



**PRACTICA 12: CALCULADORA CONVERTIDORA DE SISTEMAS DE NUMERACIÓN**

**ESTUDIANTE: EFRAIN ROBLES PULIDO**

**CODIGO: 221350095**

**NOMBRE DE LA MATERIA: SEMINARIO DE SOLUCION DE PROBLEMAS DE  
PROGRAMACION**

**SECCIÓN: D67**

## Descripción:

### Objetivo de la practica

Hacer un programa que convierta de un sistema de numeración a otro, de acuerdo con el siguiente menú, después de mostrar el resultado debe regresar al menú, el programa termina hasta que se seleccione la opción salir:

1. Decimal a Binario
2. Binario a Decimal
3. Decimal a Hexadecimal
4. Hexadecimal a Decimal
5. Binario a Hexadecimal (Directamente)
6. Hexadecimal a Binario (Directamente)
7. Salir

### Fundamentación teórica

Un sistema de numeración es un conjunto de símbolos y reglas que permiten representar datos numéricos. Los sistemas de numeración actuales son sistemas posicionales, que se caracterizan porque ***un símbolo tiene distinto valor según la posición que ocupa en la cifra.***

<b><i>Binario</i></b>	0 1 <span>© carlospes.com</span>
<b><i>Octal</i></b>	0 1 2 3 4 5 6 7
<b><i>Decimal</i></b>	0 1 2 3 4 5 6 7 8 9
<b><i>Hexadecimal</i></b>	0 1 2 3 4 5 6 7 8 9 A B C D E F

1. Decimal a Binario

Convertir un número decimal a binario es muy sencillo: basta con realizar divisiones sucesivas entre 2 y escribir los residuos obtenidos en cada división en orden inverso al que han sido obtenidos.

## 2. Binario a Decimal

Basta con numerar los dígitos de derecha a izquierda comenzando desde cero, a cada número se le asigna la correspondiente potencia base 2 y al final se suman las potencias.

## 3. Decimal a Hexadecimal

Se realiza utilizando la técnica habitual de divisiones sucesivas, como el de decimal a binario. En este caso, las divisiones son entre 16.

## 4. Hexadecimal a Decimal

Basta con numerar los dígitos de derecha a izquierda comenzando desde cero, a cada número se le asigna la correspondiente potencia base 16 y al final se suman las potencias.

## 5. Binario a Hexadecimal

Se tiene que numerar los dígitos de derecha a izquierda comenzando desde cero, a cada número se le asigna la correspondiente potencia base 2 y al final se suman las potencias, pero con la diferencia de que se hará solo para un grupo de 4 bits.

## 6. Hexadecimal a Binario

Cada letra o número del hexadecimal representará a un valor binario con un tamaño de 4 en donde solo se pondrán juntos los valores binarios.

### **Análisis del problema**

Se deberá utilizar algunos ciclos for, así como el control selectivo múltiple para hacer la asignación de valores o poder hacer el menú para cumplir con la práctica y diferentes estrategias para hacer la conversión de los sistemas numéricos como las funciones, así como las funciones en donde nos permitirá organizar y tener un mejor control del programa.

### **Datos de entrada y precondiciones**

- Valor para seleccionar el menú
- Valor por convertir

### **Datos o elementos de salida**

- Valor ya convertido

## **Desarrollo:**

### **Procedimiento en lenguaje natural**

Se deberá usar un ciclo while para mantener el menú y el control selectiva múltiple para poder realizar el menú, dentro de ellas se llamarán las funciones en donde se realizará las conversiones:

#### **1. Decimal a Binario**

Una vez introducido el valor decimal se realizará divisiones sucesivas entre 2 y guardar la división de residuo de 2 obtenidos para guardarlo en un arreglo en un ciclo for para mostrarlo con otro ciclo for donde empezara con el mas significativo hasta el menos.

#### **2. Binario a Decimal**

Una vez introducido el valor binario, los dividiremos para guardarlo en un arreglo, su residuo de la división con el 10 en un ciclo for, para después numerar los dígitos de izquierda a derecha comenzando desde el 7, y a cada número se le multiplicara con la correspondiente potencia base 2 y al final se suman las potencias.

#### **3. Decimal a Hexadecimal**

Se realiza utilizara la misma técnica de divisiones sucesivas, como el de decimal a binario. En este caso, las divisiones son entre 16. Donde usaremos un ciclo do-while para hacer las divisiones correspondientes hasta que no se pueda dividir más el valor, también se asignara el último valor obtenido en nuestro arreglo que se guardar la conversión, después usaremos el control selectivo múltiple para mostrar los valores correspondientes del hexadecimal.

