Ejercitación Unidad 1 y 2

CARLOS ARNALDO GONZALEZ - 11/05/2021 19:26

Ejercicios de la Unidad 1: Cadenas de caracteres

Ejercicio 1:

Realizar un programa que solicite al usuario el ingreso de su nombre y apellido, ambos se guardarán en dos cadenas de caracteres de 30 elementos cada una. Colocar en otra cadena primero el apellido seguido de una coma, un espacio y finalmente el nombre. Mostrar el resultado por pantalla. Ejemplo:

Nombre: Juan

Apellido: Perez

Resultado: Perez, Juan

Solución:

#include <stdio.h>  
#include <string.h>

int main()  
{  
  /\* cadenas de caracteres a usar \*/  
  char apellido[30];  
  char nombre[30];  
  char resultado[80];

  printf("Ingrese Nombre: ");  
  scanf("%s", nombre);  
  printf("Ingrese Apellido: ");  
  scanf("%s", apellido);

  strcpy(resultado, apellido); /\* copia el apellido en resultado \*/  
  strcat(resultado, ", "); /\* concatena al resultado, la coma y el espacio \*/  
  strcat(resultado, nombre); /\* concatena al resultado, el nombre \*/

  printf("Resultado: %s\n", resultado);  
}

Ejercicio 2: Realizar una función que reciba por parámetro una cadena de caracteres. Se deberá crear otra cadena con el mismo contenido pero transformando todas las letras a mayúscula. Mostrar el resultado de conversión por pantalla.

Ejercicio 3: Realizar un programa que permita ingresar una frase de hasta 80 caracteres. Mostrar por pantalla la cantidad de letras, digitos y espacios que contiene la frase ingresada, para las letras tener en cuenta tanto las minúsculas como las mayúsculas.

Solución:

#include <stdio.h>  
#include <string.h>

int main()  
{  
  char frase[80]; /\* cadena para ingresar la frase \*/  
  int letras=0; /\* variables para los contadores \*/  
  int espacios=0;  
  int digitos=0;  
  int i=0; /\* indice para recorrer la cadena \*/

  printf("Ingrese una frase (max 80 caracteres): ");  
  gets(frase); /\* no se usa scanf() porque no permite espacios \*/

  while(i < strlen(frase)) /\* i: recorrer desde cero hasta la cantidad de caracteres ingresados \*/  
  {  
    if(frase[i] == ' ' ) /\* para contar espacios \*/  
    {  
       espacios++;  
    }  
    switch( frase[i] ) /\* para contar los dígitos \*/  
    {  
      case '0':  
      case '1':  
      case '2':  
      case '3':  
      case '4':  
      case '5':  
      case '6':  
      case '7':  
      case '8':  
      case '9':  
          digitos++;  
          break;  
    }  
    /\* para contar las letras, se compara por los rangos de mayúsculas y minúsculas \*/  
    if( (frase[i] >= 'A' && frase[i] <= 'Z') || (frase[i] >= 'a' && frase[i] <= 'z') )  
    {  
         letras++;  
    }  
    i++; /\* incrementar el indice para seguir recorriendo la cadena \*/  
  }  
  /\* mostrar los resultados \*/  
  printf("Letras: %d\n", letras);  
  printf("Dígitos: %d\n", digitos);  
  printf("Espacios: %d\n", espacios);  
}

Ejercicio 4: Realizar un programa que solicite ingresar tres nombres en cadenas de 30 caracteres, convertir el contenido de cada nombre a letras minúsculas y luego mostrarlos por pantalla ordenados alfabeticamente de menor a mayor.

Ejercicio 5: Realizar una función que reciba por parámetro una letra (char) y un número entero largo (long int), deberá crear una cadena de caracteres que contenga la letra y el número con este formato:

"FACTURA: X-00001234", donde "X" es la letra recibida por parámetro y 1234 es el número recibido por parámetro (como ejemplo). Debe incluir la palabra "FACTURA" y el número se debe completar con ceros a la izquierda para totalizar siempre 8 dígitos.  Mostrar el resultado por pantalla.

Ejercicio 6: Realizar un programa que permita controlar el formato correcto de una patente de automóvil. El usuario igresará en una cadena de texto el código de la patente: "XX999XX", en total 7 caracteres, el programa debe controlar que los 2 primeros caracteres sean letras, los 3 siguientes sean dígitos y que los 2 últimos también sean letras, considerar solamente las mayúsculas. El programa debe informar por pantalla si el valor ingresado es válido o no.

Ejercicios de la Unidad 2: Vectores

Ejercicio 1: Realizar un programa que genere un vector "A", de 10 elementos enteros cargado con los números impares comenzando por el 1. Crear otro vector "B", de 5 elementos donde el valor de cada uno de esos elementos se obtiene de la suma de dos elementos consecutivos del primer vector, por ejemplo, el valor del primer elemento de "B" será igual a la suma del primer y segundo elemento de "A", el segundo elemento de "B" será la suma del tercer y cuarto elemento de "A", y así hasta competar todos los elementos. Mostrar por pantalla el contenido de ambos vectores.

Ejercicio 2: Realizar un programa que genere un vector de de tipo entero de 20 elementos cargado con números aleatorios entre 1 y 100. Mostrar el contenido de vector por pantalla. Calcular y mostrar el valor promedio de todos sus elementos. Crear un segundo vector colocando en cada posición el valor 1 si el valor original es menor al promedio calculado o el valor 100 si el valor original es mayor al promedio calculado, el vector resultante tendrá entonces solamente valores 1 y 100. Mostrar el resultado por pantalla.

