

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

PROGRAMACIÓN I

UNIDAD 2: ARREGLOS UNIDIMENSIONALES



Docente:

Prof.^a. Erika Bonfil Barragán



- EQUIPO 8

Jesús Huerta Aguilar		202041509
Javier De La Luz Ruiz		202033810
Ernesto Flores Cesáreo		202066335

Fecha de elaboración:

15/09/2021

NRC: 18438

Sección: 007

SEGUNDO SEMESTRE

Puebla, Pue.

Fecha de entrega: 17/09/2021

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

1. Escriba un programa que lea dos arreglos de números enteros ordenados ascendentemente y luego almacene en un tercer arreglo la lista ordenada de ambos. Por ejemplo, si el primer arreglo tiene los números 1,3,6,9,17 y el segundo tiene 2,4,10,17, el tercer arreglo debe contener 1,2,3,4,6,9,10,17,17. Muestre en pantalla la impresión del arreglo resultante.

CODIGO:

```
1. //Jesús Huerta Aguilar, Javier de La Luz Ruiz, Ernesto Flores Cesareo
2. //Programación I - "Programa: Concatenación de arreglos y ordenamiento as
   cendente"
3. #include <conio.h>
4. #include <stdio.h>
5. #include <stdlib.h>
6. //PROTOTIPOS
7. void lectura(int [],int);
8. void ord(int [],int);
9. void imp(int [],int);
10. void conc(int [],int [],int [],int,int,int);
11. void conor(int [],int [],int [],int);
12. //PRINCIPAL
13. int main(){
14.     //TAMAÑO PRINCIPAL DE LOS ARREGLOS
15.     int tamc,tama,tamb,k=0;
16.     do{
17.         printf("Ingrese el tama%co del arreglo A: ",164);
18.         scanf("%d",&tama);
19.         printf("Ingrese el tama%co del arreglo B: ",164);
20.         scanf("%d",&tamb);
21.         if (tama <= 0 || tamb <= 0){
22.             printf("\n[!] ERROR: Dimensiones no validas [!]");
23.             getch();
24.             system("cls");
25.         }
26.     } while (tama <= 0 || tamb <= 0);
27.     tamc = tama + tamb;
28.     int a[tama],b[tamb],c[tamc];
29.     printf("\nNOTA: Ingrese los valores ascendentemente.\n");
30.     //LECTURA DE A Y B
31.     printf("\n///// ARREGLO A\n\n");
32.     lectura(a,tama);
33.     printf("\n///// ARREGLO B\n\n");
34.     lectura(b,tamb);
35.     //CONCATENAR ARREGLOS A Y B Y ORDENAR
36.     printf("\n///// ARREGLO C\n\n");
37.     conor(c,a,b,tamc);
38.     imp(c,tamc);
39.     getch();
40.     return 0;
41. }
```

```
42. //LECTURA
43. void lectura(int arreg[],int tam){
44.     int i,band=0;
45.     for (i = 0; i < tam; i++){
46.         do{
47.             printf("\t[%02d] %c ",i+1,26);
48.             scanf("%d",&arreg[i]);
49.             if (i > 0 && arreg[i] < arreg[i-1]){
50.                 printf("\tERROR  %c\n",30);
51.             }
52.         } while (i > 0 && arreg[i] < arreg[i-1]);
53.     }
54. }
55. //IMPRECION
56. void imp(int arreg[],int tam){
57.     int i;
58.     for (i = 0; i < tam; i++){
59.         printf("\t[%02d] %c %d\n",i+1,26,arreg[i]);
60.     }
61. }
62. //CONCATENACIÓN A B
63. void conor(int arreg[],int arone[],int artwo[],int tamc){
64.     int i=0,j=0,act=0;
65.     do{
66.         if(arone[i] > artwo[j]){
67.             arreg[act] = artwo[j];
68.             j++;
69.         }
70.         else{
71.             if(arone[i] < artwo[j]){
72.                 arreg[act] = arone[i];
73.                 i++;
74.             }
75.             else{
76.                 arreg[act] = arone[i];
77.                 act++;
78.                 arreg[act] = artwo[j];
79.                 i++;
80.                 j++;
81.             }
82.         }
83.         act++;
84.     } while(act < tamc);
85. }
```