#### **4. Hexadecimal a Decimal**

Una vez introducidos los valores como una cadena, se calculará la longitud de la cadena para después usar un ciclo for para ir asignando los valores correspondientes con potencia base 16 y al final se suman las potencias, convirtiendo las letras a números para que funcione correctamente.

#### **5. Binario a Hexadecimal**

Una vez introducido el valor binario de 4 bits en una cadena para después calcular su longitud y después con un ciclo for y el control selectivo simple para hacer las

correspondientes operaciones según con las características del hexadecimal, se multiplicará con su correspondiente potencia base 2 y su posición, y al final se suman los resultados, pero con la diferencia de que se hará solo para un grupo de 4 bits.

## 6. Hexadecimal a Binario

Cada letra o número del hexadecimal se guardar en una cadena, para después calcular su longitud, después con el ciclo for podremos representar cada valor binario con un tamaño de 4 en donde solo se pondrán juntos los valores binarios con el control selectivo multiple.

### Algoritmo

*Pseudocódigo:*

```
entero i, deci, cant
entero dec_bin()
inicio
    entero bin[8]
    imprimir ("Dame numero decimal:")
    leer (deci)
    desde (i←0; i<8; inc i)
    inicio
        bin[i]←deci%2
        deci←deci/2
    fin
    imprimir ("Numero binario es:")
    desde(i←7; i>=0; dec i)
    inicio
        imprimir (bin[i])
    fin
fin
entero bin_dec()
inicio
    entero dig[8], bin
    imprimir ("Dame un numero binario de 8(de izq a der.):")
    leer (bin)
    desde(i←0; i<8; inc i)
    inicio
        dig[i] ←bin%10
        bin←bin/10
    fin
    desde(i←7; i>=0; dec i)
    inicio
```

```

    deci← (pow(2,i)*dig[i])+deci
fin
imprimir ("Numero decimal es:",deci)
fin
entero dec_hex()
inicio
    entero dig[20],resi,resp,j
    imprimir ("Dame un numero decimal:")
    leer (deci)
    hacer
    inicio
        resi←deci%16
        resp←deci/16
        dig[i] ←resi
        deci←resp
        inc i
    fin
    mientras(resp>16)
    dig[i] ←resp
    imprimir ("Numero hexadecimal es:")
    desde(j←i;j>=0;dec j)
    inicio
        segun sea(dig[j])
        inicio
            caso 10:
                imprimir ("A")
                interrumpir
            caso se 11:
                imprimir ("B")
                interrumpir
            caso 12:
                imprimir ("C")
                interrumpir
            caso 13:
                imprimir ("D")
                interrumpir
            caso 14:
                imprimir ("E")
                interrumpir
            caso 15:
                imprimir ("F")
                interrumpir
            otro:
                imprimir (dig[j])

```

```

        fin
    fin
fin
entero hex_dec()
inicio
    caracter hex[100],dig
    imprimir ("Dame un numero hexadecimal:")
    leer (hex)
    cant←strlen(hex)

    desde(i←0;i<cant;inc i)
    inicio
        dig←hex[cant-1-i]
        si(dig>='0' Y dig<='9')
            inicio
                deci+=pow(16,i)*(dig-'0')
            fin
        sino
            deci+=pow(16,i)*(dig-'A'+10)

        imprimir ("Numero decimal es:",deci)
    fin
fin
entero bin_hex()
inicio
    entero hex
    caracter bin[100],dig //NO MAYOR A 4 BITS
    imprimir ("Dame un numero binario de 4(de izq a der.):")
    leer(bin)
    cant←strlen(bin)

    for(i←0;i<cant;inc i)
    inicio
        dig←bin[cant-1-i]
        si(dig>='0' Y dig<='9')
            inicio
                hex+=pow(2,i)*(dig-'0')
            fin
        sino
            hex+=pow(2,i)*(dig-'A'+10)
        fin
    imprimir ("Numero hexadecimal es:")
    según sea(hex)
    inicio

```

```

    caso 10:
        imprimir ("A")
        interrumpir
    case 11:
        imprimir ("B")
        interrumpir
    case 12:
        imprimir ("C")
        interrumpir
    case 13:
        imprimir ("D")
        interrumpir
    case 14:
        imprimir ("E")
        interrumpir
    case 15:
        imprimir ("F")
        interrumpir
    default:
        imprimir ("%d",hex)
    fin

```

fin

```

entero hex_bin(){
    caracter bin[100]
    imprimir ("Dame un numero hexadecimal:")
    leer(bin)
    cant←strlen(bin)
    imprimir("Numero binario es:")
    desde(i←0;i<cant;inc i)
    inicio
        segun sea (bin[i])
        inicio
            caso '0':
                imprimir ("0000")
                interrumpir
            caso '1':
                imprimir ("0001")
                interrumpir
            caso '2':
                imprimir ("0010")
                interrumpir
            caso '3':
                imprimir ("0011")
                interrumpir
        fin
    fin
}

