

Actividad 1.1

Sistemas de Numeración

Ricardo David López Arellano
Departamento de Ingeniería en Computación
CUCEI
Universidad de Guadalajara
ricardo.lopez1361@alumnos.udg.mx

Abstract

El diccionario es un espacio donde es posible crear nuevas palabras, de tal manera que se puede expandir el lenguaje (metalenguaje).

I. ORIGINALIDAD

Me comprometo a producir trabajo académico íntegro, lo que significa un trabajo que se adhiere a los estándares intelectuales y académicos de atribución exacta de las fuentes, uso y recolección de datos apropiados, y transparencia en el reconocimiento de las contribuciones de las ideas, descubrimientos, interpretaciones y conclusiones de otros. Acepto que la trampa en los exámenes, el plagio o la fraudulenta representación de las ideas o lenguaje de otros como propio, la falsificación de datos o cualquier otra instancia de deshonestidad académica, violan los estándares de LA MATERIA, así como los estándares del mundo en general en el campo del conocimiento y las relaciones.

II. INTRODUCCIÓN

Un sistema de numeración es un conjunto de símbolos y reglas de generación que permiten construir todos los números válidos.

III. OBJETIVOS DE LA ACTIVIDAD

- Implementar dos funciones para la conversión de los sistemas de numeración.

IV. METODOLOGÍA

- Diseña e implementa dos funciones para la conversión entre sistemas de numeración en lenguaje C, donde la declaración de dichas funciones es la siguiente:

```
uint8_t base_o(uint8_t * binary_number, uint8_t k, const char * base_string, unsigned int binary_length)
void to_base(char * base_string, const uint8_t * binary_number, uint8_t k, unsigned int binary_length)
```

Siendo $base_o$ una función que permite convertir una cadena de texto a $base_string$ que contiene los dígitos de la base $B = 2^k$ (k es el número de dígitos).

V. CONTENIDO

A. Código:

```
include<stdlib.h> include<conio.h> include<stdio.h> include<stdint.h>
int num; int opc;
void base_o(uint8_t * binary_number, uint8_t k, const char * base_string, unsigned int binary_length) void binario() int aux; if (num == 0)
aux = num; num = num / 2; binario();
printf(" void octal() int aux1; if (num == 0) return;
aux1 = num; num = num / 8; octal();
printf("
void hex() printf(""); printf("
int main() do printf(" ¶ PRACTICA 1 ¶"); printf("de numeros a distintas opciones:"); printf("Introduce un numero: ");
scanf("
printf("1- Binario 2- Octal 3- Hexadecimal 4- Salir Ingresa una opcion: "); scanf("switch (opc) case 1: printf("valor en
binario es: "); binario(); printf(""); system("pause"); system("cls"); break;
case 2: printf("valor en octal es: "); octal(); printf(""); system("pause"); system("cls"); break;
case 3: printf("valor en hexadecimal es: "); hex(); printf(""); system("pause"); system("cls"); break;
case 4: exit(4);
default: system("cls"); printf("opcion no esta disponible..."); system("pause"); system("cls"); while (opc = 5 );
```

VI. RESULTADOS

1) EJERCICIO 1:

Respuesta: _____ MARCO _____

```
7 //ALTO
80 //ANCHO
2 - //restamos 2 caracteres a la pila (las orillas superiores)
    swap
2 - //restamos 2 caracteres a la pila (las orillas inferiores)
    swap
"+ " S. //Ingresamos la orilla superior izquierda
    dup
    2 /
    8 -
    dup
while dup 0 { //ciclo para hacer linea superior
    "- " S.
    1 -
    drop
"} S. //simbolo antes de finalizar la palabra 'principal' " MenuPrincipal " S. //palabras puestas de 'titulo'
"+ " S. //simbolo antes de comenzar la palabra 'menu'
    dup
while dup 0 { //ciclo para hacer linea inferior
    "- " S.
    1 -
    drop
    swap
    "+ " S. //Ingresamos la orilla superior derecha
        cr
        swap
        drop
        swap
        dup
        swap
        rot
        swap
while dup 0 { //ciclo para hacer lineas de los costados
    "} " S. //primer linea izquierda del primer renglón
        swap
        dup
        while dup 0 {
            " " S. //Imprimir espacios
                1 -
                drop
                swap
                1 -
            "} " S. //Linea final del primer renglón
                cr
                drop
            "+ " S. //Ingresamos la orilla inferior izquierda
                dup
                while dup 0 { //ciclo de la linea final
                    "- " S. 1 -
                    swap
                    "+ " S. //Ingresamos la orilla inferior derecha
```

```

swap
drop
cr
drop
drop
//———— IMPRIMIR MENU —————
//iniciamos en 3 en la pila y de ahí imprimimos todas las opciones, en cada uno de los siguientes if se hará lo mismo
mientras vas bajando o subiendo en el menú se seguirá imprimiendo el menú. (termina la programación de menú hasta
que sale el siguiente comando //—— :*)
3
repeat
if dup 3 =
3
2
OS:gotoxy //ubicamos en la posición dada anteriormente
"ESC[7m" S.
"1. Imprime del 0-9" S.
"ESC[0m" S.
3
3
OS:gotoxy
"2. Imprime del -9-0" S.
3
4
OS:gotoxy
"3. Imprime de la a-z" S.
3
5
OS:gotoxy
"4. Imprime de la A-Z" S.
3
6
OS:gotoxy
"5. Salir" S.
//pos 4 en la pila
elif dup 4 =
3
2
OS:gotoxy
"1. Imprime del 0-9" S.
3
3
OS:gotoxy
"ESC[7m" S.
"2. Imprime del -9-0" S.
"ESC[0m" S.
3
4
OS:gotoxy
"3. Imprime de la a-z" S.
3
5
OS:gotoxy
"4. Imprime de la A-Z" S.
3
6