Solución:

#include <stdio.h>  
#include <time.h>  
#include <stdlib.h>

int main()  
{

     int vec1[20];  
     int vec2[20];  
     int i;  
     float promedio;

     srand(time(NULL));  
     /\* cargar el vector \*/  
     for(i=0; i<20; i++)  
     {  
         vec1[i] = 1 + rand() % 100;  
     }  
     /\* mostrar \*/  
     for(i=0; i<20; i++)  
     {  
         printf("%d ", vec1[i]);  
     }  
     printf("\n");  
     /\* calcular promedio \*/  
     promedio = 0;  
     for(i=0; i<20; i++)  
     {  
          promedio += vec1[i];  
     }  
     promedio = promedio / 20;  
     printf("Promedio: %.2f\n", promedio);  
     /\* cargar el vector 2 \*/  
     for(i=0; i<20; i++)  
     {  
         if(vec1[i] > promedio)  
         {  
              vec2[i] = 100;  
         }  
         if(vec1[i] < promedio)  
         {  
              vec2[i] = 1;  
         }  
     }  
     /\* mostrar vector 2 \*/  
     for(i=0; i<20; i++)  
     {  
         printf("%d ", vec2[i]);  
     }

}

Ejercicio 3: Realizar una función que reciba por parámetro dos números enteros, A y B, la función deberá controlar que A sea menor que B, si eso no ocurre no debe realizar  ninguna acción. Si la condición es correcta deberá crear un vector de números enteros con la cantidad de elementos que resulte de B-A+1 y colocar en sus elementos la serie de valores comenzando en el valor de B y terminando en el valor de A, es decir en forma decreciente. Mostar por pantalla el vector resultante.

El valor de A debe ser mayor o gual a cero y el de B debe ser menor o igual a cien.

Ejercicio  4: Realizar un programa que declare un vector de tipo entero de 20 elementos y asigne el valro cro a todos sus elementos. A continuación deberá cargar las 10 primeras posiciones del vector con números aleatorios entre 1 y 100 y mostrar por pantalla el contenido completo del vector, posteriormente deberá completar las 10 posiciones restantes colocando el doble de cada valor cargado en las primeras 10 posiciones, Ejemplo: si en la posición 0 se cargó el valor 35, en la posicón 10 se deberá guardar el doble de 35, es decir 70. así hasta completar la carga del vector. Volver a mostrar por pantalla el contenido del vector.

Ejercicios de la Unidad 2: Matrices

Ejercicio 1: Realizar un programa que declare una matriz entera de 4 filas y 5 columnas y que proceda a cargarla de la siguiente manera: el primer elemento de cada fila será un número aleatorio entre 1 y 10, los siguientes valores de la fila serán el doble del anterior, ejemplo: si el primer valor es 3, los siguientes valores de la fila serán 6, 12, 24, 48, hasta completar los 5 elementos de cada fila.

Mostrar los elementos de la matriz por pantalla. Determinar si alguna fila comienza con un número impar y mostrar el número de fila correspondiente.

Ejercicio 2: Realizar un programa que genere una matriz cuadrada de 4 x 4 de tipo entera y que proceda a cargarla con números aleatorios entre 1 y 1000. A continuación deberá calcular y mostrar las sumas de la diagonal principal y secundaria de la matriz.

Solución:

#include <stdio.h>  
#include <time.h>  
#include <stdlib.h>

int main()  
{  
    int mat[4][4];  
    int f;  
    int c;  
    int suma1=0;  
    int suma2=0;  
    /\* A X X B  
        X A B X  
        X B A X  
        B X X A \*/

    srand(time(NULL));

    /\* cargar la matriz \*/  
    for(f=0; f<4; f++)  
    {  
        for(c=0; c<4; c++)  
        {  
             mat[f][c] = 1 + rand() % 1000;  
             printf("%d ", mat[f][c]);  
        }  
        printf("\n");  
    }

    /\* suma diagonal principal usando 2 ciclos repetitivos \*/  
    for(f=0; f<4; f++)  
    {  
        for(c=0; c<4; c++)  
        {  
            if(f==c){  
                suma1 = suma1 + mat[f][c];  
            }  
        }  
    }  
    printf("Diagonal principal: %d\n", suma1);

    /\* suma diagonal principal usando 1 ciclo repetitivo \*/  
    suma1=0;  
    for(f=0; f<4; f++)  
    {  
        suma1 = suma1 + mat[f][f];  
    }  
    printf("Diagonal principal: %d\n", suma1);

    /\* suma diagonal secundaria \*/  
    suma2=0;  
    for(f=0; f<4; f++)  
    {  
        suma2 = suma2 + mat[f][3-f];  
    }  
    printf("Diagonal secundaria: %d\n", suma2);

    /\* suma diagonal secundaria con un ciclo for y 2 variables de control \*/  
    suma2=0;  
    for(f=0, c=3; f<4; f++, c--)  
    {  
        suma2 = suma2 + mat[f][c];  
    }  
    printf("Diagonal secundaria: %d\n", suma2);

}

Ejercicio 3: Realizar un programa que permita al usuario ingresar valores de tipo float para cargar una matriz de 2 filas y 3 columnas, los valores representan la cantidad de kilos de papel que se cargan en una máquina de reciclado por turno de trabajo, hay 2 máquinas (2 filas) y 3 trunos (3 columnas). Una vez ingresados todos los valores en la matriz el programa deberá determinar qué máquina procesó la mayor cantidad de papel y qué turno procesó la menor cantidad de papel, Mostrar ambos resultados por pantalla.

Ejercicio 4: Realizar un programa que defina una matriz de enteros de 5 filas y 5 columnas, cargar la matriz completa en forma aleatoria con valores 0 y 1 unicamente. A continuación el programa deberá determinar qué fila tiene la mayor cantidad de unos (1) y qué columna tiene la mayor cantidad de ceros (0), mostrar la matriz y los resultados por pantalla.

**no hay intervenciones**

Colegio Universitario IES