```



```
    caso '4':
        imprimir ("0100")
        interrumpir
    caso '5':
        imprimir ("0101")
        interrumpir
    caso '6':
        imprimir ("0110")
        interrumpir
    caso '7':
        imprimir ("0111")
        interrumpir
    caso '8':
        imprimir ("1000")
        interrumpir
    caso '9':
        imprimir ("1001")
        interrumpir
    caso 'A':
        imprimir ("1010")
        interrumpir
    caso 'B':
        imprimir ("1011")
        interrumpir
    caso 'C':
        imprimir ("1100")
        interrumpir
    caso 'D':
        imprimir ("1101")
        interrumpir
    caso 'E':
        imprimir ("1110")
        interrumpir
    caso 'F':
        imprimir ("1111")
        interrumpir
    fin
```

fin

fin

Principal

inicio

entero opc

mientras(opc!=7)

inicio

imprimir ("MENU:1.Decimal a Binario 2.Binario a Decimal 3.Decimal a Hexadecimal")

imprimir ("\n4.Hexadecimal a Decimal 5.Binario a Hexadecimal 6.Hexadecimal a Binario 7.Salir")

leer(opc)

i←dec←0

segun sea(opc)

inicio

caso 1:

dec\_bin()

imprimir ("\n")

interrumpir

caso 2:

bin\_dec()

imprimir ("\n")

interrumpir

caso 3:

dec\_hex()

imprimir ("\n")

interrumpir

caso 4:

hex\_dec()

imprimir ("\n")

interrumpir

caso 5:

bin\_hex()

imprimir ("\n")

interrumpir

caso 6:

hex\_bin()

imprimir ("\n")

interrumpir

otro:

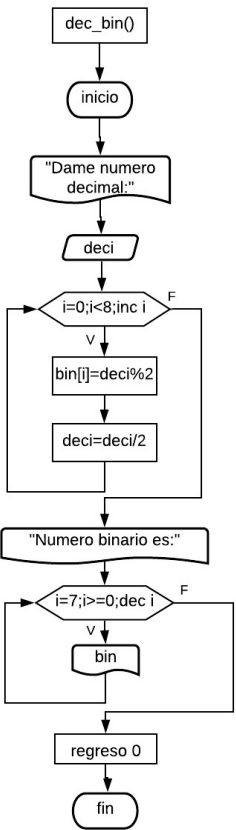
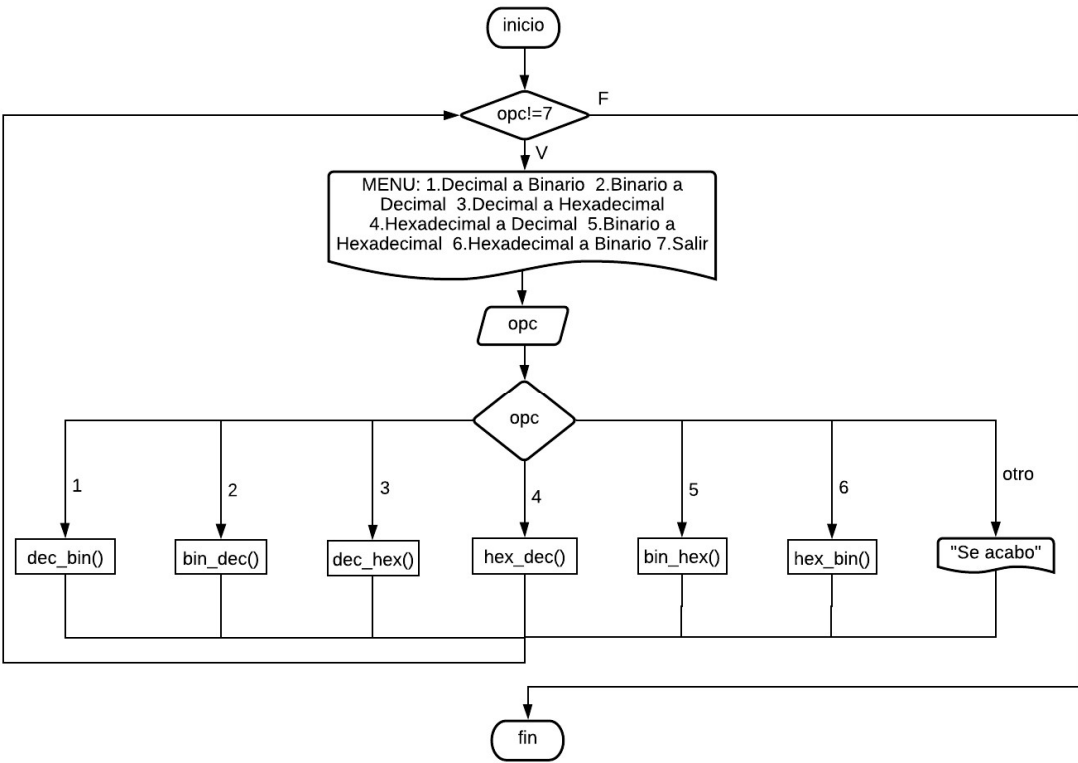
imprimir ("Se acabo\n")

