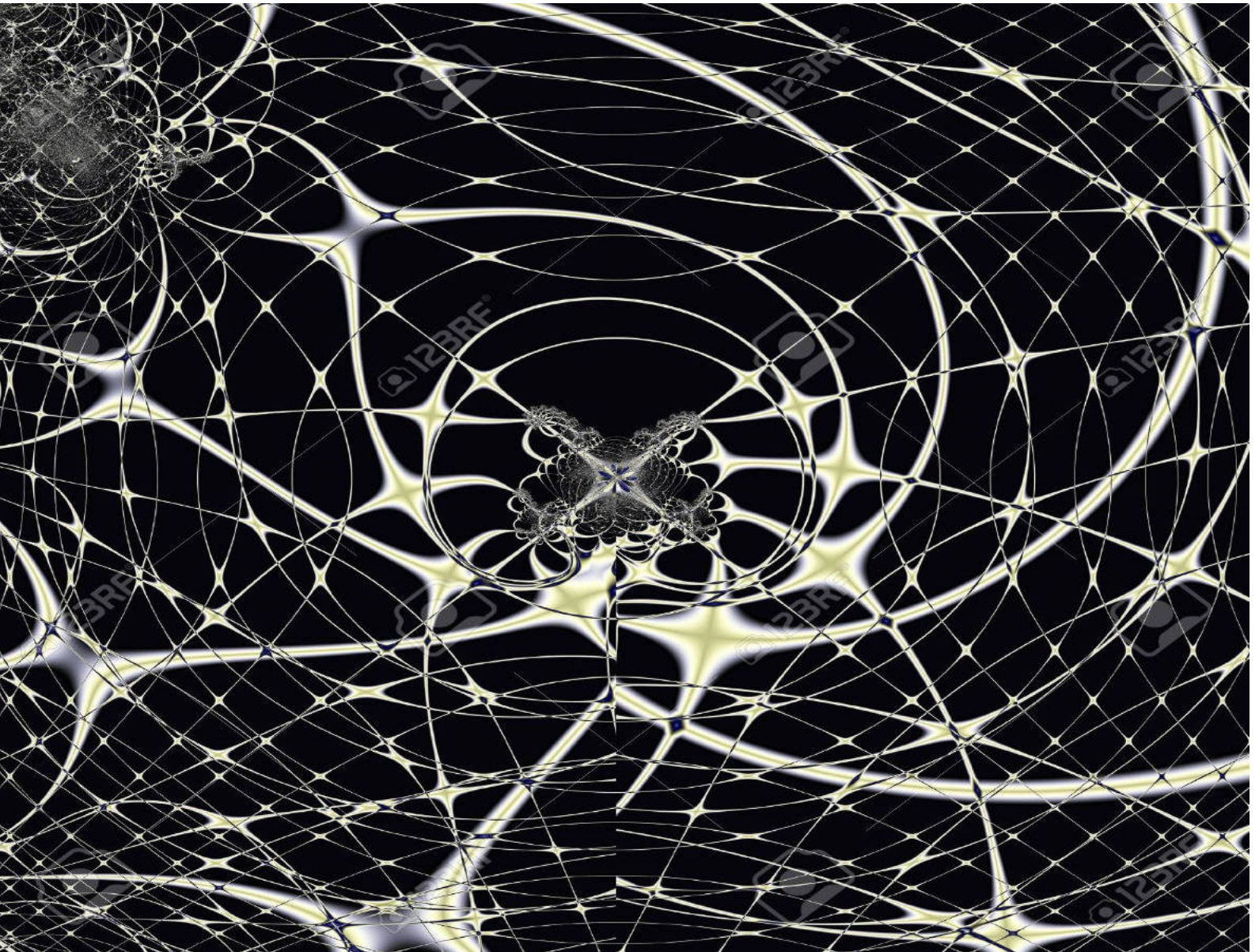


PRÁCTICA DE TALLER DE PROGRAMACIÓN

array



1. EJERCICIOS

- 1.1. Que rellene un array con los 100 primeros números enteros y los muestre en pantalla en orden ascendente y luego descendente.
- 1.2. Que rellene un array con los números pares y otro con los impares comprendidos entre 1 y 100 y los muestre en pantalla en orden ascendente y descendente
- 1.3. Que lea 10 números por teclado, los almacene en un array y muestre la suma, resta, multiplicación y división de todos.
- 1.4. Que lea 10 números por teclado, los almacene en un array y los ordene de forma ascendente.
- 1.5. Que se ingrese una frase en un arreglo y se invierta el orden de la misma guardándose en otro. Mostrar ambos arrays
- 1.6. Que se ingrese una frase en un arreglo y se invierta el orden del mismo sin usar otro array.
- 1.7. Que lea una cadena y diga cuantas vocales hay.

