

Algoritmos y Programación I - 75.02
Guía de Ejercicios N° 1
Programas lineales - Introducción / Generales

Nota: Los siguientes ejercicios introductorios se refieren a programas ANSI-C no modularizados (autocontenidos dentro de la función main()).

- 1) Un negocio de deportes hace un 25 % de descuento en pelotas de fútbol y de béisbol. Se requiere escribir un programa que le permita a un empleado ingresar los precios originales de las pelotas. El programa deberá usar esta entrada de datos para calcular el precio rebajado. La salida del programa deberá mostrar, en líneas separadas, el precio original y el precio con descuento.
- 2) Suponga que debe escribir un programa para calcular la resistencia total de un circuito en serie. En tal circuito, la resistencia total es la suma de todos los valores individuales de las resistencias. Suponer que el circuito consiste en una cantidad de 2 resistores de 56 ohmios, 4 de 33 ohmios y 1 de 15 ohmios. Responda las siguientes preguntas:
 - a. Cuántas salidas requiere este problema de programación?
 - b. Cuántos datos de entrada tiene el problema?
 - c. Escriba un programa que lea de teclado el conjunto de valores de las resistencias (para cualquier circuito serie), y calcule la resistencia total del circuito, mostrando el resultado con 2 decimales por pantalla.
 - d. Pruebe el algoritmo escrito para la parte c), usando la siguiente muestra de datos: 47K, 12K, 680R y , 2M2.

Nota: En los cuerpos de los resistores es común expresar la unidad Ohm, "Ω", con la letra "R", por una cuestión de tipografía. Por lo tanto, 680R es equivalente a 680Ω.

- 3) La fórmula de la desviación estándar normal z, usada en aplicaciones de estadística es $z = (X-m) / s$, donde m es el valor medio y s es la desviación estándar. Usando esta fórmula, escriba un programa que calcule y despliegue el valor de la desviación estándar normal cuando $X = 85.3$, $m = 80$ y $s = 4$.
- 4) La ecuación de una curva normal utilizada en aplicaciones estadísticas es:

$$y = \frac{1}{s \times \sqrt{2\pi}} \times e^{-\frac{1}{2} \left[\frac{(x-m)}{s} \right]^2}$$

¿Cuántos datos de entrada tiene este problema?

¿Cuántas salidas se requieren en este problema?

Determinar un algoritmo para convertir los datos entrada en datos de salida.

Pruebe el algoritmo para $m = 90$, $s = 4$, $x = 80$, $\pi = 3.1416$.

- 5) Escribir un programa que calcule la raíz cuadrada y el inverso de un número ingresado por el teclado. Antes de calcular la raíz cuadrada se debe verificar que el número sea no negativo y antes de calcular el inverso, se debe verificar que el número no sea cero.
- 6) Escribir un programa que calcule las soluciones de una ecuación cuadrática con coeficientes reales.
- 7) Usando una instrucción do, escribir un programa para validar y aceptar una calificación. El programa debe aceptar una calificación continuamente mientras no se introduzcan calificaciones inválidas. Una calificación inválida es cualquiera menor que 0 o mayor que 100. Si la calificación es inválida, el programa debe imprimir un mensaje, de otra forma, la calificación debe sumarse al total de calificaciones ingresadas. Por último, debe desplegar el total.
- 8) Escribir un programa que permita calcular en forma iterativa el factorial de un número ingresado por teclado. Finalizado el ingreso de datos, mostrar el resultado por stdout. Realizar todas las validaciones que considere necesarias.
- 9) Escribir un programa que lea un número entero positivo e imprima por stdout los números naturales impares menores a él.

- 10) Idem ejercicio anterior pero para los números pares.
- 11) Escribir un programa que invierta los dígitos de un número positivo entero. (Hint: usar operadores módulo, %, y división, /, para ir obteniendo los dígitos uno a uno).
- 12) Escribir un programa que permita determinar el máximo y el mínimo de un conjunto de valores ingresados por teclado (sin almacenar la totalidad de los valores leídos). A tal efecto debe ingresarse primero la cantidad esperada de elementos a procesar, y luego el lote de datos de a uno por vez. Finalizado el ingreso de datos, mostrar el resultado por stdout con 3 decimales. Realizar todas las validaciones que considere necesarias.
- 13) Escribir un programa que permita calcular el promedio aritmético de una serie de valores ingresados por teclado (sin almacenar la totalidad de los valores leídos). A tal efecto debe ingresarse primero la cantidad esperada de elementos a promediar, y luego el lote de datos de a uno por vez. Finalizado el ingreso de datos, mostrar el resultado por stdout con 3 decimales. Realizar todas las validaciones que considere necesarias.
- 14) El valor aproximado del número de Euler, e , se puede obtener con la siguiente fórmula:

$$e = 1 + 1/1! + 1/2! + 1/3! + 1/4! + 1/5! + \dots$$

Escribir un programa que calcule el valor aproximado de e mediante un ciclo repetitivo que termine cuando la diferencia entre dos aproximaciones sucesivas difiera en menos de $10 \cdot e^{-9}$. Parametrizar adecuadamente el software desarrollado.

- 15) Suponga que se realizaron cuatro experimentos, cada uno de los cuales tiene seis resultados de ensayo. Los resultados de cada experimento se muestran abajo. Escribir un programa que use `while` anidados para calcular y desplegar el promedio de los resultados del ensayo para cada experimento.

Resultados del primer experimento: 23.2, 31, 16.9, 27, 25.4, 28.6

Resultados del segundo experimento: 34.8, 45.2, 27.9, 36.8, 33.4, 39.4

Resultados del tercer experimento: 19.2, 16.8, 10.2, 20.9, 25.4, 18.6

Resultados del cuarto experimento: 36.9, 39, 49.3, 45.1, 52.4, 21.7

- 16) Un profesor usa un programa para calcular la nota de sus estudiantes basándose en la calificación obtenida en dos exámenes y en la calificación de cinco programas. El profesor le ha pedido que modifique el programa para que se pueda ingresar la calificación de todos los estudiantes. El nuevo programa debe contar la cantidad de 1s, 2s, 3s, 4s, 5s, 6s, 7s, 8s, 9s y 10s que han obtenido sus estudiantes. El profesor tiene 26 estudiantes en su clase de C++ este curso. En la solución de este problema se deberá utilizar un ciclo `while`.
- 17) Escribir un programa que pruebe la efectividad de la función de biblioteca `rand()`. Comience por inicializar 10 contadores, como `cuentacero`, `cuentauno`, `cuentados`, ..., hasta `cuantanueve` a cero. Luego genere una gran cantidad de números pseudoaleatorios entre 0 y 9. cada vez que ocurra un 0 se aumenta `cuentacero`, y así con todos los dígitos decimales. Por último imprima el número de ceros, unos, etc. que ocurrieron y el porcentaje de cada ocurrencia.

Nota: Las funciones de la biblioteca "conio.h" no son estándar, por lo que prohíbe su utilización dentro del Curso.