
Arreglos

Ejercicio 3.1

Realice un programa que permita cargar números enteros positivos en un arreglo (de hasta 1000 elementos) hasta que el usuario ingrese 0 (este valor marca el final de los valores útiles).

Luego, se deberá obtener otro arreglo que contenga los mismos valores pero en forma inversa y a continuación imprimir este último.

Ejemplo:

Ingrese número: **20**

Ingrese número: **21**

Ingrese número: **14**

Ingrese número: **0**

El arreglo revertido contiene: 14, 21, 20.

Ejercicio 3.2

Realice un programa que permita cargar números enteros positivos en un arreglo (de hasta 1000 elementos) hasta que el usuario ingrese 0 (este valor marca el final de los valores útiles). No se permitirá al usuario cargar 2 veces el mismo valor, en dicho caso se mostrara un mensaje. Luego, imprima el arreglo resultante.

Ejercicio 3.3

Elaborar un programa que presente un menú con las siguientes opciones: *AND*, *OR*, *NOT*, *SALIR*.

Luego, para cada opción elegida el programa deberá permitir la carga de uno o dos (según corresponda) arreglos booleanos de 6 elementos cada uno. El 0 tendrá un valor de verdad FALSO y el 1 VERDADERO. Se pide obtener el arreglo booleano resultante de la operación lógica sobre el/los arreglo/s correspondiente/s..

Ejemplo:

1 – AND

2 – OR

3 – NOT

4 – SALIR

Seleccione una opción: **1**

Ingrese los elementos del primer arreglo: **1 1 1 0 0 0**

Ingrese los elementos del segundo arreglo: **1 1 0 0 0 1**

El arreglo resultante es: 1 1 0 0 0 0

1 – AND

2 – OR

3 – NOT

4 – SALIR

Seleccione una opción: **3**

Ingrese los elementos del arreglo: **1 1 1 0 0 0**

El arreglo resultante es: 0 0 0 1 1 1

Ejercicio 3.4

Elaborar un programa que permita cargar dos arreglos numéricos de hasta 1000 elementos. Se efectuará la carga de cada uno (hasta que se ingrese 0 o se complete la capacidad del arreglo).

Luego se presenta el siguiente menú con las siguientes opciones: *UNION*, *INTERSECCION*, *DIFERENCIA*, *SALIR*. A continuación se efectuará la operación entre ambos arreglos como si fueran conjuntos (considerar que los conjuntos no pueden tener elementos repetidos).

Ejercicio 3.5

Realizar un programa en el que se ingrese un texto de hasta 100 caracteres finalizando con ENTER (utilizar ‘\r’ para detectar el ingreso de ENTER). Luego, el usuario deberá elegir un carácter al cual reemplazar y con qué carácter reemplazarlo.

Arreglos

Ejemplo:

Ingrese texto: ***La mar estaba serena***

Ingrese carácter a reemplazar: ***a***

Ingrese carácter de reemplazo: ***e***

Resultado: Le mer estebe serene

Ejercicio 3.6

Realizar un programa en el que se ingrese un texto de hasta 75 caracteres finalizando con ENTER (utilizar el caracter ‘\r’ para detectarlo). Luego imprimir dicho texto centrado y, en la siguiente línea, alineado hacia la derecha. Tener en cuenta que el ancho de la pantalla de texto es de 80 columnas.

El texto debe contener ‘\0’ como caracter final a fin de que pueda imprimirse con `printf(“%s”, texto);`

Ejemplo:

Ingrese texto: ***La bella y graciosa moza marchose a lavar la ropa***

La bella y graciosa moza marchose a lavar la ropa

La bella y graciosa moza marchose a lavar la ropa

Ejercicio 3.7

Programar una función que reciba como parametro un texto y que elimine todo lo que no sean letras que encuentre al principio y al final del texto. Utilizarla en un programa en el que se ingrese un texto de hasta 1000 caracteres y luego se imprima la versión “limpia” del texto. El texto debe contener ‘\0’ como caracter final a fin de que pueda imprimirse con `printf(“%s”, texto);`

Ejemplo:

Ingrese texto:... ***La bella y graciosa moza marchose a lavar la ropa, \$@***

La bella y graciosa moza marchose a lavar la ropa