



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Edgar Tista García

Asignatura: Estructura de Datos y Algoritmos I

Grupo: 1

No de Práctica(s): 1

Integrante(s): Díaz Hernández Marcos Bryan

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada:

Semestre: 2020-2

Fecha de entrega: 12 de febrero de 2020

Observaciones:

CALIFICACIÓN: _____

Objetivo de la práctica

Utilizar arreglos unidimensionales y multidimensionales para dar solución a problemas computacionales.

Introducción

La práctica consistió en la elaboración de arreglos para poder resolver problemas, además este reporte contendrá el análisis de los ejercicios, donde se dará nota a las dificultades que experimenté al resolverlos, inclusive algunos comentarios acerca del conocimiento reforzado.

Análisis de los ejemplos de la guía de laboratorio

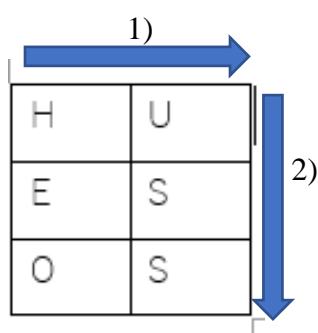
- Primer ejemplo: Código (la escitala espartana).

El programa tiene dos opciones: cifrar o descifrar una palabra, por ello primero solicita una palabra, que modificará, esta se almacena en un arreglo llamado “texto”. En este caso usare huesos como palabra que se almacena en el arreglo texto.

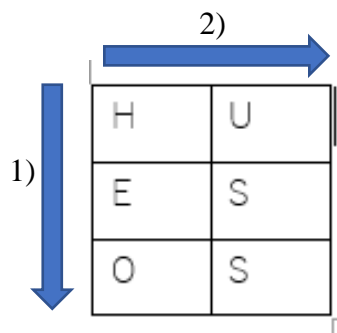
En la primera opción: Cifrar una palabra.

La palabra en el arreglo “texto” con el ciclo for llena el arreglo “escítala”, primero los renglones de izquierda a derecha, y continua con el siguiente renglón. Al imprimir el arreglo primero se imprime la columna de arriba hacia abajo y continua con la siguiente columna.

La salida es: Heouss.



-Guardado en el arreglo



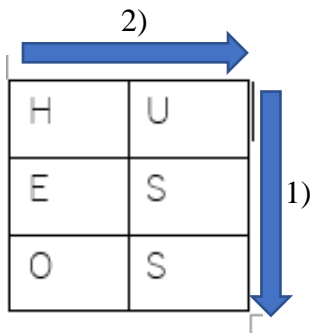
-Impresión del arreglo

La palabra es: Huesos
La salida es: Heouss.

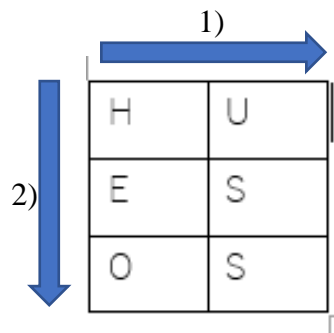
En la segunda opción: Descifrar una palabra.

La palabra en el arreglo “texto” con el ciclo for, llena el arreglo “escítala”, primero las columnas de arriba hacia abajo y continúan con la siguiente columna. Al imprimir el arreglo, primero se imprime el renglón de izquierda a derecha y continua con el siguiente renglón.

La salida es: Huesos.



-Guardado en el arreglo



-Impresión del arreglo

La palabra es: Heouss
La salida es: Huesos

Error en el programa.

```
printf("Escriba el texto a cifrar:\n");
scanf("%s", texto);
```

