

Unidad-6: Estructuras de Datos Estáticos.

ACTIVIDADES - I

1. Define un *array* de 12 números enteros y asigna los valores según la tabla que se muestra a continuación. Muestra el contenido de todos los elementos del *arrays*. ¿Qué sucede con los valores de los elementos que no han sido inicializados?

Índice	0	1	2	3	4	5	6	7	8	9	10	11
Valor	39	-2			0		14		5	120		

2. Define un *array* de 10 caracteres y asigna los valores según la tabla que se muestra a continuación. Muestra el contenido de todos los elementos del *arrays*. ¿Qué sucede con los valores de los elementos que no han sido inicializados?

Índice	0	1	2	3	4	5	6	7	8	9
Valor	'a'	'x'			'@'		' '	'+'	'Q'	

3. Escribe un programa que lea 10 números por teclado y que luego los muestre en orden inverso, es decir, el primero que se introduce es el último en mostrarse y viceversa.
4. Define tres *arrays* de 20 números enteros cada uno, con nombres numero, cuadrado y cubo. Carga el *array* numero con valores aleatorios entre 0 y 100. En el *array* cuadrado se deben almacenar los cuadrados de los valores que hay en el *array* numero. En el *array* cubo se deben almacenar los cubos de los valores que hay en numero. A continuación, muestra el contenido de los tres *arrays* dispuestos en tres columnas.
5. Escribe un programa que pida 10 números por teclado y que luego muestre los números introducidos junto con las palabras máximo y mínimo al lado del máximo y del mínimo respectivamente.
6. Escribe un programa que lea 15 números por teclado y que los almacene en un *array*. Rota los elementos de ese *array*, es decir, el elemento de la posición 0 debe pasar a la posición 1, el de la 1 a la 2, etc. El número que se encuentra en la última posición debe pasar a la posición 0. Finalmente, muestra el contenido del *array*.

7. Escribe un programa que genere 100 números aleatorios del 0 al 20 y que los muestre por pantalla separados por espacios. El programa pedirá entonces por teclado dos valores y a continuación cambiará todas la ocurrencias del primer valor por el segundo en la lista generada anteriormente. Los números que se han cambiado deben aparecer entrecomillados.
8. Realiza un programa que pida la temperatura media que ha hecho en cada mes de un determinado año y que muestre a continuación un diagrama de barras horizontales con esos datos. Las barras del diagrama se pueden dibujar a base de asteriscos o cualquier otro carácter.
9. Realiza un programa que pida 8 números enteros y que luego muestre esos números junto con la palabra “par” o “impar” según proceda.
10. Escribe un programa que genera 20 números enteros aleatorios entre 0 y 100 y que los almacene en un *array*. El programa debe de ser capaz de pasar todos los números pares a las primeras posiciones del *array* (del 0 en adelante) y todos los números impares a las celdas restantes.
11. Realiza un programa que pida 10 números por teclado y que los almacene en un *array*. A continuación se mostrará el contenido de ese *array* junto al índice (0 – 9) utilizando para ello una tabla. Seguidamente el programa pasará los primos a las primeras posiciones, desplazando el resto de números (los que no son primos) de tal forma que no se pierda ninguno. Al final se debe mostrar el *array* resultante. Ejemplo:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	20	5	7	4	32	9	2	14	11	6

Array inicial

Índice	0	1	2	3	4	5	6	7	8	9
Valor	5	7	2	11	20	4	32	9	14	6

Array final

12. Realiza un programa que pida 10 números por teclado y que los almacene en un *array*. A continuación se mostrará el contenido de ese *array* junto al índice (0 – 9). Seguidamente el programa pedirá dos posiciones a las que llamaremos “inicial” y “final”. Se debe comprobar que inicial es menor que final y que ambos números están entre 0 y 9. El programa deberá colocar el número de la posición inicial en la posición final, rotando el resto de números para que no se pierda ninguna. Al final se debe mostrar el *array* resultante. Ejemplo:

Por ejemplo, para inicial = 3 y final = 7:

Índice	0	1	2	3	4	5	6	7	8	9
Valor	20	5	7	4	32	9	2	14	11	6

Array inicial

Índice	0	1	2	3	4	5	6	7	8	9
Valor	6	20	5	7	32	9	2	4	14	11

Array final

13. Escribe un programa que rellene un *array* de 100 elementos con números enteros aleatorios comprendidos entre 0 y 500 (ambos incluidos). A continuación el programa mostrará el *array* y preguntará si el usuario quiere destacar el máximo o el mínimo. Seguidamente se volverá a mostrar el *array* escribiendo el número destacado entre dobles asteriscos. Ejemplo:

```
459 204 20 250 178 90 353 32 229 357 224 454 260 310 140 249 332 426 423 413 96 447 465 29\
8 459 411 118 480 302 417 42 82 126 82 474 362 76 190 104 21 257 88 21 251 6 383 47 78 392\
394 244 494 87 253 376 379 98 364 237 13 299 228 409 402 225 426 267 330 243 209 426 435 \
309 356 173 130 416 15 477 34 28 377 193 481 368 466 262 422 275 384 399 397 87 218 84 312\
480 207 68 108
¿Qué quiere destacar? (1 - mínimo, 2 - máximo): 1
459 204 20 250 178 90 353 32 229 357 224 454 260 310 140 249 332 426 423 413 96 447 465 29\
8 459 411 118 480 302 417 42 82 126 82 474 362 76 190 104 21 257 88 21 251 **6** 383 47 78\
392 394 244 494 87 253 376 379 98 364 237 13 299 228 409 402 225 426 267 330 243 209 426 \
435 309 356 173 130 416 15 477 34 28 377 193 481 368 466 262 422 275 384 399 397 87 218 84\
312 480 207 68 108
```

- 14.** Escribe un programa que pida 8 palabras y las almacene en un *array*. A continuación, las palabras correspondientes a colores se deben almacenar al comienzo y las que no son colores a continuación. Los colores que conoce el programa deben estar en otro *array* y son los siguientes: verde, rojo, azul, naranja, rosa, negro, blanco y morado. Ejemplo:

Array original:

0	1	2	3	4	5	6	7
casa	azul	verde	orden	morado	bombilla	bici	rosa

Array resultado:

0	1	2	3	4	5	6	7
azul	verde	morado	rosa	casa	orden	bombilla	bici

- 15.** Introducir por teclado un número *n*; a continuación solicitar al usuario que teclee *n* números. Realizar la media de los números positivos, la media de los negativos y contar el número de ceros introducidos.

16. Un restaurante nos ha encargado una aplicación para colocar a los clientes en sus mesas. En una mesa se pueden sentar de 0 (mesa vacía) a 4 comensales (mesa llena). Cuando llega un cliente se le pregunta cuántos son. De momento el programa no está preparado para colocar a grupos mayores de 4, por lo tanto, si un cliente dice por ejemplo que son un grupo de 6, el programa dará el mensaje **“Lo siento, no admitimos grupos de 6, haga grupos de 4 personas como máximo e intente de nuevo”**. Para el grupo que llega, se busca siempre la primera mesa libre (con 0 personas). Si no quedan mesas libres, se busca donde haya un hueco para todo el grupo, por ejemplo si el grupo es de dos personas, se podrá colocar donde haya una o dos personas. Inicialmente, las mesas se cargan con valores aleatorios entre 0 y 4. Cada vez que se sientan nuevos clientes se debe mostrar el estado de las mesas. Los grupos no se pueden romper aunque haya huecos sueltos suficientes. El funcionamiento del programa se ilustra a continuación:

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	0	2	4	1	0	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 2

Por favor, siéntense en la mesa número 3.

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	1	0	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 4

Por favor, siéntense en la mesa número 7.

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	1	4	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 3

Tendrán que compartir mesa. Por favor, siéntense en la mesa número 6.

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	4	4	2	1	1

¿Cuántos son? (Introduzca -1 para salir del programa): 4

Lo siento, en estos momentos no queda sitio.

Mesa nº	1	2	3	4	5	6	7	8	9	10
Ocupación	3	2	2	2	4	4	4	2	1	1

17. Desarrollar el juego de la cámara secreta, que consiste en abrir una cámara mediante su combinación secreta, que está formada por una combinación de dígitos del uno al cinco. El jugador especificará cuál es la longitud de la combinación, a mayor longitud mayor será la dificultad del juego. La aplicación genera, de forma aleatoria, una combinación secreta que el usuario tendrá que acertar. En cada intento se muestra como pista, para cada dígito de la combinación introducido por el jugador, si es mayor, menor o igual que el correspondiente en la combinación secreta. Puedes visualizar la combinación secreta con asteriscos la primera vez y cada vez que un dígito es acertado, sustituimos el asterisco por el número, y si no se ha acertado indicamos en esa posición si es mayor o menor. Visualiza también los intentos que está utilizando el usuario o usuaria para abrir la cámara. Intenta hacer una interfaz intuitiva.