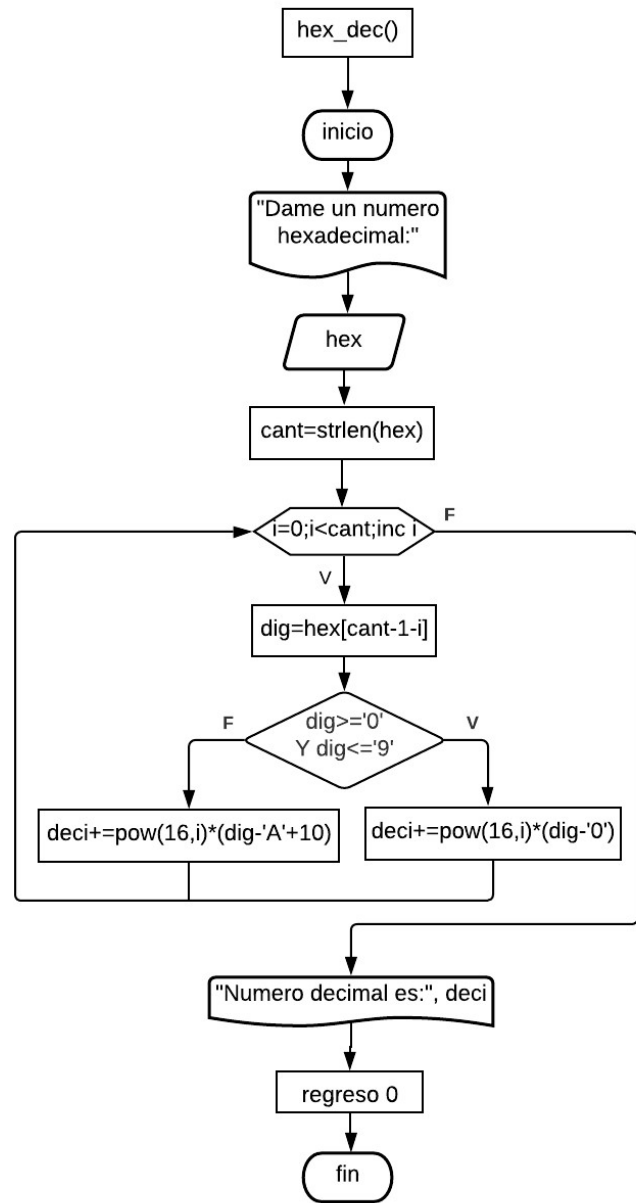
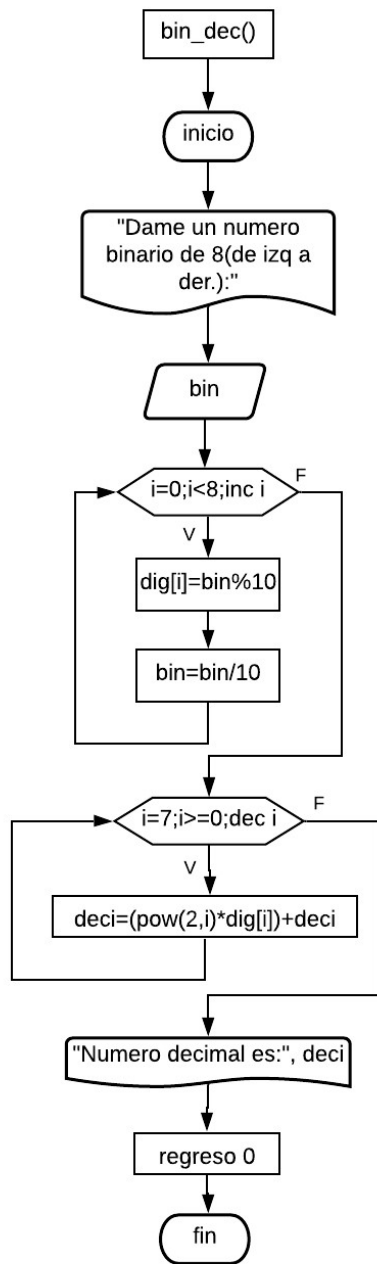
fin

