Ortega Silva Jorge Eduardo

Como ejecutarlo

Las funciones son llamadas directamente desde el código fuente, las funciones están ordenadas tal como se requiere en la práctica, para ejecutar el punto 3, se requiere quitar el ultimo comentario en el main, dado que los tres puntos que se piden, se complementan con el anterior, se puso todo en un mismo archivo con el nombre de “MultBin.c”

Descripción

Primer ejercicio

```
void binPol(char *b1){
    int i=0, j=0, cont=0, tam=0;
    tam=strlen(b1);
    for(i=0, j=tam-1; i<tam-1; i++, j--){
        if(b1[i]=='1'){
            if(cont!=0)printf(" + ");
            printf("x^%d",j);
            cont++;
        }
    }
    if(b1[tam-1]=='1'){
        if(cont!=0)printf(" + ");
        printf("1");
    }
    if(b1[tam-1]=='0' && cont==0){
        printf("0");
    }
}
```

Para este primer punto, se ve el tamaño de la cadena, para después recorrerla y ver donde se localizan los unos, en donde se encuentre uno, se imprimirá su correspondiente valor en su forma polinomial, todo esto con el uso de las variables del for, para el bit menos significativo, existen dos casos si es cero y no se ha imprimido nada se imprime un cero, pero si es un uno, se imprime uno

Segundo ejercicio

```c
void multBin(unsigned char *b1, unsigned char *b2, unsigned char *res){
    int j=0, i=0, cont=0, aux=0, unos=0;
    unsigned char copyB2[4];
    unsigned char auxB2[4];
    unsigned char r[4]="0011"; //residuo del polinomio irreducible x^4+x+1
    unsigned char mult[4][4];
    unsigned char listaMult[4][4];
    unsigned char resultado[4];

    for(i=0; i<4; i++){//primero sin cambios
        mult[0][i]=b2[i];
    }
    for(i=0; i<4; i++){//guardar el valor de b2
            auxB2[i]=b2[i];
    }
    for(j=0; j<4; j++){
        for(i=0; i<4; i++){
            copyB2[i]=b2[i];
        }
        if(b2[0]=='0') { //si el bit mas significativo es 0
            b2[3]='0';//corrimiento
            for(i=0; i<3; i++){
                b2[i]=copyB2[i+1];
            }

        }else{ //si el bit mas significativo es 1
            b2[3]='0';//Corrimiento
            for(i=0; i<3; i++){
                b2[i]=copyB2[i+1];
            }//XOR con el residuo
            for(i=0; i<4; i++){
                if (b2[i]==r[i]){
                    b2[i]='0';
                }else{
                    b2[i]='1';
                }
            }
        }
        //printf("\n%s", b2);
        for(i=0; i<4; i++){
            mult[j+1][i]=b2[i];
        }
    }
    for(i=0; i<3; i++){//valor original de b2
        b2[i]=auxB2[i];
    }

    for(i=0, aux=3;i<4;i++,aux--){
        if(b1[i]=='1'){
            for(j=0;j<4;j++)
                listaMult[cont][j]=mult[aux][j];
            cont++;
        }
    }
```

```
    for(i=0;i<4;i++){
        unos=0;
        for(j=0;j<cont;j++){
            if(listaMult[j][i]=='1'){
                unos++;
            }
        }
        if((unos%2)==1){
            resultado[i]='1';
        }else{
            resultado[i]='0';
        }
    }

    for(i=0; i<4; i++){
        res[i]=resultado[i]; //valor que devuelve
    }

}
```

Todo lo que se hace en esta función, simula el procedimiento que se vio en clase, primero declaramos las variables necesarias, entre ellas el residuo del polinomio, se obtiene todos los múltiplos de la segunda cadena de bits (1*b2, x*b2, x^2*b2, x^3*b2), esto teniendo en cuenta los dos casos que se nos pueden presentar, cuando el bit mas significativo es uno, o es cero. Después separamos los múltiplos que necesitamos, de acuerdo en donde se encuentren los unos en la primera cadena de bits; finalmente se cuentan los unos por columna del resultado anterior, se obtiene su modulo del número de unos, con la finalidad de simular el funcionamiento de aplicar XOR, estos resultados, se guardan en la cadena que se regresara con el resultado final, proporcionada como parámetro.

Capturas de pantalla

<u>Primer ejercicio</u>

```
Binario: 001
Version polinomial
1
```

```
Binario: 1010
Version polinomial
x^3 + x^1
```

```
Binario: 10
Version polinomial
x^1
```

```
Binario: 1110
Version polinomial
x^3 + x^2 + x^1
```

```
Binario: 1
Version polinomial
1
```

<u>Segundo ejercicio</u>

```
binario 1
1000
binario 2
0001
Resultado b1*b2 mod (x^4 + x + 1)
1000
```

```
binario 1
1001
binario 2
0101
Resultado b1*b2 mod (x^4 + x + 1)
1011
```

```
binario 1
1001
binario 2
0000
Resultado b1*b2 mod (x^4 + x + 1)
0000
```

```
binario 1
1100
binario 2
0110
Resultado b1*b2 mod (x^4 + x + 1)
1110
```

```
binario 1
1111
binario 2
1110
Resultado b1*b2 mod (x^4 + x + 1)
1010
```