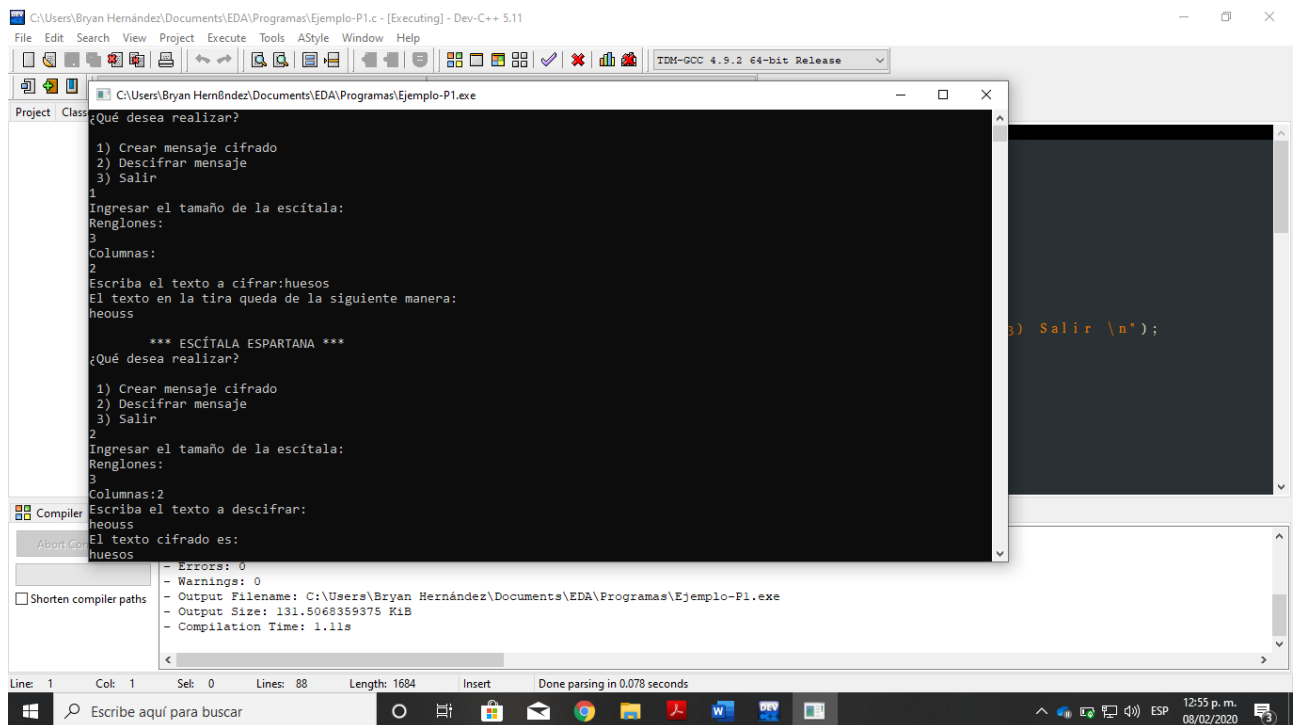
El único error que encontré fue que faltó "&" para guardar la palabra en el arreglo "texto".

Detalles.

```
while (1){
    printf("\n\t*** ESCÍTALA ESPARTANA ***\n");
    printf("¿Qué desea realizar?\n");
    printf("1) Crear mensaje cifrado.\n");
```

Había en el código caracteres especiales y no tenían la forma ASCII para representarlos, no es un error, pero no sé ve bien cuando se ejecuta el programa.

Evidencia de implementación.



Analisis de los ejercicios propuestos

Ejercicio 1.

Este primer ejercicio fue un código proporcionado.

El código nos planteaba llevar a cabo un arreglo de tres dimensiones, el cual ya estaba inicializado, el programa únicamente imprimía los valores. La situación problema era un error en el orden de impresión de estos elementos del arreglo, por ello el cambiar el orden de estos elementos solucionaba la impresión de los valores.

Parte problema.