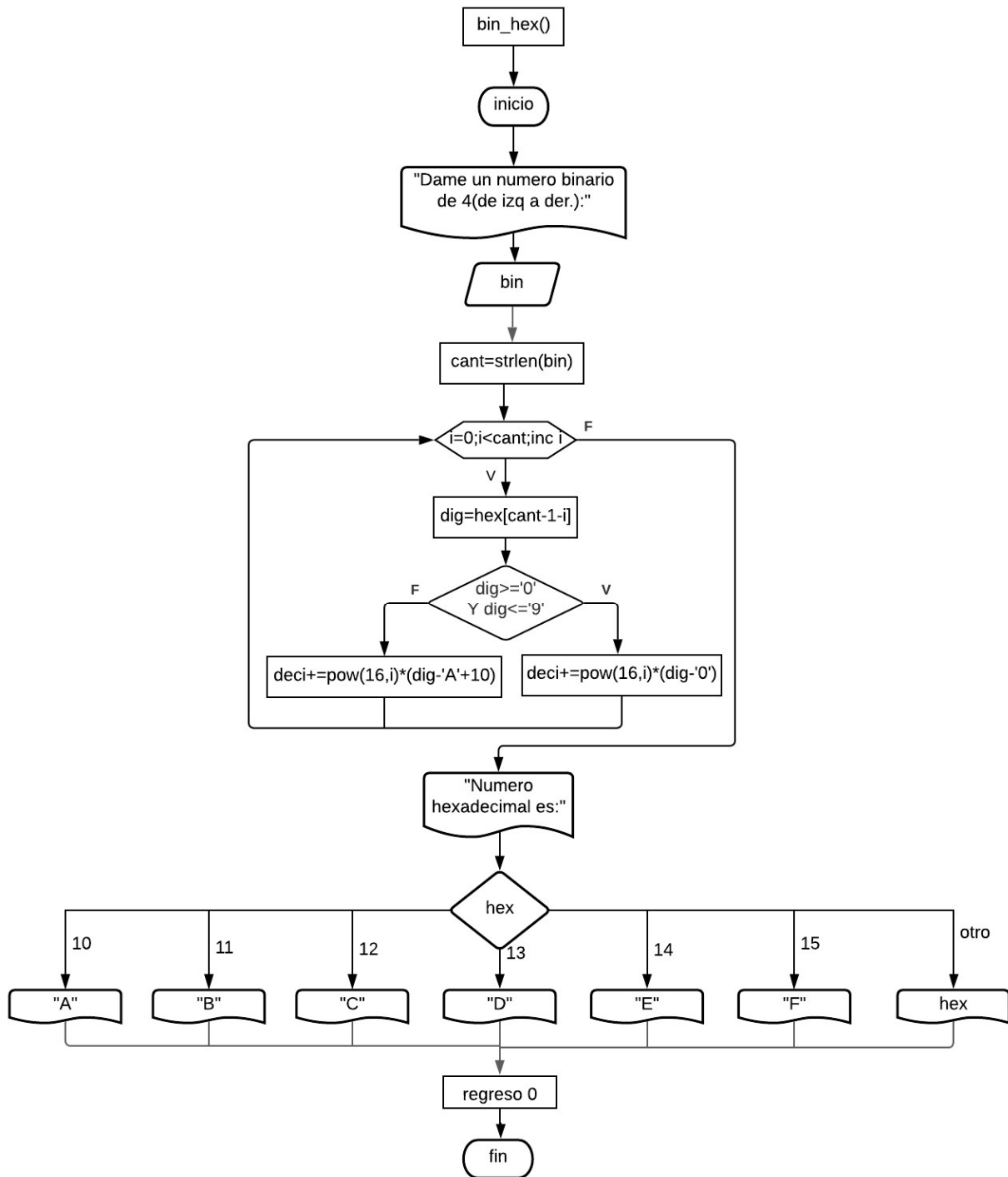
fin

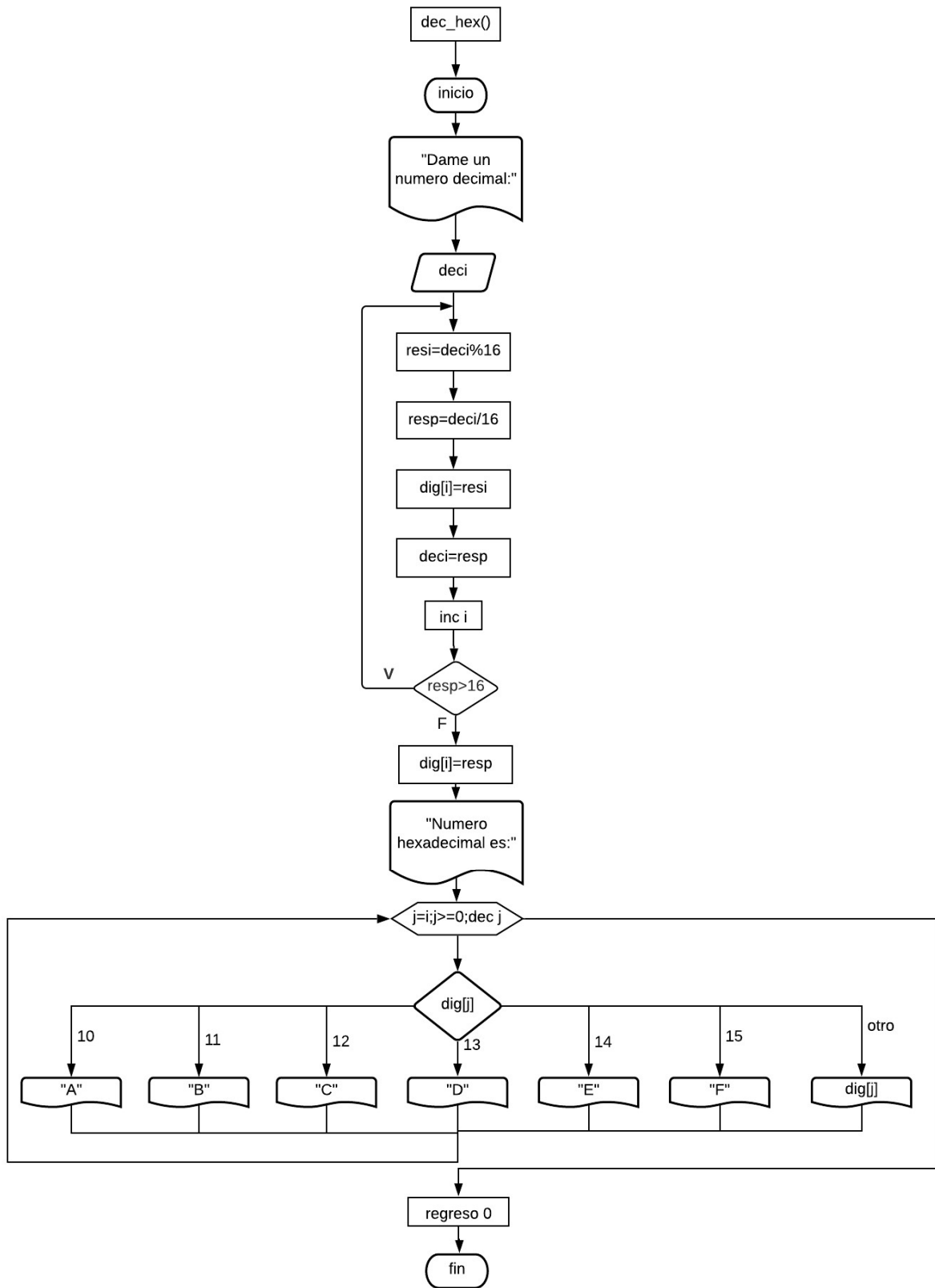
fin

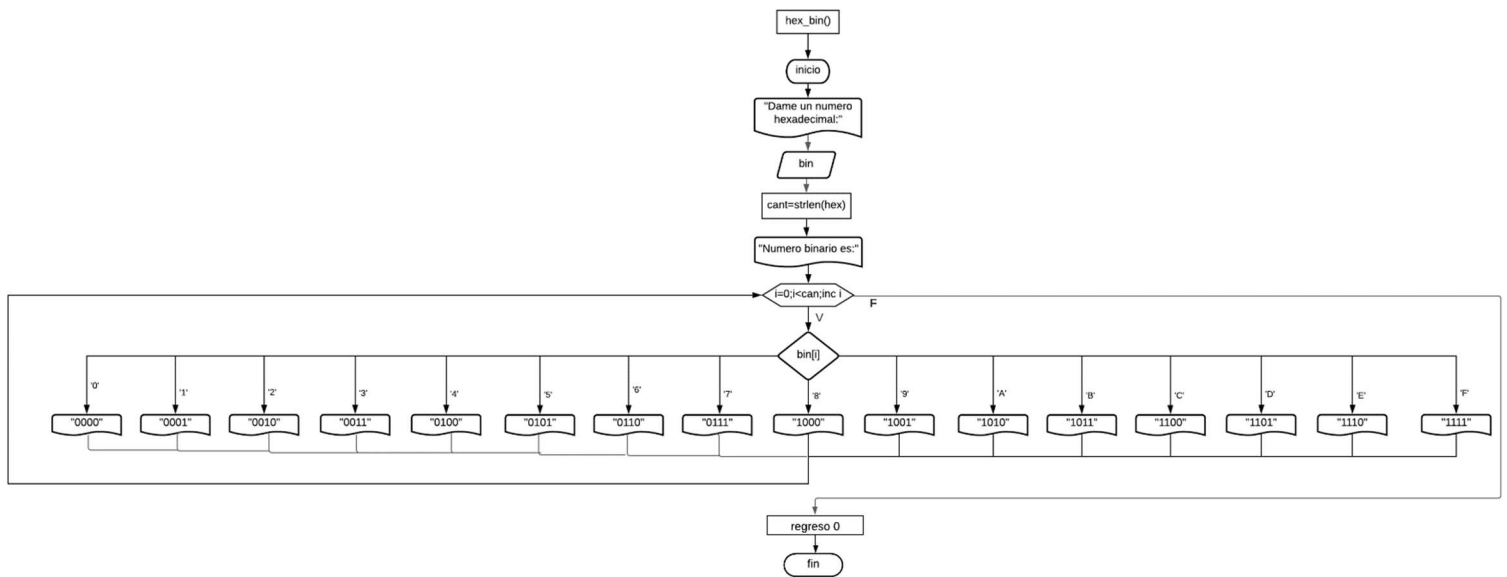
Diagrama de flujo











### Código fuente del programa en lenguaje C

```

//Efrain Robles Pulido
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <ctype.h>
int i,deci,cant;
int dec_bin(){
    int bin[8];
    printf("Dame numero decimal:");
    scanf("%d",&deci);
    for(i=0;i<8;i++){
        bin[i]=deci%2;
        deci=deci/2;
    }
    printf("Numero binario es:");
    for(i=7;i>=0;i--){
        printf("%d", bin[i]);
    }
}

int bin_dec(){
    int dig[8], bin;
    printf("Dame un numero binario de 8(de izq a der.):");
    scanf("%d",&bin);
    for(i=0;i<8;i++){

```

```

        dig[i]=bin%10;
        bin=bin/10;
    }
    for(i=7;i>=0;i--){
        deci=(pow(2,i)*dig[i])+deci;
    }
    printf("Numero decimal es: %d\n",deci);
}

int dec_hex(){
    int dig[20],resi,resp,j;
    printf("Dame un numero decimal:");
    scanf("%d",&deci);
    do{
        resi=deci%16;
        resp=deci/16;
        dig[i]=resi;
        deci=resp;
        i++;
    }while(resp>16);
    dig[i]=resp;
    printf("\nNumero hexadecimal es:");
    for(j=i;j>=0;j--){
        switch(dig[j]){
            case 10:
                printf("A");