Para resolver este problema investigue y utilice a la función *tolower()* para esto tendrá que incluir a la librería *ctype.h*

```

1  #include <stdio.h>
2  #include <ctype.h>
3
4  int main()
5  {
6      char cadena[] = "ESTO ES UNA CADENA DE PRUEBA";
7      int i;
8
9      for(i = 0; cadena[i]; i++)
10         cadena[i] = tolower(cadena[i]);
11
12     printf("%s\n", cadena);
13     return 0;
14 }
```

```

$ gcc -o minusculas minusculas.c
$ ./minusculas
esto es una cadena de prueba
$
```

1.8. Que lea una cadena y diga cuantas mayúsculas hay.

Tenga en cuenta que un caracter es un código ascii, por lo tanto pueden leerse como números. Vea el siguiente ejemplo:

```

1  /*ascii.c*/
2
3  #include<stdio.h>
4
5  int main() {
6      char x;
7      int j;
8
9      x = 'a';
```

```
10
11     j = (int)x;
12
13     printf("la letra %c es el ascii %d \n", x, j);
14
15     printf("y el nunmero 65 es la letra %c\n", 65);
16
17     return 0;
18 }
```

```
$ gcc -o ascii ascii.c
$ ./ascii
la letra a es el ascii 97
y el nunmero 65 es la letra A
$
```

El rango de mayúsculas se encuentra entre el ascii 65 y el 90.

1.9. Escribir un programa en que dados dos arreglos me diga si son idénticos o no.

1.10. Dada una palabra decir si es palíndromo o no.

¿Qué cambio debería tener que realizar para que se puede validar una frase palíndromo?

2. SOLUCIONES

2.1. Ejercicio 1

```
1  /*arraylal100.c
2  Que rellene un array con los 100 primeros números enteros y los muestre en
   pantalla en orden ascendente y luego descendente.*/
3
4  #include <stdio.h>
5
6  int main(void)
7  {
8      int x, j, tabla[100];
9
10     for (x = 0; x <= 100; x++) {
11         tabla[x] = x;
12     }
13
14     for (x = 0, j = 99; x < 100; x++, j--) {
15         printf("%d\t%d\n", tabla[x], tabla[j]);
16     }
17
18     return 0;
19 }
20
```

```
$ gcc -o arraylal100 arraylal100.c
```

2.2. Ejercicio 2

```
1  /*pares-impares.c
2  Que rellene un array con los números pares y otro con los impares comprendidos
   entre 1 y 100 y los muestre en pantalla en orden ascendente y descendente*/
3
4
5  #include <stdio.h>
6
7  int main()
8  {
9      int x, z, i, j, pares[50], impares[50];
10
11     i = 0;
12     j = 0;
13     for ( x = 1; x <= 100; x++ ){
14
15         if (x%2==0){
16             pares[i] = x;
17             i++;
18         }else{
19             impares[j] = x;
20             j++;
21         }
22     }
23 }
```

```

22     }
23
24     printf("Listado de numeros\n");
25     printf("Pares\t\t\tImpares\n");
26
27     for (x = 0, z = 49; x < i; x++, z--){
28         printf("%d\t%d\t\t%d\t%d\n", pares[x], pares[z], impares[x], impares[z])
29         ;
30     }
31     return 0;
32 }