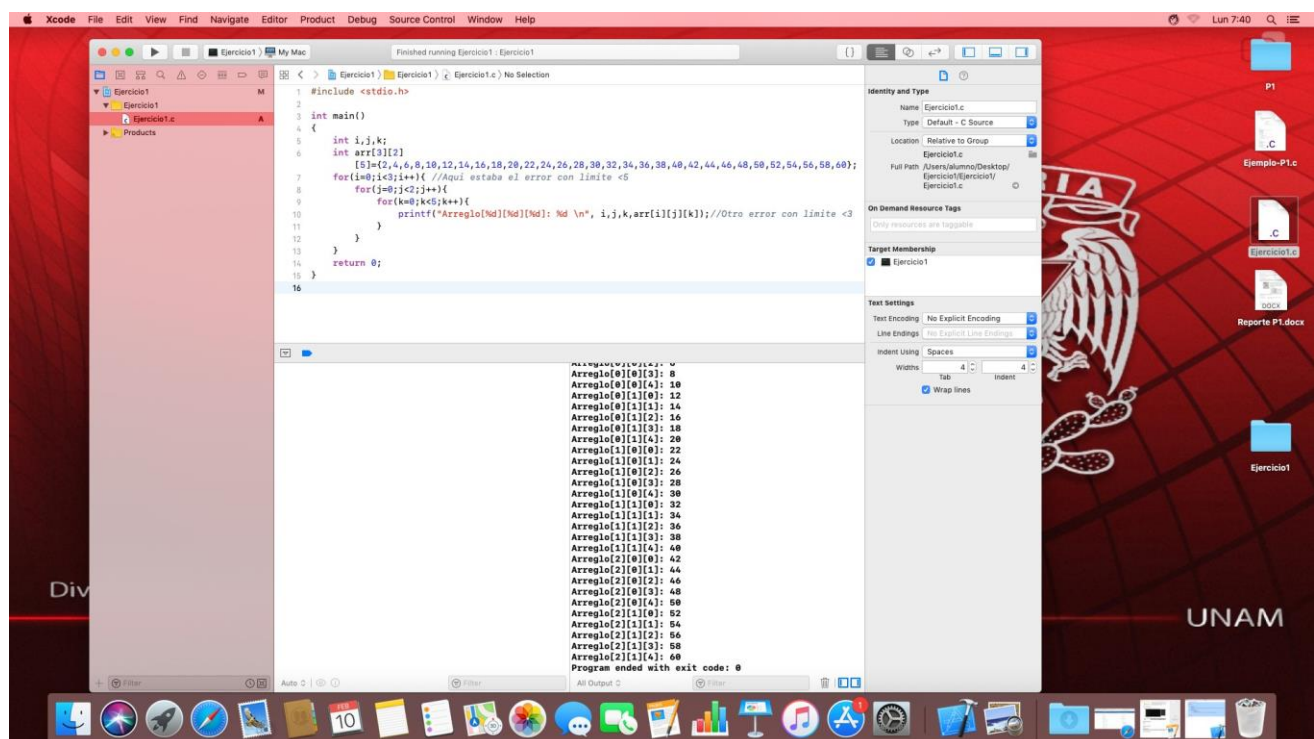
El ejercicio no fue difícil ya que solo era cuestión de modificar los valores, aún así fue algo tardado hacerlo por ser algo que copié y después analicé.

Solución.

- El contador i tenía como límite $i < 5$, y cambiándolo a $i < 3$ se solucionaba.
- El contador k tenía como límite $k < 3$ y cambiándolo a $k < 5$ se solucionaba.

Relación con teoría.

El arreglo era como lo habíamos realizado en clase teórica, por ello fue más sencillo encontrar el punto que causaba problema. En realidad si tuve que pensar el donde se encontraba el error, y de ahí modificar y corregir el programa.



The screenshot shows the Xcode IDE with a C program named 'Ejercicio1.c'. The code defines a 3D array 'Arreglo' of size 5x5x5 and prints its elements. The output window shows the elements printed in a row-major order, grouped by their third dimension index. The program ends with exit code 0.

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int i,j,k;
6     int arr[5][5][5];
7     for(i=0;i<5;i++){ //Aqui estaba el error con limite <5
8         for(j=0;j<5;j++){
9             for(k=0;k<5;k++){
10                 printf("Arreglo[%d][%d][%d]: %d\n", i,j,k,arr[i][j][k]); //Otro error con limite <3
11             }
12         }
13     }
14     return 0;
15 }
16
```

Output:

```
Arreglo[0][0][0]: 0
Arreglo[0][0][1]: 1
Arreglo[0][0][2]: 2
Arreglo[0][0][3]: 3
Arreglo[0][0][4]: 4
Arreglo[0][1][0]: 5
Arreglo[0][1][1]: 6
Arreglo[0][1][2]: 7
Arreglo[0][1][3]: 8
Arreglo[0][1][4]: 9
Arreglo[0][2][0]: 10
Arreglo[0][2][1]: 11
Arreglo[0][2][2]: 12
Arreglo[0][2][3]: 13
Arreglo[0][2][4]: 14
Arreglo[0][3][0]: 15
Arreglo[0][3][1]: 16
Arreglo[0][3][2]: 17
Arreglo[0][3][3]: 18
Arreglo[0][3][4]: 19
Arreglo[0][4][0]: 20
Arreglo[0][4][1]: 21
Arreglo[0][4][2]: 22
Arreglo[0][4][3]: 23
Arreglo[0][4][4]: 24
Arreglo[1][0][0]: 25
Arreglo[1][0][1]: 26
Arreglo[1][0][2]: 27
Arreglo[1][0][3]: 28
Arreglo[1][0][4]: 29
Arreglo[1][1][0]: 30
Arreglo[1][1][1]: 31
Arreglo[1][1][2]: 32
Arreglo[1][1][3]: 33
Arreglo[1][1][4]: 34
Arreglo[1][2][0]: 35
Arreglo[1][2][1]: 36
Arreglo[1][2][2]: 37
Arreglo[1][2][3]: 38
Arreglo[1][2][4]: 39
Arreglo[1][3][0]: 40
Arreglo[1][3][1]: 41
Arreglo[1][3][2]: 42
Arreglo[1][3][3]: 43
Arreglo[1][3][4]: 44
Arreglo[1][4][0]: 45
Arreglo[1][4][1]: 46
Arreglo[1][4][2]: 47
Arreglo[1][4][3]: 48
Arreglo[1][4][4]: 49
Arreglo[2][0][0]: 50
Arreglo[2][0][1]: 51
Arreglo[2][0][2]: 52
Arreglo[2][0][3]: 53
Arreglo[2][0][4]: 54
Arreglo[2][1][0]: 55
Arreglo[2][1][1]: 56
Arreglo[2][1][2]: 57
Arreglo[2][1][3]: 58
Arreglo[2][1][4]: 59
Arreglo[2][2][0]: 60
Arreglo[2][2][1]: 61
Arreglo[2][2][2]: 62
Arreglo[2][2][3]: 63
Arreglo[2][2][4]: 64
Arreglo[2][3][0]: 65
Arreglo[2][3][1]: 66
Arreglo[2][3][2]: 67
Arreglo[2][3][3]: 68
Arreglo[2][3][4]: 69
Arreglo[2][4][0]: 70
Arreglo[2][4][1]: 71
Arreglo[2][4][2]: 72
Arreglo[2][4][3]: 73
Arreglo[2][4][4]: 74
Arreglo[3][0][0]: 75
Arreglo[3][0][1]: 76
Arreglo[3][0][2]: 77
Arreglo[3][0][3]: 78
Arreglo[3][0][4]: 79
Arreglo[3][1][0]: 80
Arreglo[3][1][1]: 81
Arreglo[3][1][2]: 82
Arreglo[3][1][3]: 83
Arreglo[3][1][4]: 84
Arreglo[3][2][0]: 85
Arreglo[3][2][1]: 86
Arreglo[3][2][2]: 87
Arreglo[3][2][3]: 88
Arreglo[3][2][4]: 89
Arreglo[3][3][0]: 90
Arreglo[3][3][1]: 91
Arreglo[3][3][2]: 92
Arreglo[3][3][3]: 93
Arreglo[3][3][4]: 94
Arreglo[3][4][0]: 95
Arreglo[3][4][1]: 96
Arreglo[3][4][2]: 97
Arreglo[3][4][3]: 98
Arreglo[3][4][4]: 99
Arreglo[4][0][0]: 100
Arreglo[4][0][1]: 101
Arreglo[4][0][2]: 102
Arreglo[4][0][3]: 103
Arreglo[4][0][4]: 104
Arreglo[4][1][0]: 105
Arreglo[4][1][1]: 106
Arreglo[4][1][2]: 107
Arreglo[4][1][3]: 108
Arreglo[4][1][4]: 109
Arreglo[4][2][0]: 110
Arreglo[4][2][1]: 111
Arreglo[4][2][2]: 112
Arreglo[4][2][3]: 113
Arreglo[4][2][4]: 114
Arreglo[4][3][0]: 115
Arreglo[4][3][1]: 116
Arreglo[4][3][2]: 117
Arreglo[4][3][3]: 118
Arreglo[4][3][4]: 119
Arreglo[4][4][0]: 120
Arreglo[4][4][1]: 121
Arreglo[4][4][2]: 122
Arreglo[4][4][3]: 123
Arreglo[4][4][4]: 124
Program ended with exit code: 0
```

Ejercicio 2.

Este fue un ejercicio propuesto.

El problema pedia crear un arreglo unidimensional de 15 elementos que el usuario diera, y con estos hacer distintas operaciones. Lo más conflictivo que tuve al hacer este ejercicio fue la parte de la opción 2 que consistia en la multiplicación de los elementos. Igual la opción 3 del ejercicio se me dificulto bastante, esta pide la suma de los elementos divisibles entre 4, debido a esto no pude terminar el ejercicio y me quede estancado en este segundo problema. La opción 4 fue la única que no me dio problema.