                break;
            case 11:
                printf("B");
                break;
            case 12:
                printf("C");
                break;
            case 13:
                printf("D");
                break;
            case 14:
                printf("E");
                break;
            case 15:
                printf("F");
                break;
            default:
                printf("%d",dig[j]);
        }
    }
}

```



```
}  
}
```

```
int hex_dec(){  
    char hex[100],dig;  
    printf("Dame un numero hexadecimal:");  
    scanf("%s",hex);  
    cant=strlen(hex);
```

```
  
    for(i=0;i<cant;i++){  
        dig=hex[cant-1-i];  
        if(dig>='0' && dig<='9'){  
            deci+=pow(16,i)*(dig-'0');  
        }else  
            deci+=pow(16,i)*(dig-'A'+10);  
        }  
    printf("Numero decimal es: %d\n",deci);  
}
```

```
int bin_hex(){  
    int hex;  
    char bin[100],dig;//NO MAYOR A 4 BITS  
    printf("Dame un numero binario de 4(de izq a der.):");  
    scanf("%s",bin);  
    cant=strlen(bin);
```

```
  
    for(i=0;i<cant;i++){  
        dig=bin[cant-1-i];  
        if(dig>='0' && dig<='9'){  
            hex+=pow(2,i)*(dig-'0');  
        }else  
            hex+=pow(2,i)*(dig-'A'+10);  
        }  
    printf("Numero hexadecimal es:");  
    switch(hex){  
    case 10:  
        printf("A");  
        break;  
    case 11:  
        printf("B");  
        break;  
    case 12:  
        printf("C");  
        break;  
    case 13:
```

```

        printf("D");
        break;
    case 14:
        printf("E");
        break;
    case 15:
        printf("F");
        break;
    default:
        printf("%d",hex);
    }
}

int hex_bin(){
    char bin[100];
    printf("Dame un numero hexadecimal:");
    scanf("%s",bin);
    cant=strlen(bin);
    printf("Numero binario es:");
    for(i=0;i<cant;i++){
        switch(bin[i]){
            case '0':
                printf("0000");
                break;
            case '1':
                printf("0001");
                break;
            case '2':
                printf("0010");
                break;
            case '3':
                printf("0011");
                break;
            case '4':
                printf("0100");
                break;
            case '5':
                printf("0101");
                break;
            case '6':
                printf("0110");
                break;
            case '7':
                printf("0111");
                break;