EJECUCIÓN:

```

Seleccionar a:\Principal\Escritorio\Probleuario 6\problema 1V2.exe
Ingrese el tamaño del arreglo A: 5
Ingrese el tamaño del arreglo B: 3

NOTA: Ingresa los valores ascendentemente.

///// ARREGLO A

    [01] → 2
    [02] → 4
    [03] → 3
    ERROR ▲
    [03] → 6
    [04] → 8
    [05] → 10

///// ARREGLO B

    [01] → 1
    [02] → 3
    [03] → 7

///// ARREGLO C

    [01] → 1
    [02] → 2
    [03] → 3
    [04] → 4
    [05] → 6
    [06] → 7
    [07] → 8
    [08] → 10

a:\Principal\Escritorio\Probleuario 6\problema 1V2.exe
Ingrese el tamaño del arreglo A: -2
Ingrese el tamaño del arreglo B: 3

[!] ERROR: Dimenciones no valdías [!]
```

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

2. Escriba un programa que lea un arreglo de números enteros, y un número x, y escriba en la pantalla todos los índices del arreglo donde está x. por ejemplo, si el arreglo es 1,2,3,100,23,2,2,1 y x es 2. El programa debe escribir: 1,5,6.

CODIGO:

```
1. /*2.programa que lee un arreglo de números enteros y un número x y escriba
   ba
2. por pantalla todos los índices donde se encuentra x.*/
3.
4. //cargar librerías
5. #include<stdio.h>
6. #include <stdlib.h>
7. //Prototipos
8. void llenado (int [],int *enteros);
9. void escritura (int [],int);
10. void busqueda (int [],int *x, int);
11.
12. //Principal
13. int main (){
14.
15.     //Declarar variables
16.     int enteros,x;
17.
18.     //Ingreso de datos por teclado para definir dimensión del arreglo
19.     printf ("\n-----
>Ingrese el tamaño del arreglo unidimensional:",164);
20.     scanf ("%d",&enteros);
21.     //Declarar arreglo
22.     int numeros[enteros];
23.     system ("cls");
24.     //Llamada de primera función void
25.     llenado (numeros,&enteros);
26.     system ("cls");
27.     //Llamada de segunda función void
28.     escritura (numeros,enteros);
29.     //Llamada de tercera función void
30.     busqueda (numeros,&x,enteros);
```

```
31.
32.     //Cierre del programa
33.     printf ("\n\n");
34.     system ("pause");
35.     return 0;
36. }
37.
38. //Funci3n de tipo void para dimensionar el arreglo
39. void llenado (int numeros[],int *enteros){
40.     //Ingreso de datos por teclado para asignar valores al arreglo
41.     int i;
42.     for (i = 0;i < *enteros;i++){
43.         printf ("\nIngrese el numero %d que se registrara en el arreglo:"
44.             ,i+1);
45.         scanf ("%d",&numeros[i]);
46.         system ("cls");
47.     }
48.
49. //Funci3n de tipo void para imprimir los valores en el arreglo
50. void escritura (int numeros[],int enteros){
51.     //Escritura de todos los valores en el arreglo
52.     int i;
53.     printf ("\n\tValores agregados al arreglo:");
54.     for (i = 0;i < enteros;i++){
55.         printf("[%d]",numeros[i]);
56.     }
57.     printf ("\n\t-----");
58. }
59.
60. //Funci3n de tipo void para buscar e imprimir 3ndice solicitado por el u
    suario
61. void busqueda (int numeros[],int *x,int enteros){
62.
63.     //Variables utilizadas en la funci3n
64.     int i,error,error2;
65.     int mayor=numeros[0];
```

```
66.     int menor=mayor;
67.
68.     //Parámetros mayor menor para lanzar mensajes de error
69.     for (i=0;i < enteros;i++){
70.         if (numeros[i] > mayor){
71.             mayor=numeros[i];
72.         }
73.         if (numeros[i] < menor){
74.             menor=numeros[i];
75.         }
76.     }
77.
78.     //Bucle do while para poder ingresar un valor x válido
79.     do{
80.         error=0;
81.         printf ("\n\nIngresa el valor x a buscar en los %ndices:",161);
82.         scanf ("%d",&*x);
83.         if (*x < menor){
84.             printf ("\n\t[!] ERROR: El valor %d se encuentra por debajo d
e los valores en el arreglo [!]\n",*x);
85.             error=1;
86.         }
87.         if (*x > mayor){
88.             printf ("\n\t[!] ERROR: El valor %d se encuentra por encima d
e los valores en el arreglo [!]\n",*x);
89.             error=1;
90.         }
91.     }while (error != 0);
92.
93.     //Impresión en pantalla de los índices
94.     printf ("\n\tEl valor %d se encuentra en el %ndice/los %ndices:",*x
,161,161);
95.     for(i=0;i < enteros;i++){
96.         if (numeros[i] == *x){
97.             printf ("{%d}",i);
98.         }
99.     }
100. }
```