```
b = 0;
b = A[i] % 4;
if (b != 0)
{
    c = c + A[i];
}
```

Parte problema.

Esta parte del código fue la que no me dejo avanzar en la práctica, porque yo no sabia como plantearla y norecordaba los operadores elementales de c.

Solución.

Llegando a casa tuve que repasar mis apuntes de arreglos, además me di cuenta de que estaba mal la primer opción de la suma de elementos. La forma en que pude resolverlo fue con un identificador, en donde se almacenara la suma de los elementos.

Relación con teoría.

La teoria de la practica envolvia los arreglos, además de conocimientos básicos de c, pero la faltade estudio y de práctica si afectan, lo que vimos en clases practicamente fue el fundamento para que en la práctica los arreglos no dieran problema, la falla fue mia.

The screenshot displays a C++ development environment. The main window shows the source code for 'Ejercicio2.c'. The code includes a loop that prompts the user to select an operation (1: Suma de elementos, 2: Multiplicacion de los elementos, 3: Suma de los elementos divisibles entre 4, 4: Multiplicar cada elemento por 3). It then displays the original array and performs the selected operation. The 'Compiler' window shows the compilation results, indicating 0 errors and 0 warnings. The 'Output' window shows the execution of the program, displaying the original array and the result of the selected operation (Suma de los elementos: 86).

```
10 while(1)
11 {
12     printf("\nSeleccione la operacion que desea realizar\n");
13     printf("1) Suma de elementos\n");
14     printf("2) Multiplicacion de los elementos\n");
15     printf("3) Suma de los elementos divisibles entre 4\n");
16     printf("4) Multiplicar cada elemento por 3\n");
17     scanf("%d", &a);
18     switch(a)
19     {
20         case 1:
21             printf("\nArreglo original:\n");
22             for(i=0; i<15; i++)
23                 printf("%d ", A[i]);
24             printf("\n");
25             b=0;
26             for(i=0; i<15; i++)
27                 b=b+A[i];
28             printf("\nLa suma de los elementos es: %d\n", b);
29             break;
30         case 2:
31             printf("\nArreglo original:\n");
32             for(i=0; i<15; i++)
33                 printf("%d ", A[i]);
34             printf("\n");
35             for(i=0; i<15; i++)
36                 A[i]=A[i]*2;
37             printf("\nArreglo resultante:\n");
38             for(i=0; i<15; i++)
39                 printf("%d ", A[i]);
40             printf("\n");
41             break;
42         case 3:
43             printf("\nArreglo original:\n");
44             for(i=0; i<15; i++)
45                 printf("%d ", A[i]);
46             printf("\n");
47             for(i=0; i<15; i++)
48                 if(A[i]%4==0)
49                     A[i]=A[i]+A[i];
50             printf("\nArreglo resultante:\n");
51             for(i=0; i<15; i++)
52                 printf("%d ", A[i]);
53             printf("\n");
54             break;
55         case 4:
56             printf("\nArreglo original:\n");
57             for(i=0; i<15; i++)
58                 printf("%d ", A[i]);
59             printf("\n");
60             for(i=0; i<15; i++)
61                 A[i]=A[i]*4;
62             printf("\nArreglo resultante:\n");
63             for(i=0; i<15; i++)
64                 printf("%d ", A[i]);
65             printf("\n");
66             break;
67     }
68 }
```

Compilation results...

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Bryan Hernández\Documents\EDA\P1\Ejercicio2.exe
- Output Size: 129,270,507,812,5 KiB
- Compilation Time: 0.81s
```

Line: 14 Col: 20 Set: 0 Lines: 79 Length: 1616 Insert Done parsing in 0 seconds

Ejercicio 3.

