
Estructuras de Datos y Algoritmos 1

Práctico 1

Tema: Introducción al Lenguaje C/C++

Escucho y olvido, veo y recuerdo, hago y aprendo.

Confucio

Objetivos

- Presentar el lenguaje C y C++.
- Introducir conceptos de dirección de memoria, pasaje de parámetros en C y C++.
- Presentar procesos secuenciales de datos.
- Repasar estructuras de control y arreglos en C/C++.
- Presentar el IDE de C++ a usar en el curso y una metodología de resolución de problemas que hace uso tempranamente de las facilidades de la herramienta.
- Introducir los conceptos de programación estructurada, pre y pos condiciones, casos de prueba y estudio empírico de la velocidad de crecimiento del tiempo de corrida

Los ejercicios resaltados deben ser implementados en máquina. Para estos ejercicios las consultas serán evacuadas sobre la base del trabajo realizado por el estudiante. Se harán sugerencias y comentarios a partir del código y casos de prueba desarrollados por cada estudiante.

Para los ejercicios marcados con * debe realizarse un estudio empírico del mejor caso y peor caso y, cuando sea posible, investigar el comportamiento promedio

Ejercicios

Básicos

Lecturas Recomendadas

Deitel & Deitel Como programar en C y C++, Capítulos 1- 4

<http://www.cplusplus.com/doc/tutorial/index.html>, Basics of C++

- 1) Realizar la guía 1.
- 2) Diseñar una rutina que, dadas dos variables, numerador y denominador simplifique la fracción numerador / denominador (sugerencia: usar el algoritmo de Euclides). Por ejemplo, $30/12 = 5/2$
- 3) Escriba un algoritmo que lea tres valores reales positivos y determine si pueden ser las longitudes de los lados de un triángulo (considerar que la suma, al menos, de los cuadrados de dos de sus lados tiene que ser mayor igual al cuadrado del otro lado).

4) Escriba un algoritmo que lea tres valores reales positivos y determine si pueden ser las longitudes de los lados de un triángulo rectángulo (considerar el Teorema de Pitágoras).

5)

a) Leer n números (n será ingresado inicialmente por pantalla) e indicar su promedio, la cantidad de números pares, cantidad de múltiplos de 3.

b) Ídem pero para una serie de números terminada por 0. ¿Qué hubo que cambiar?

6) Procesar una serie de n números e indicar la cantidad de secuencias estrictamente ascendentes que contiene. Por ejemplo, la serie 1, 2, 3, 1, 1, 4, 8, 8, 9 tiene como secuencias estrictamente ascendentes a: {123, 148, 89}, por lo tanto el resultado es: 3.

7) Leer una serie de números (terminada por 0) que representan sueldos de los funcionarios de una empresa y calcular qué porcentaje de los mismos se encuentran por sobre un determinado valor, cuál es el promedio de sus sueldos y de la empresa en general (este valor es ingresado por pantalla).

Por ejemplo, dados los sueldos: 1000, 1500, 2000, 4000, 1500, 0 y valor = 1800

(1) Porcentaje $> 1800 = 40\%$

(2) Promedio de sueldos $> 1800 = 3000$

(3) Promedio de la empresa = 2000

8) Pedir un número n , luego una serie de n números más, e imprimir el mayor de la misma, así como la suma de todos los números desde el principio hasta dicho máximo inclusive.

Por ejemplo, dado $n = 7$ y la serie 1, 12, 2, 4, 50, 3, 40

Máximo = 50

Suma Hasta el Máximo = 69

9)

a) Dada una serie de números formada únicamente por los números 1, 2 y 3 y terminada por un 0, contar la cantidad de veces que aparece la subsecuencia 1-2-3.

b) Modificar el algoritmo de anterior para que considere como válidas sub-secuencias como 1-1-2-2-2-2-3, 1-2-2-3, etc., donde pueden aparecer varios números 1 juntos, varios 2 y varios 3.

Aritmética de Direcciones

10) Hacer una función que intercambie el contenido de dos variables enteras recibidas como parámetro.

11) Indique que hace la siguiente función:

```
int misterio (char *p) {  
    char *t = p;  
    while (*t != '\0' )  
        t++;  
    return t - p;  
}
```

12) Indique que hace la siguiente función:

```
void misterio (char *p, char *t){  
    while (*p++ = *t++);  
}
```

13) Indique que hace la siguiente función:

```
int misterio (char *s, char *t){  
    for (; *s == *t; s++, t++)  
        if (*s == '\0')  
            return 0;  
    return *s - *t;  
}
```

14) Indique que hace la siguiente función:

```
int misterio (char *s, char *t){  
    int i;  
  
    for (i = 0; s[i] == t[i]; i++)  
        if (s[i] == '\0')  
            return 0;  
    return s[i] - t[i];  
}
```

15) Compare las funciones de los dos ejercicios anteriores.

Arreglos y matrices

Lecturas Previas

Deitel & Deitel Como programar en C y C++, Capítulos 5-7

<http://www.cplusplus.com/doc/tutorial/index.html>

- Control structures
- Advanced Data

- 16)
- Escriba una función promedio que reciba un vector, dos enteros izq y der ($izq \leq der$) y retorne el promedio de los elementos comprendidos entre las posiciones izq y der (inclusive).
 - Dado un vector de N elementos, hallar su mínimo, la cantidad de veces que se repite y el lugar (posición) donde aparece por primera vez.
- 17)
- Escriba una función `int posicionDelMaximoDelVector (vector v)` que devuelve la posición del máximo elemento encontrado dentro del vector v.
 - Escriba una rutina que reciba un vector y lo ordene.
- 18)
- Escribir una función que reciba un vector origen y otro destino, una posición orig y otra des y un largo l y copie desde la posición orig del vector origen l elementos a la posición des del vector destino.
 - Diseñar una rutina que dado un array de int desordenado lo ordene en forma ascendente o descendente según el valor del parámetro orden.
- 19) Dados dos vectores ordenados v1 y v2 de dimensiones n1 y n2 respectivamente escribir una función que genere un tercer vector v3 que contenga los elementos de los dos anteriores intercalados manteniendo el orden.
- 20) Escriba una función que reciba una matriz bidimensional de reales, una posición i,j y retorne un boolean respondiendo si esta posición es un punto de silla de la matriz (es a la vez el mínimo de su fila y el máximo de su columna o viceversa).
- 21) Ídem anterior pero considerando también las diagonales.
- 22) Escriba una función que reciba una matriz bidimensional de reales, y retorne un int que corresponde a la cantidad de puntos de silla de la matriz.

