

Guía de Ejercicios 2 - Estructuras de Iteración

Advertencia

La resolución conjunta o grupal de los ejercicios aquí presentes no está permitida, excepto en la medida en que puedas pedir ayuda a tus compañeros de clase y a otras personas, y siempre que esa ayuda no se reduzca a que otro haga el trabajo por vos.

El código fuente entregado por un estudiante debe ser escrito en su totalidad por dicha persona.

Ejercicio 2.1

Escribir un programa que imprima los números enteros **pares** entre el 0 y el 100 por pantalla.

Nota: Recuerde que el 0 es considerado par.

Ejercicio 2.2

Escribir un programa que realice la sumatoria de los números del 1 al 100 y luego imprima el resultado por pantalla.

Ejercicio 2.3

Escribir un programa que calcule el promedio de 10 valores numéricos reales ingresados por teclado y lo imprima por pantalla.

Ejercicio 2.4

Escribir un programa que determine si un número entero ingresado por teclado es primo y lo informe por pantalla.

Ejercicio 2.5

Escribir un programa que realice la sumatoria del rango de números enteros entre dos valores ingresados por teclado.

Ejercicio 2.6

Escribir un programa que imprima los números enteros del 1 al 100 por pantalla. Pero para los números múltiplos de 3, debe imprimir la palabra **fizz** en vez del número, para los números múltiplos de 5, debe imprimir la palabra **buzz** en vez del número, y para aquellos números que son múltiplos tanto de 3 como de 5, debe imprimir la palabra **fizzbuzz** en vez del número.

Ayuda: Para los números del 1 al 15, se debería obtener la siguiente salida:

```
1
2
fizz
4
buzz
fizz
7
8
fizz
buzz
11
fizz
13
14
fizzbuzz
```

Ejercicio 2.7

Escribir un programa que reciba un número entero del 1 al 10 y luego imprima la tabla de multiplicar del mismo.

Ayuda: Para el número 2, se debería obtener la siguiente salida:

```
2, 4, 6, 8, 10, 12, 14, 16, 18, 20
```

Ejercicio 2.8

Escribir un programa donde se ingresa una serie de números reales. Cuando la suma de los números ingresados alcance un valor mayor que 300, se deberá finalizar el ingreso y se informará cuántos elementos tiene la serie.

Nota: Se deben admitir tanto números enteros como reales que, a su vez, podrán ser positivos o negativos.

Ejercicio 2.9

Escribir un programa donde se ingresa una serie de números enteros y luego se informa la cantidad de valores pares e impares ingresados. El programa finaliza cuando se ingresa 0.

Nota: El cero es la condición de finalización con lo cual no debe ser tomado en cuenta para el informe final del programa.

Ejercicio 2.10

Escribir un programa que acumule números enteros ingresados por teclado hasta que dicha acumulación supere el valor 100 o el valor ingresado sea 0. Al finalizar el programa, se debe informar por pantalla el número acumulado y el promedio de los valores ingresados.

Nota: El cero es la condición de finalización con lo cual no debe ser tomado en cuenta para el cálculo del promedio.

Ejercicio 2.11

Escribir un programa que reciba dos numeros enteros positivos. Si el primer número es par deberá calcular el factorial del mismo pero si es impar deberá calcula la potencia del primero elevado por el segundo.

Ejercicio 2.12

Escribir un programa que imprima por pantalla todas las fichas del juego dominó, sin repetir ninguna.

Ejercicio 2.13

Escribir un programa que reciba un número binario de 4 bits y luego imprima su representación en base 10. Resolver aplicando una estructura de iteración.

Nota: El primer bit ingresado debe considerarse como el bit más significativo (MSB).

Ayuda:

1°	2°	3°	4°	Salida por stdout
1	0	0	0	El número ingresado es 8
1	0	1	1	El número ingresado es 11

Ejercicio 2.14

Un tirador realiza 10 disparos a un blanco cuyo centro coincide con el origen de coordenadas. Leyendo la abscisa y la ordenada de cada impacto por consola, deberá calcularse el puntaje final obtenido. El blanco tiene cuatro zonas que, de acuerdo a la distancia del impacto al origen (R), son:

Zona	Denominación	Puntaje
0 <= R <= 1	centro	10
1 < R <= 5	medio	5
5 < R <= 10	externo	1
10 < R	falla	0

Ejercicio 2.15

Escribir un programa que reciba por teclado pares de valores no nulos X e Y, que representan las coordenadas rectangulares de distintos puntos en el plano. El fin de datos se indica con X e Y iguales a cero.