Este fue un ejercicio propuesto.

El problema pide crear un arreglo de 4x6 y a partir de 6 valores que se almacenan en el primer renglon, llenar los demás con el triple de cada elemento del renglon anterior hasta llenar el arreglo. No pude llegar al ejercicio por quedarme atorado en el ejercicio 2, pero fue sencillo de hacer una vez ya entrado en la resolución de los problemas.

Relación con teoría.

Para poder resolverlo fue necesario únicamente utilizar arreglos y bucles, todo parte de la teoría de clase y de antecedentes. Cuando lo resolví me sentí extrañado, ya que en varias ocasiones existe un problema al compilar o al ejecutar, incluso no se obtienen los valores esperados, me gustó poder resolverlo rápido.

3.1 En caso de que se introdujeran valores muy grandes, como los introduje, no me marcaba error, aunque en ciertas cantidades de millones, marcaba negativos en los renglones siguientes, así lo que hice fue cambiar de int a long para poder utilizar valores mucho más grandes, aunque ambos tipos de dato usan 32 bits.

The screenshot shows a C++ IDE with the source code for Ejercicio3.c and its execution output. The source code is as follows:

```
1 #include <stdio.h>
2 int main()
3 {
4     long i, j, A[4][6];
5     printf("Inserte 6 valores del primer renglón:");
6
7     for(i=0; i<4; i++)
8     {
9         for(j=0; j<6; j++)
10         {
11             printf("\nInserte el valor:");
12             scanf("%ld", &A[i][j]);
13         }
14     }
15
16     for(i=0; i<3; i++)
17     {
18         for(j=0; j<6; j++)
19             A[i+1][j] = 3 * A[i][j];
20     }
21
22     for(i=0; i<4; i++)
23     {
24         for(j=0; j<6; j++)
25             printf("\nArreglo[%ld][%ld]: %ld", i, j, A[i][j]);
26     }
```

The execution output shows the input values for the first row and the resulting values for the entire 4x6 array:

```
Inserte 6 valores del primer renglón:
Inserte el valor:345
Inserte el valor:567
Inserte el valor:7686
Inserte el valor:32234
Inserte el valor:567
Inserte el valor:789
Arreglo[0][0]:345
Arreglo[0][1]:567
Arreglo[0][2]:7686
Arreglo[0][3]:32234
Arreglo[0][4]:567
Arreglo[0][5]:789
Arreglo[1][0]:1035
Arreglo[1][1]:1701
Arreglo[1][2]:23058
Arreglo[1][3]:96702
Arreglo[1][4]:1701
Arreglo[1][5]:2367
Arreglo[2][0]:3105
Arreglo[2][1]:5103
Arreglo[2][2]:69174
Arreglo[2][3]:290106
Arreglo[2][4]:5103
Arreglo[2][5]:7101
Arreglo[3][0]:9315
Arreglo[3][1]:15309
Arreglo[3][2]:207522
Arreglo[3][3]:870318
Arreglo[3][4]:15309
Arreglo[3][5]:21303
Process exited after 5.778 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejercicio 4.

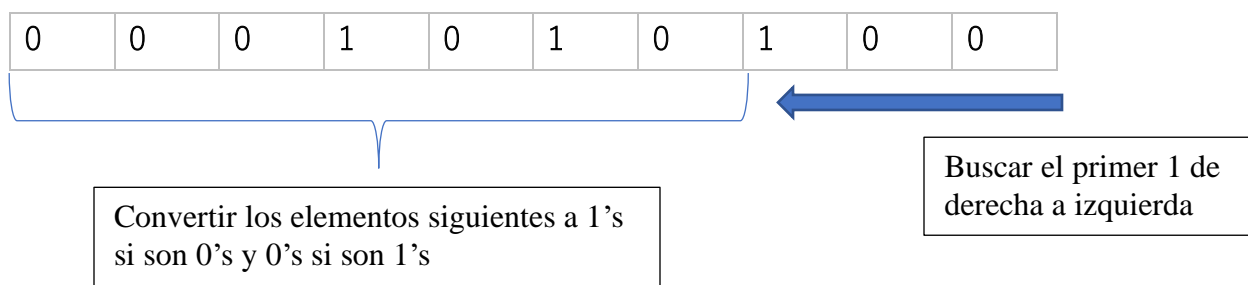
Este fue un ejercicio propuesto.

Lo que solicita el programa es un arreglo de 10 de elementos que se el usuario debe llenar con 1's y 0's, para poder llevar a cabo 3 distintas operaciones. Este ejercicio fue el más difícil para mi, la última operación de complemento a 2 no me salía, y estuve haciendola por horas, el principal problema fue que no pude plantear correctamente el proceso para sacar el complemento.

Parte problema.