```

```

OS:gotoxy
"5. Salir" S.
//pos 5 en la pila
elif dup 5 =
3
2
OS:gotoxy
"1. Imprime del 0-9" S.
3
3
OS:gotoxy
"2. Imprime del -9-0" S.
3
4
OS:gotoxy
"ESC[7m" S.
"3. Imprime de la a-z" S.
"ESC[0m" S.
3
5
OS:gotoxy
"4. Imprime de la A-Z" S.
3
6
OS:gotoxy
"5. Salir" S.
//pos 6 en la pila
elif dup 6 =
3
2
OS:gotoxy
"1. Imprime del 0-9" S.
3
3
OS:gotoxy
"2. Imprime del -9-0" S.
3
4
OS:gotoxy
"3. Imprime de la a-z" S.
3
5
OS:gotoxy
"ESC[7m" S.
"4. Imprime de la A-Z" S.
"ESC[0m" S.
3
6
OS:gotoxy
"5. Salir" S.
//7 en la pila (ESC)
elif dup 7 =
3
2
OS:gotoxy
"1. Imprime del 0-9" S.

```

```

3
3
OS:gotoxy
"2. Imprime del -9-0" S.
3
4
OS:gotoxy
"3. Imprime de la a-z" S.
3
5
OS:gotoxy
"4. Imprime de la A-Z" S.
3
6
OS:gotoxy
"ESC[7m" S.
"5. Salir" S.
"ESC[0m" S.
//—— : *
//——RESTRICCIONES ——
elif dup 1 =
7 +
elif dup 2 =
6 +
elif dup 8 =
6 -
elif dup 9 =
7 -
//————— TECLAS —————
//en los siguientes if moveremos la posicion dependiendo la flecha que eligas...
OS:key
if dup 1792834 =
swap
1 +
swap
drop
elif dup 1792833 =
swap
1 -
swap
drop
elif dup 1792835 =
swap
2 +
swap
drop
elif dup 1792836 =
swap
2 -
swap
drop
//————— EJERCICIOS —————
//aqui inician los ejercicios que si al dar enter haces la opcion solicitada
elif dup 10 =
drop
cr cr

```

```

//opcion 1
if dup 3 =
  0 .
  0
  dup
  while 9 ;
  1 +
  dup .
  dup
  drop
//opcion 2
elif dup 4 =
  cr
  -9
  while dup
  dup S.
  ++
  0 .
  drop
//opcion 3
elif dup 5 =
  cr cr
for 97 upto 122
  dup emit
//opcion 4
elif dup 6 =
  cr cr cr
  65
  repeat
  dup emit
  ++
  dup
  until 91 =
  end
  drop
//opcion 5
elif dup 7 =
  jump final
until dup 27 = end
final: cr cr cr cr

```

VII. CONCLUSIONES

En conclusión de esta tarea puedo decir que estos ejercicios ya fueron complicados ya que su complejidad ya era mucho mas alta, se puede notar viendo el tamaño de código que se hizo muy extenso pero con mis conocimientos y pidiendo algo de ayuda a los compañeros pude realizarlos correctamente. Es bueno aprender un lenguaje nuevo ya que yo ni si quiera había utilizado Linux ni mucho menos Latex, así que es una buena experiencia.

AGRADECIMIENTOS

Quiero hacer agradecimiento a mi profesor por explicarme cuando tenia dudas sobre cómo hacer los ejercicios, a mis compañeros porque varias veces me brindaron ayuda cuando tenia problemas y a mis padres en apoyarme cuando los necesito.

REFERENCES

- [1] Becerra Alvarez, E. C. (2022, 4 octubre). ForEmb. <https://drive.google.com/file/d/1hmdyOIhfwLkDTMcB8XZV8USGuOheWac2/view>