```

```
$ gcc -o pares-impares pares-impares.c
```

2.3. Ejercicio 3

```

1  /*operaciones.c
2  Que lea 10 números por teclado, los almacene en un array y muestre la suma,
3  resta, ómultiplicacin y ódivisin de todos.*/
4
5  #include <stdio.h>
6
7  int main()
8  {
9      int x, tabla[10];
10     int sum, res, mul;
11     float div;
12
13     for (x=0; x<10; x++) {
14         printf("Introduzca número\n");
15         scanf( "%d", &tabla[x] );
16     }
17
18     sum=tabla[0];
19     res=tabla[0];
20     mul=tabla[0];
21     div=tabla[0];
22
23     for (x = 1; x < 10; x++){
24         sum=sum+tabla[x];
25         res=res-tabla[x];
26         mul=mul*tabla[x];
27         div=div/tabla[x];
28     }
29
30     printf("Los numeros ingresados fueron:\n");
31
32     for (x = 0; x < 10; x++){
33         printf("%d ", tabla[x]);
34     }
35

```

```

36     printf("\nLos resultados de las operaciones son:\n");
37     printf("Suma: %d\n", sum);
38     printf("Resta: %d\n", res);
39     printf("óMultiplicacin: %d\n", mul);
40     printf("óDivisin: %f\n", div);
41
42     return 0;
43 }

```

```
$ gcc -o operaciones operaciones.c
```

2.4. Ejercicio 4

```

1  /*ordenamiento.c
2  Que lea 10 números por teclado, los almacene en un array y los ordene de forma
3  ascendente.
4  */
5
6  #include <stdio.h>
7
8  int main()
9  {
10     float aux, numeros[10];
11     int i, j, n=10;
12
13     for (i=0; i<n; i++){
14         printf("Escriba un número");
15         scanf("%f", &numeros[i]);
16     }
17
18     printf("Numeros ingresados:\n");
19
20     for(i = 0; i < 10; i++)
21         printf("%f\t", numeros[i]);
22
23     for(i = 0; i < n - 1; i++){
24         for(j = i + 1; j < n; j++){
25             if(numeros[i] < numeros[j]){
26                 aux = numeros[i];
27                 numeros[i] = numeros[j];
28                 numeros[j] = aux;
29             }
30         }
31     }
32
33     printf("\nNumeros ordenados:\n");
34     for (i = n - 1; i >= 0; i--){
35         printf("%f\t", numeros[i]);
36     }
37     printf("\n");
38
39     return 0;

```

```
40 | }
```

```
$ gcc -o ordenamiento ordenamiento.c
```

2.5. Ejercicio 5

```
1  /*invertir.c
2  Que se ingrese una frase en un arreglo y se invierta el orden de la misma
   águardndose en otro. Mostrar ambos arrays
3  */
4
5  #include<stdio.h>
6
7  int main(){
8
9      char frase[100], invertida[100];
10     int i, largo;
11
12     printf("ingresar frase: ");
13     fgets(frase, 100, stdin);
14
15     printf("%s\n", frase);
16     /*calculo el largo de la frase ingresada*/
17     largo = 0;
18     i = 0;
19     while (frase[largo] != '\0'){
20         largo++;
21     }
22
23     printf("largo: %d\n",largo);
24
25     /*invierto el arreglo y lo guardo en el otro*/
26     while (largo >= 0){
27         invertida[i] = frase[largo-1];
28         largo--;
29         i++;
30     }
31
32     invertida[i+1] = '\0';
33
34     printf("frase original: %s\nfrase invertida: %s\n", frase, invertida);
35     return 0;
36 }
```

```
$ gcc -o invertir invertir.c
```

2.6. Ejercicio 6

```
1  /*invertir2.c
```

```

2  Que se ingrese una frase en un arreglo y se invierta el orden del mismo sin
   usar otro array.
3  */
4
5  #include<stdio.h>
6
7  int main(){
8
9      char frase[100], aux;
10     int i, largo;
11
12     printf("ingresar frase: ");
13     fgets(frase, 100, stdin);
14
15     printf("frase original: %s\n", frase);
16
17     /*calculo el largo de la frase ingresada*/
18     largo = 0;
19     i = 0;
20     while (frase[largo] != '\0'){
21         largo++;
22     }
23
24     printf("largo: %d\n",largo);
25
26     /*invierto el arreglo */
27     while (largo-2 >= i){
28         aux = frase[i];
29         frase[i] = frase[largo-2];
30         frase[largo-2] = aux;
31         largo--;
32         i++;
33     }
34
35     printf("frase invertida: %s\n", frase);
36     return 0;
37 }