```

```

        case '8':
            printf("1000");
            break;
        case '9':
            printf("1001");
            break;
        case 'A':
            printf("1010");
            break;
        case 'B':
            printf("1011");
            break;
        case 'C':
            printf("1100");
            break;
        case 'D':
            printf("1101");
            break;
        case 'E':
            printf("1110");
            break;
        case 'F':
            printf("1111");
            break;
    }
}

int main(){
    int opc;
    while(opc!=7){
        printf("\t\t\tMENU: \n1.Decimal a Binario  2.Binario a Decimal  3.Decimal a Hexadecimal");
        printf("\n4.Hexadecimal a Decimal  5.Binario a Hexadecimal  6.Hexadecimal a Binario\n7.Salir\n");
        scanf("%d",&opc);
        i=dec=0;
        switch(opc){
        case 1:
            dec_bin();
            printf("\n");
            break;
        case 2:
            bin_dec();
            printf("\n");
            break;
        case 3:

```

```
        dec_hex();  
        printf("\n");  
        break;  
    case 4:  
        hex_dec();  
        printf("\n");  
        break;  
    case 5:  
        bin_hex();  
        printf("\n");  
        break;  
    case 6:  
        hex_bin();  
        printf("\n");  
        break;  
    default:  
        printf("Se acaba\n");  
    }  
}  
}
```

## Resultados obtenidos:

```

      MENU:
1.Decimal a Binario  2.Binario a Decimal  3.Decimal a Hexadecimal
4.Hexadecimal a Decimal  5.Binario a Hexadecimal  6.Hexadecimal a Binario
7.Salir
1
Dame numero decimal:50
Numero binario es:00110010
      MENU:
1.Decimal a Binario  2.Binario a Decimal  3.Decimal a Hexadecimal
4.Hexadecimal a Decimal  5.Binario a Hexadecimal  6.Hexadecimal a Binario
7.Salir
2
Dame un numero binario de 8(de izq a der.):00110010
Numero decimal es: 50
      MENU:
1.Decimal a Binario  2.Binario a Decimal  3.Decimal a Hexadecimal
4.Hexadecimal a Decimal  5.Binario a Hexadecimal  6.Hexadecimal a Binario
7.Salir
3
Dame un numero decimal:90
Numero hexadecimal es:5A
      MENU:
1.Decimal a Binario  2.Binario a Decimal  3.Decimal a Hexadecimal
4.Hexadecimal a Decimal  5.Binario a Hexadecimal  6.Hexadecimal a Binario
7.Salir
4
Dame un numero hexadecimal:5A
Numero decimal es: 90
      MENU:
1.Decimal a Binario  2.Binario a Decimal  3.Decimal a Hexadecimal
4.Hexadecimal a Decimal  5.Binario a Hexadecimal  6.Hexadecimal a Binario
7.Salir
5
Dame un numero binario de 4(de izq a der.):0010
Numero hexadecimal es:2
      MENU:
1.Decimal a Binario  2.Binario a Decimal  3.Decimal a Hexadecimal
4.Hexadecimal a Decimal  5.Binario a Hexadecimal  6.Hexadecimal a Binario
7.Salir
6
Dame un numero hexadecimal:5A
Numero binario es:01011010
      MENU:
1.Decimal a Binario  2.Binario a Decimal  3.Decimal a Hexadecimal
4.Hexadecimal a Decimal  5.Binario a Hexadecimal  6.Hexadecimal a Binario
7.Salir
7
Se acabo

Process returned 0 (0x0)   execution time : 100.441 s
Press any key to continue.
```

## **Conclusión**

Esta práctica fue difícil de hacer ya que se tuvo que comprobar cada función que se tuvo que hacer para lograr el objetivo de la practica en donde se utilizo varias estrategias para hacer la conversión del sistema numérico. Tuve que utilizar estrategias utilizando los arreglos para hacer la conversión, también utilizar los controles selectivos como simple y el multiple y los ciloc for , while y do-while. También fue muy tardado hacer el diagrama de flujo ya que se tenia que hacer uno por cada conversión de la práctica.