



Nombre de la práctica	P04.- Comprobación de tablas de verdad de compuertas básicas en circuitos integrados.			No.	4
Asignatura:	Métodos Numéricos	Carrera:	Ingeniería en Sistemas Computacionales.	Duración de la práctica (Hrs)	
				Fecha:	20/Feb/24

Integrantes.	Matricula
1.-Fabiola Castañeda Mondragón	202223144

ARREGLOS

1.-Asignación de memoria:

```
C ARREGLO.c > ...
  Click here to ask Blackbox to help you code faster
1  #include <stdio.h>
2  int main () {
3  int miArreglo[8];
4  miArreglo[0] = 5;
5  miArreglo[1] = 10;
6  miArreglo[2] = miArreglo[0] + miArreglo[1];
7  system("Pause");
8  return 0;
9  } // Fin main
```

Código ejecutado:

```
fabiola2004@fabiola2004:~/Documentos/MÉTODOS NÚMERICOS/6MARZO$ gcc -o s AR
REGLO.c
fabiola2004@fabiola2004:~/Documentos/MÉTODOS NÚMERICOS/6MARZO$ ./s
fabiola2004@fabiola2004:~/Documentos/MÉTODOS NÚMERICOS/6MARZO$
```

- 2.- Crear un programa que declare un arreglo llamado "vector" de 10 posiciones.
Asignar el valor de 10 a cada posición del arreglo.
Mostrar todas las posiciones del arreglo.

```
C arreglo.c > main()
1  #include <stdio.h>
2
3  int main(){
4
5      int vector[10];
6      int i = 0;
7
8      while(i < 10){
9          vector[i] = 10;
10         i ++;
11     }
12
13     printf("Elementos del arreglo:\n");
14     i = 0;
15     while(i < 10){
16         printf("vector %d: %d\n",i ,vector[i]);
17         i++;
18     }
19
20     return 0;
21 }
```

Código ejecutado:

```
● fabiola2004@fabiola2004:~/Documentos/MÉTODOS NÚMERICOS/6MARZO$ ./s
Elementos del arreglo:
vector 0: 10
vector 1: 10
vector 2: 10
vector 3: 10
vector 4: 10
vector 5: 10
vector 6: 10
vector 7: 10
vector 8: 10
vector 9: 10
```

3.-Código repetitivo:

```
C rep.c > main()
  Click here to ask Blackbox to help you code faster
1  #include <stdio.h>
2  int main () {
3  int vector [10];
4  vector[0] = 10;
5  vector[1] = 10;
6  vector[2] = 10;
7  vector[3] = 10;
8  vector[4] = 10;
9  vector[5] = 10;
10 vector[6] = 10;
11 vector[7] = 10;
12 vector[8] = 10;
13 vector[9] = 10;
14 printf ("%d", vector[0]);
15 printf ("%d", vector[1]);
16 printf ("%d", vector[2]);
17 printf ("%d", vector[3]);
18 return 0;
19 } // Fin main
```

Código ejecutado:

```
fabiola2004@fabiola2004:~/Documentos/MÉTODOS NUMÉRICOS/6MARZO$ ./s
10101010fabiola2004@fabiola2004:~/Documentos/MÉTODOS NUMÉRICOS/6MARZO$
```

4.- Programa:

```
C arre.c > ...
  Click here to ask Blackbox to help you code faster
1  #include <stdio.h>
2  int main () {
3  int vector [10];
4  int i = 0;
5  while (i<10){
6  vector[i] = 10;
7  i++;
8  }
9  i = 0;
10 while (i<10){
11 printf ("%d", vector[i]);
12 i++;
13 }
14
15 return 0;
16 } // Fin main
```



```
168
170
172
174
176
178
180
182
184
186
188
190
192
194
196
198
200
```

6.-Crear un arreglo de 100 posiciones. #
Llenar el orden inverso al índice.
Mostrar el arreglo en pantalla.

```
C arre.c > ...
Click here to ask Blackbox to help you code faster
1  #include <stdio.h>
2
3  int main() {
4      int arreglo[100];
5
6      for (int i = 0; i < 100; i++) {
7          arreglo[i] = 100 - i;
8      }
9      printf("Arreglo lleno con valores en orden inverso al índice:
10     for (int i = 0; i < 100; i++) {
11         printf("%d\n", arreglo[i]);
12     }
13
14     return 0;
15 }
```

Código ejecutado:

```
• fabiola2004@fabiola2004:~/Documentos/MÉTODOS NÚMERICOS/6MARZO$ ./s
Arreglo lleno con valores en orden inverso al índice:
100
99
98
97
96
95
94
93
92
91
90
89
88
87
```

...

```
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
```

7.-

- ☐ Datos
 - ☐ $A = [3, 5, 6, 8, 4, 7, 8, 5, 3, 1]$
 - ☐ $B = [3, 4, 6, 8, 9, 1, 2, 3, 0, 9]$
- ☐ Realizar las siguientes operaciones
 - ☐ $A[3] \bmod (B[2]/2)$
 - ☐ $B[A[1]] - A[9]$
 - ☐ $A[0] + A[1+2]$
 - ☐ $A[5] + B[5]$
 - ☐ $(A[3]/B[2])/2$

```
C Ejercicio.c > main()
  Click here to ask Blackbox to help you code faster
1  #include <stdio.h>
2
3  int main() {
4      int A[] = {3, 5, 6, 8, 4, 7, 8, 5, 3, 1};
5      int B[] = {3, 4, 6, 8, 9, 1, 2, 3, 0, 9};
6      int resultado1;
7      int resultado1 = A[3] % (B[2] / 2);
8      int resultado2 = B[A[1]] - A[9];
9      int resultado3 = A[0] + A[1+2];
10     int resultado4 = A[5] + B[5];
11     int resultado5 = (A[3] / B[2]) / 2;
12
13     printf("Resultado 1: %d\n", resultado1);
14     printf("Resultado 2: %d\n", resultado2);
15     printf("Resultado 3: %d\n", resultado3);
16     printf("Resultado 4: %d\n", resultado4);
17     printf("Resultado 5: %d\n", resultado5);
18
19     return 0;
20 }
```

Código ejecutado:

```
fabiola2004@fabiola2004:~/Documentos/MÉTODOS NÚMER
● ICOS/14marzo$ ./s
Resultado 1: 2
Resultado 2: 0
Resultado 3: 11
Resultado 4: 8
Resultado 5: 0
```


- 8.-Crea un arreglo de 20 posiciones.
Asignarle a cada elemento un valor.
Calcula el promedio de todos los elementos.
Calcula la multiplicación de todos los elementos.

```
C tarea.c > main()
Click here to ask Blackbox to help you code faster
1  #include <stdio.h>
2  int main() {
3      int arreglo[20];
4      float promedio = 0;
5      long long int multiplicacion = 1;
6      for(int i = 0; i < 20; i++) {
7          arreglo[i] = i + 1;
8          promedio += arreglo[i];
9          multiplicacion *= arreglo[i];
10     }
11     promedio /= 20;
12     printf("Arreglo: ");
13     for(int i = 0; i < 20; i++) {
14         printf("%d ", arreglo[i]);
15     }
16     printf("\nPromedio: %.2f\n", promedio);
17     printf("Multiplicación: %lld\n", multiplicacion);
18
19     return 0;
20 }
```

Código ejecutado:

```
fabiola2004@fabiola2004:~/Documentos/MÉTODOS NÚMER
• ICOS/14marzo$ ./s
Arreglo: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20
Promedio: 10.50
Multiplicación: 2432902008176640000
```