```

```
$ gcc -o invertir2 invertir2.c
```

2.7. Ejercicio 7

```

1  /*contarvocales.c
2  Que lea una cadena y diga cuantas vocales hay.*/
3  #include <stdio.h>
4  #include <ctype.h>
5
6  int main()
7  {
8
9      int sum, x;
10     char frase[50];

```



```

11     printf("Introduzca una frase: ");
12     fgets(frase, 50, stdin);
13
14     for(x = 0; x < 50; x++)
15         frase[x] = tolower(frase[x]);
16
17     sum = 0;
18     for(x = 0; x < 50; x++){
19         switch (frase[x]){
20             case 'a':
21                 sum++;
22                 break;
23             case 'e':
24                 sum++;
25                 break;
26             case 'i':
27                 sum++;
28                 break;
29             case 'o':
30                 sum++;
31                 break;
32             case 'u':
33                 sum++;
34                 break;
35             default:
36                 break;
37         }
38     }
39
40     printf("\nEn la frase hay %d vocales: ", sum);
41     printf("\n");
42
43     return 0;
44 }
45
46

```

```
$ gcc -o contarvocales contarvocales.c
```

2.8. Ejercicio 8

```

1  /*contarMayusculas.c
2  Que lea una cadena y diga cuantas úmayusculas hay.*/
3
4  #include <stdio.h>
5
6  int main()
7  {
8      int sum=0,x;
9      char frase[50];
10
11     printf("Introduzca una frase: ");

```

```

12     fgets(frase, 50, stdin);
13
14     for(x = 0; x < 50; x++){
15         if (frase[x] >= 65 && frase[x] <= 90){
16             sum++;
17         }
18     }
19
20     printf("En la frase hay %d úmaysculas\n",sum);
21
22     return 0;
23 }

```

```
$ gcc -o contarMayusculas contarMayusculas.c
```

2.9. Ejercicio 9

```

1  /*iguales.c
2  Escribir un programa en que dados dos arreglos me diga si son éidnticos o no.
3  */
4
5  #include<stdio.h>
6
7  int main(){
8
9      char palabra1[50], palabra2[50];
10     int bandera, x;
11
12     printf("Ingrese la primera palabra o frase: ");
13     fgets(palabra1, 50, stdin);
14
15     printf("Ingrese la segunda palabra o frase; ");
16     fgets(palabra2, 50, stdin);
17
18     bandera = 1;
19
20     for (x = 0; x < 50; x++){
21
22         if (palabra1[x] != palabra2[x] || (palabra1[x] == '\0' &&
23             palabra2[x] != '\0' ) || (palabra1[x] != '\0' &&
24             palabra2[x] == '\0' ) ){
25
26             bandera = 0;
27             break;
28         }else{
29             if (palabra1[x] == '\0' && palabra2[x] == '\0' )
30                 break;
31         }
32     }
33
34     if (bandera == 1)
35         printf("\nLas palabras son iguales\n");

```

```

36     else
37         printf("\nLas palabras son diferentes\n");
38
39     return 0;
40 }

```

```
$ gcc -o iguales iguales.c
```

2.10. Ejercicio 10

```

1  /*palindromo.c
2  Dada una palabra decir si es ípalndromo o no.
3  */
4  #include <stdio.h>
5
6  int main()
7  {
8      /*Bandera para saber si es o no palindromo*/
9      int palindromo;
10     /*contadores para el inicio y el final de la frase*/
11     int i, j, largo;
12
13     char frase[100],c;
14
15     printf("Ingrese la frase: ");
16     scanf("%s", frase);
17
18     /*Cuento el largo de lo que se ingreso*/
19     for(largo = 0; frase[largo] != '\0'; largo++);
20
21     printf("largo: %d\n", largo);
22
23     /*verifico si es capicua*/
24     palindromo = 1;
25
26     for(i = 0, j = largo; i <= largo/2; i++, j--) {
27         printf("Comprobando %c == %c\n", frase[i], frase[j-1]);
28         if (frase[i] != frase[j-1]) {
29             palindromo = 0;
30             break;
31         }
32     }
33     if (palindromo == 1)
34         printf("\nEs un ípalndrimo.\n");
35     else
36         printf("\nNo es un ípalndrimo.\n");
37
38     return 0;
39 }

```

```
$ gcc -o palindromo palindromo.c
```