EJERCICIOS ADICIONALES PROPUESTOS

- 23) Diseñar una rutina que dados dos números a real y b entero, calcule la potencia a^b . La rutina debe funcionar correctamente para $b > 0$, $b = 0$ y $b < 0$.
- 24) Ternas pitagóricas. Un triángulo rectángulo puede tener lados cuyas longitudes sean **enteras**. Por ejemplo 3, 4, 5. Encuentre todas las ternas pitagóricas con lados no mayores de 500 unidades.
- 25) Un palíndromo es un número, frase, o texto que se lee igual hacia adelante y hacia atrás por ejemplo c/u de los siguientes enteros de 5 dígitos son palíndromos: 12321, 55555, 45554 y 11611. Escriba un algoritmo que lea un entero de 5 dígitos y determine si es un palíndromo. (sugerencia: use los operadores de división y módulo para separar los dígitos).
- 26) Escribir una rutina que dado un vector y una clave responda si clave se encuentra o no en el vector, devolviendo, en caso afirmativo la posición de su primera ocurrencia y en caso negativo -1.
- 27) ¿Cómo mejoraría la rutina anterior si supiera que el vector está ordenado?

28) Diseñar una rutina que dados un vector y un elemento del mismo (pivote) lo reorganice de manera que queden primero todos los elementos menores que límite, el propio elemento límite y luego los elementos mayores que límite.

29) Ordenación por comparación:

- a) Escriba una función que recibiendo un vector de enteros con capacidad para n elementos y cuyos i primeros elementos se encuentran ordenados inserte un elemento x de forma que se mantenga la condición de orden pero ahora con i+1 elementos.
- b) Implemente el Insert Sort.
- c) Implemente el Shell Sort.

30) La clasificación por conteo se basa en, para cada elemento, contar cuantos menores o iguales que él se encuentran presentes en la entrada. Implemente un algoritmo de clasificación basado en esta técnica.

31) Escriba una función mediana que reciba un vector y retorne la mediana de los elementos. La mediana de un conjunto de observaciones se define como el valor que ocupa el lugar central una vez ordenadas las mismas o el promedio de las dos centrales cuando el número de observaciones es par. Trate de resolver el problema reutilizando funciones ya construidas.

EJERCICIOS de Cadenas de Caracteres

- 1) Escriba una función **unsigned int *splitByChar(char *str, char *separadores);** que recibiendo un string str compuesto por cero o más números enteros positivos (de uno o más dígitos) separados por los caracteres contenidos en el string separadores, retorne un vector de enteros con las siguientes propiedades:
- En la primera posición debe aparecer la cantidad de números que se encuentran en el string str separados por los caracteres separadores.
 - En las siguientes posiciones deben aparecer los números en el orden en el que aparecen en el string str.
 - El vector no debe tener posiciones sin utilizar. Se debe reservar la memoria justa.
 - Se debe reservar la memoria necesaria utilizando la palabra reservada new.

El vector debe tener tantas posiciones como números se encuentran en el string más uno. El vector resultante no puede ser vacío, por lo menos siempre tiene una posición que contiene la cantidad de posiciones que vienen después. El string str puede estar vacío, en cambio el string separadores no.

NOTAS:

- No se pueden utilizar funciones incluidas en bibliotecas del sistema.
- Los string vienen bien formados y los separadores no van a ser números.
- Esta función no debe afectar de ninguna forma a los string.

Ejemplos:

splitByChar("12:1999;0:24", ";;:&")

RESULTADO:

4	12	1999	0	24
---	----	------	---	----

`splitByChar("5", ",")`

RESULTADO:

1	5
---	---

`splitByChar("", "%&=")`

RESULTADO:

0

2) Escriba una función **char *combine(unsigned int *vec, char separador);**

que recibiendo un vector vec compuesto por uno o más números enteros positivos (de uno o más dígitos), que en la primera posición contiene el número de posiciones que le siguen, retorne un string con las siguientes propiedades:

- Si la posición cero del vector es igual a cero, se debe retornar un nuevo string vacío.
- En caso contrario hay que retornar un string que contenga todos los números (excepto la posición cero) separados por el carácter separador.
- El string nunca debe comenzar ni terminar con el separador.
- Se debe reservar la memoria necesaria utilizando la palabra reservada new.

NOTAS:

- No se pueden utilizar funciones incluidas en bibliotecas del sistema.
- El vector viene bien formado.
- Esta función no debe afectar de ninguna forma al vector.

Ejemplos:

`combine(vec,`

4	12	1999	0	24
---	----	------	---	----

`, ";")`

retorno: "12;1999;0;24"

`combine(vec`

0

`, ";")`

retorno: ""

Cuando hablamos de vector estamos diciendo: Un puntero al elemento en la posición cero del vector.

Cuando hablamos de strings estamos diciendo: Un puntero a una serie de caracteres consecutivos en memoria terminados por el carácter '\0'. El tipo de datos que se usa es char*.