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

EJECUCIÓN:

```
a:\Principal\Escritorio\Problemario 6\Problema 2.exe

----->Ingrese el tamaño del arreglo unidimensional:6_

a:\Principal\Escritorio\Problemario 6\Problema 2.exe

Ingrese el numero 1 que se registrara en el arreglo:1

a:\Principal\Escritorio\Problemario 6\Problema 2.exe

Ingrese el numero 2 que se registrara en el arreglo:23

a:\Principal\Escritorio\Problemario 6\Problema 2.exe

Ingrese el numero 3 que se registrara en el arreglo:1_

a:\Principal\Escritorio\Problemario 6\Problema 2.exe

Ingrese el numero 4 que se registrara en el arreglo:1_

a:\Principal\Escritorio\Problemario 6\Problema 2.exe

Ingrese el numero 5 que se registrara en el arreglo:9

a:\Principal\Escritorio\Problemario 6\Problema 2.exe

Ingrese el numero 6 que se registrara en el arreglo:3

a:\Principal\Escritorio\Problemario 6\Problema 2.exe

Valores agregados al arreglo:[1][23][1][1][9][3]
-----
Ingrese el valor x a buscar en los indices:1

El valor 1 se encuentra en el indice/los indices:{0}{2}{3}

Presione una tecla para continuar . . .
```


BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

3. Escriba un programa que almacene en un arreglo los primeros 80 números primos y los visualice.

CODIGO:

```
1. //Jesús Huerta Aguilar, Javier de La Luz Ruiz, Ernesto Flores Cesareo
2. //Programación I - Arreglo numeros primos.
3. #include <stdio.h>
4.
5. void arreglofinal(int *primo, int x);
6.
7. int main(){
8.
9.     int i, j, cont, x=0, primo[100];
10.
11.     for ( i = 1; i <= 410; i++)
12.     {
13.         cont = 0;
14.         for ( j = 1; j < i; j++)
15.         {
16.             if (i%j == 0)
17.             {
18.                 cont++;
19.             }
20.         }
21.         if (cont < 2)
22.         {
23.             primo[x] = i;
24.             x++;
25.         }
26.     }
27.
28.     arreglofinal(primo, x);
29.
30.     system("pause");
31.     return 0;
32. }
33.
34. void arreglofinal(int *primo, int x){
35.
36.     int i;
37.
38.     printf("\n Estos son los numeros primos hasta 80: \n\n");
39.     for ( i = 0; i < x; i++)
40.     {
41.         printf(" %d -- [ %d ] \n", i, *(primo+i));
42.     }
43. }
```