Se pide determinar e informar por pantalla:

- a) Cantidad de puntos que pertenecen a cada cuadrante.
- b) Sumatoria de las distancias al origen de los puntos pertenecientes al primer cuadrante.

Ejercicio 2.16

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 elevado a la -9.

Ejercicio 2.17

Escribir un programa que obtenga los factores positivos de un número positivo.

Ayuda: Los factores pueden dividir exactamente a un número sin un residuo o decimal. Por ejemplo, 30 dividido entre 10 es 3, y 30 dividido entre 15 es 2, así que 2, 3, 10 y 15 son todos factores de 30.

Ejercicio 2.18

Escribir un programa que simule 5 tiradas de 4 dados y calcule el puntaje obtenido. En caso de realizar **generalala** (4 dados iguales) el programa deberá interrumpir las tiradas para imprimir por pantalla una felicitación al jugador y luego terminar. De no obtener generala, se deberá imprimir el puntaje final del jugador al finalizar las tiradas.

Las reglas de esta versión modificada de **La Generala** no son exactamente iguales a las del juego original, ya que se juega con 4 dados y los puntajes se calculan de la siguiente manera:

Tirada	Denominación	Puntaje
4 dados iguales	Generala	50
3 dados iguales	Póker	40
2 dados iguales	Full	30
Ningún dado igual	Agua	0

Ayuda: Se recomienda utilizar la función estándar **rand** para la generación de números aleatorios.

Ejercicio 2.19

Escribir un programa que tome un número entero en base 10 (decimal) y lo muestre en base 2 (binario) por pantalla.

Nota: El número obtenido puede ser mostrado de manera inversa (bit menos significativo a la izquierda).

Ejercicio 2.20

Escribir un programa que tome un número entero en base 10 (decimal) y lo muestre en base 16 (hexadecimal).

Nota: Realizar el ejercicio sin usar **%x**. El número obtenido puede ser mostrado de manera inversa (bit menos significativo a la izquierda).

Ejercicio 2.21

Se tienen 5 sensores de temperatura, que registran 15 temperaturas distintas cada uno. Esas temperaturas se ingresan de a una por vez, primero las 15 del primer sensor, luego las 15 del segundo, y así sucesivamente.

Terminado el ingreso de datos, se deberá mostrar por pantalla:

- a) Temperatura promedio detectada por cada sensor.
- b) Máxima temperatura registrada por cada sensor.
- c) Número de sensor que registró la temperatura máxima.

Ejercicio 2.22

Escribir un programa que permita determinar el máximo y el mínimo de un conjunto de valores numéricos reales ingresados por stdin (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.

Ejercicio 2.23

Escribir un programa que pida el número de legajo (entero de 8 cifras) y la calificación de un examen de los estudiantes de un curso. El fin de ingreso de datos se indica con el ingreso de un legajo igual a cero.

El programa deberá informar:

- a) Cantidad de estudiantes del curso.
- b) Promedio de las calificaciones.
- c) Cantidad de estudiantes aprobados (se aprueba con calificación igual o mayor que 6).
- d) Mayor calificación.
- e) La cantidad de estudiantes por cada calificación conceptual, según la siguiente tabla:

Calificación conceptual	Calificación numérica
Sobresaliente	10
Muy bueno	8 ó 9
Bueno	6 ó 7
Regular	4 ó 5
Insuficiente	1 a 3

Ejercicio 2.24

Cerca del final del nivel 1-1 en Super Mario Bros de Nintendo, Mario debe saltar sobre pirámides de bloques adyacentes, como se muestra a continuación.



Codifiquemos dichas pirámides en C, aunque en texto, usando numerales (#) para los ladrillos, como se muestra a continuación (cada numeral es un poco más alto que ancho, por lo que las pirámides en sí también son más altas que anchas):

```
#
##
###
####
```

Dejaremos que el usuario decida qué tan altas deben ser las pirámides solicitando primero un número entero positivo entre 1 y 8, inclusive.

Nota: Si el usuario no ingresa un número entero positivo dentro del rango indicado, el programa deberá volver a solicitar un número al usuario:

```
$ ./mario
Ingrese el alto: -1
Ingrese el alto: 0
Ingrese el alto: 42
Ingrese el alto: 50
Ingrese el alto: 5
#
##
###
####
#####
```

Referencias

Algunos ejercicios fueron obtenidos y adaptados de:

- Guía de Trabajos Prácticos 2011 - Informática I - Departamento de Electrónica - UTN FRBA
- Guía de Ejercicios - Algoritmos y Programación I - UBA FIUBA
- Set de Problemas - CS50 - Harvard University