```
for (i = 9; i >= 0; i--)
{
    if (A[i] == 1)
    {
        i = i - 1;
        for (i = i; i >= 0; i--)
        {
            if (A[i] == 0)
                A[i] = 1;
            else
                A[i] = 0;
        }
    }
}
```

Poder determinar este proceso fue lo más complejo para mi, por el hecho de que la operación tiene condiciones, y formas de resolverse, así que tuve que considerar la búsqueda del primer 1 y a partir de ese elemento convertir los 0's a 1's y viceversa.



Una vez logré plantear esta parte correctamente en mi mente me lance a programarla, igual estaba bastante estresado de no poder hacerlo. Eso solo me condujo a seguir sin poder resolverlo.

Solución.

Lo que hice fue reiniciar esa parte del problema por completo y plantear todo de nuevo, llego el punto en que estaba en la pista de como resolverlo hasta que tuve que plantear la forma de conservar el primer uno y convertir los demás elementos.

Relación con teoría.

La teoría estaba al 100% envuelta en la solución del problema, por la parte de los arreglos y en parte por la tarea del complemento a 2. La gran dificultad fue plasmar lo que estaba pensando en el lenguaje y hacer que funcionara para lo que estaba buscando.

The screenshot displays a C++ development environment with three main components:

- Source Code (Untitled4.c):** A C++ program that calculates the binary representation of a decimal number. It uses a loop to calculate powers of 2 and another loop to print the binary digits. The code includes comments in Spanish.
- Compilation Results:** Shows successful compilation with 0 errors and 0 warnings. The output file is named 'Untitled4.exe' and has a size of 129.6015625 KiB.
- Program Output:** The program prompts the user to enter a decimal number. The user enters 512. The program then prints the binary representation of 512, which is 1000000000.

```
92 }
93 }
94 else
95 {
96     c = A[i] * B[i];
97 }
98 A[i] = c;
99 }
100 for(i=0; i<10; i++)
101     printf("\nA[%d]: %d", i, A[i]);
102
103 b=0;
104 for(i=1; i<10; i++)
105     b = b + A[i];
106
107 if(A[0] == 1)
108     b = b * (-1);
109 else
110     b = b;
111
112 printf("\nEl numero decimal es: %d\n", b);
113 break;
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Bryan Hernández\Documents\EDA\P1\Untitled4.exe
- Output Size: 129.6015625 KiB
- Compilation Time: 0.70s

Inserte diez dígitos, estos deben ser solo ceros y unos

Inserta el valor:1

Inserta el valor:0

Inserta el valor:1

Inserta el valor:0

Inserta el valor:1

Inserta el valor:1

Inserta el valor:0

Inserta el valor:0

Inserta el valor:1

Inserta el valor:1

Elige una operación:

- 1) Binario puro
- 2) Signo y magnitud
- 3) Complemento a 2

A[0]:512

A[1]:0

A[2]:128

A[3]:0

A[4]:32

A[5]:16

A[6]:0

A[7]:0

A[8]:2

A[9]:1

El numero decimal es:691

Process exited after 114 seconds with return value 26

Presione una tecla para continuar . . .

Conclusiones

El objetivo se logró cumplir, sin embargo me di cuenta de que me hace falta mucha práctica en la programación y estudiar mas. Aunque pude terminar los ejercicios se me dificultaron varias partes de estos. La posible parte negativa de la práctica no esta relacionada con los temas que esta maneja, esta relacionada con la falta de conocimientos que tengo o la poca practica, porque todos los ejercicios tienen solución. Por último algo que me gustaría serian algunas clases pre-práctica en el laboratorio para practica y adquirir o reforzar los conocimientos que se utilizaran en la práctica, se que no es posible pero estaría padre, con respecto a la práctica no tengo más que decir.