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

EJECUCIÓN:

```
a:\Principal\Escritorio\Problemario 6\problema 3.exe

Estos son los numeros primos hasta 80:

0 -- [ 1 ]
1 -- [ 2 ]
2 -- [ 3 ]
3 -- [ 5 ]
4 -- [ 7 ]
5 -- [ 11 ]
6 -- [ 13 ]
7 -- [ 17 ]
8 -- [ 19 ]
9 -- [ 23 ]
10 -- [ 29 ]
11 -- [ 31 ]
12 -- [ 37 ]
13 -- [ 41 ]
14 -- [ 43 ]
15 -- [ 47 ]
16 -- [ 53 ]
17 -- [ 59 ]
18 -- [ 61 ]
19 -- [ 67 ]
20 -- [ 71 ]
21 -- [ 73 ]
22 -- [ 79 ]
23 -- [ 83 ]
24 -- [ 89 ]
25 -- [ 97 ]
26 -- [ 101 ]
27 -- [ 103 ]
28 -- [ 107 ]
29 -- [ 109 ]
30 -- [ 113 ]
31 -- [ 127 ]
32 -- [ 131 ]
33 -- [ 137 ]
34 -- [ 139 ]
```

```
a:\Principal\Escritorio\Problemario 6\problema 3.exe

45 -- [ 197 ]
46 -- [ 199 ]
47 -- [ 211 ]
48 -- [ 223 ]
49 -- [ 227 ]
50 -- [ 229 ]
51 -- [ 233 ]
52 -- [ 239 ]
53 -- [ 241 ]
54 -- [ 251 ]
55 -- [ 257 ]
56 -- [ 263 ]
57 -- [ 269 ]
58 -- [ 271 ]
59 -- [ 277 ]
60 -- [ 281 ]
61 -- [ 283 ]
62 -- [ 293 ]
63 -- [ 307 ]
64 -- [ 311 ]
65 -- [ 313 ]
66 -- [ 317 ]
67 -- [ 331 ]
68 -- [ 337 ]
69 -- [ 347 ]
70 -- [ 349 ]
71 -- [ 353 ]
72 -- [ 359 ]
73 -- [ 367 ]
74 -- [ 373 ]
75 -- [ 379 ]
76 -- [ 383 ]
77 -- [ 389 ]
78 -- [ 397 ]
79 -- [ 401 ]
80 -- [ 409 ]
Presione una tecla para continuar
```

4. Implemente un programa en lenguaje C que:
- Solicite el tamaño de dos vectores.
 - Lea cada uno de los dos vectores.
 - Realice el producto punto de los vectores leídos.
 - Muestre el resultado.

El producto punto se define:

$$\begin{bmatrix} 2 \\ 3 \\ 5 \\ 4 \\ 6 \end{bmatrix} + \begin{bmatrix} 4 \\ 8 \\ 1 \\ 2 \\ 5 \end{bmatrix} = 2 \times 4 + 3 \times 8 + 5 \times 1 + 4 \times 2 + 6 \times 5 = 8 + 24 + 5 + 8 + 30 = 75$$

CODIGO:

```
1. //Jesús Huerta Aguilar, Javier de La Luz Ruiz, Ernesto Flores Cesareo
2. //Programación I - "Programa: Producto punto de dos vectores (arreglos)"
3. #include <conio.h>
4. #include <stdio.h>
5. #include <stdlib.h>
6. //PROTOTIPOS
7. void lectura(int [],int);
8. void punto(int [],int [],int);
9. //PRINCIPAL
10. int main(){
11.     //TAMAÑO PRINCIPAL DE LOS ARREGLOS
12.     int tam;
13.     do{
14.         printf("Ingrese el tama%co de los arreglos: ",164);
15.         scanf("%d",&tam);
16.         if (tam <= 0){
17.             printf("\n[!] ERROR: Dimenciones no valdias [!]);
18.             getch();
19.             system("cls");
20.         }
21.     } while (tam <= 0);
22.     int a[tam],b[tam];
23.     //LECTURA DE A Y B
24.     printf("\n///// ARREGLO A\n\n");
25.     lectura(a,tam);
26.     printf("\n///// ARREGLO B\n\n");
27.     lectura(b,tam);
28.     //PRODUCTO PUNTO
29.     printf("\n///// PRODUCTO PUNTO A %c B\n\n",250);
30.     punto(a,b,tam);
31.     getch();
32.     return 0;
```

```
33. }
34. //LECTURA
35. void lectura(int arreg[],int tam){
36.     int i,res;
37.     for (i = 0; i < tam; i++){
38.         printf("\t[%02d] -> ",i+1);
39.         scanf("%d",&arreg[i]);
40.     }
41. }
42. //PRODUCTO PUNTO
43. void punto(int ara[],int arb[],int tam){
44.     int i,mult,res=0;
45.     printf("\t[A%cB] = ",250);
46.     for (i = 0; i < tam; i++){
47.         if (i < tam-1){
48.             printf("%d%c%d + ",ara[i],158,arb[i]);
49.         }
50.         else{
51.             printf("%d%c%d = ",ara[i],158,arb[i]);
52.         }
53.     }
54.     for (i = 0; i < tam; i++){
55.         mult = ara[i]*arb[i];
56.         if (i < tam-1){
57.             printf("%d + ",mult);
58.         }
59.         else{
60.             printf("%d = ",mult);
61.         }
62.         res += mult;
63.     }
64.     printf("%d\n",res);
65. }
```

