de modo que, si en el primer intento no obtiene la calificación máxima, se le responderá el correo con los fallos, y el segundo intento será el que tendrá la calificación final de la práctica. La fecha máxima para el envío del primer intento será el domingo 08/12/2019, cualquier envío realizado luego de las 23:59:59 de tal día será considerada la entrega final.

Dentro del archivo *cadenas.c* debe ir en comentario su número de cédula, nombre y apellido.

Por favor, no use otros medios para enviar la solución ni escriba otros comentarios adicionales en el email. **Atención:** si su programa no compila, entonces el evaluador no compila. Si una de las rutinas cae en un bucle infinito entonces el evaluador cae en un bucle infinito. Por esa razón, en todos estos casos será imposible darle una nota, lo que simplemente se traduce en que usted tiene cero en el intento en el cual ocurra una de las circunstancias mencionadas. **No envíe programas que no compilan.**

## 7 Recomendaciones

1. Lea enteramente este enunciado antes de proceder al diseño e implementación. Asegúrese de comprender bien su diseño. Haga un boceto de su programa y como es esperado que funcione. Por cada rutina, plantee qué es lo que va a hacer, cómo va a resolver el problema. Esta es una situación que amerita de diseño y desarrollo.

2. Use el grupo de Telegram para plantear las dudas de comprensión. Pero de ninguna manera comparta código, pues es considerado **plagio**.

3. No se debe modificar la plantilla que se envía, los prototipos facilitados deben estar

tal cual acá planteados, tampoco debe agregar parámetros adicionales, no debe crear más funciones o procedimientos que los acá descritos, al añadir nuevas variables debe asignarles nombres que ayuden a comprender el funcionamiento del código.

4. Escribe tu código como si fuese un poema para esa persona que tanto te atrae, de fácil comprensión, con una elegancia y estilo apropiado.