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

EJECUCIÓN:

```
a:\Principal\Escritorio\Problematario 6\problema 4.exe
Ingrese el tamaño de los arreglos: 4

///// ARREGLO A

    [01] -> 5
    [02] -> 3
    [03] -> 4
    [04] -> 2

///// ARREGLO B

    [01] -> 8
    [02] -> 6
    [03] -> 1
    [04] -> 0

///// PRODUCTO PUNTO A · B

    [A·B] = 5×8 + 3×6 + 4×1 + 2×0 = 40 + 18 + 4 + 0 = 62

a:\Principal\Escritorio\Problematario 6\problema 4.exe
Ingrese el tamaño de los arreglos: 3

///// ARREGLO A

    [01] -> -2
    [02] -> 5
    [03] -> 0

///// ARREGLO B

    [01] -> 12
    [02] -> 1
    [03] -> -54

///// PRODUCTO PUNTO A · B

    [A·B] = -2×12 + 5×1 + 0×-54 = -24 + 5 + 0 = -19

a:\Principal\Escritorio\Problematario 6\problema 4.exe
Ingrese el tamaño de los arreglos: -43

[!] ERROR: Dimenciones no valdias [!]
```

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

5. Implemente un programa en lenguaje C que:
- Lea números hasta teclear el cero.
 - Guardar en un arreglo los números negativos leídos.
 - Guardar en un arreglo los números positivos leídos.
 - Mostrar ambos arreglos.

CODIGO:

```
1. //Jesús Huerta Aguilar, Javier de La Luz Ruiz, Ernesto Flores Cesareo
2. //Programación I - Leer numeros negativos y positivos
3. #include <conio.h>
4. #include <stdio.h>
5. #include <stdlib.h>
6. void arreglopositivo(int [], int x);
7. void arreglonegativo(int [], int z);
8. int main(){
9.     int x=0, z=0, num, positivo[100], negativo[100];
10.    printf("\n Escriba '0' para dejar de meter datos. \n");
11.    do{
12.        printf("\n Ingrese un numero ");
13.        scanf("%d", &num);
14.        if (num > 0){
15.            positivo[x] = num;
16.            x++;
17.        }else{
18.            if (num < 0){
19.                negativo[z] = num;
20.                z++;}
21.        }
22.    } while (num != 0);
23.    arreglopositivo(positivo,x);
24.    arreglonegativo(negativo,z);
25.    getch();
26.    return 0;
27.}
28.void arreglopositivo(int sech[], int x){
29.    int i;
30.    printf("\n Estos son los numeros positivos: \n");
31.    for ( i = 0; i < x; i++){
32.        printf(" (%d) \n", sech[i]);
33.    }
34.}
35.void arreglonegativo(int pepe[], int z){
36.    int i;
37.    printf("\n Estos son los numeros negativos: \n");
38.    for ( i = 0; i < z; i++){
39.        printf(" (%d) \n", pepe[i]);
40.    }
41.}
```

EJECUCIÓN:

 a:\Principal\Escritorio\Problemario 6\problema 5.exe

Escriba '0' para dejar de meter datos.

Ingrese un numero -2

Ingrese un numero 4

Ingrese un numero 5

Ingrese un numero -54

Ingrese un numero 15

Ingrese un numero -13

Ingrese un numero 0

Estos son los numeros positivos:

(4)

(5)

(15)

Estos son los numeros negativos:

(-2)

(-54)

(-13)