

Estructuras de control

Unidad 3

Guía de ejercicios Nº3

Pensamiento computacional (90)
Cátedra: Balbiano

Los ejercicios que encontrarán a continuación los harán ejercitar los temas de expresiones, estructuras de control iterativas y estructuras de control condicionales. Se busca que se puedan aplicar los conocimientos vistos en la sesión, específicamente el uso de while y for, if y expresiones.

Expresiones

Para esta sección de los ejercicios, revisar los temas de *operadores lógicos* y de *comparación*.

1. Escribir la expresión para saber si un número es más grande que otro. Guardarla en una variable de tipo bool e imprimirla por pantalla para ver su valor.
2. Repetir el punto anterior pero con la expresión que determina que una letra NO es vocal.
3. Repetir pero para la expresión que permite saber si un número es par y menor a 10.

Estructuras de control condicionales

4. Crear una función que dado un número, devuelva su valor absoluto. El valor absoluto de un número es el mismo número sin considerar el signo.
5. Crear el programa al que sea imposible ganarle en el juego de “Piedra, papel o tijera”. Cada elemento va a ser representado con una letra: **R** para piedra, **P** para papel y **T** para tijera.
 - a. Hacer una función que le haga al usuario ingresar alguna de esas letras, e imprima por pantalla la jugada para ganarle. Por ejemplo:

```
> ¡Juguemos! Ingresá piedra ( R), papel (P) o tijera (T)
> P
> Tijera. ¡Te gané!
```

ATENCIÓN: Observar cómo se usa una frase inicial para darle a entender al usuario lo que tiene que hacer (en este caso ingresar alguna de las tres letras).
 - b. Mostrar por pantalla el mensaje “NO vale” cuando el usuario ingresa una letra no válida (distinta de **R**, **P** o **T**).
6. Escribir código que dado dos enteros, determine si la suma de ambos da menos que 100. Si la suma de ambos es menor a 100, calcular cuánto falta para llegar a 100 y mostrar por pantalla un mensaje con ese valor. Si la suma es mayor a 100, mostrar un mensaje diciendo “Llega a 100”.
Extra: ¿Cómo harían para que el programa quede generalizado para cualquier límite, a elección del usuario, y no solo para 100?.
7. Se tienen letras para representar las estaciones del año:
 - V para verano
 - O para otoño
 - I para invierno

- P para primavera

Crear una función que dada una letra, imprima por pantalla la estación del año que representa (es decir, si se ingresa **V** se mostrará por pantalla el mensaje “Verano”). En caso de no representar a ninguna estación mostrar un mensaje que diga “error”. Probar la función creada llamándola con A, P, O, B, V e I.

Estructuras de control iterativas

8. Se quiere hacer un programa para enseñar a unos niños a contar. Crear una función que reciba un número entero e imprima por pantalla los números del 1 hasta ese número con la estructura de control iterativa **for**.
9. Se quiere mejorar el programa para enseñar matemáticas pensado en el ejercicio anterior. Ahora se necesita una funcionalidad que permita a los niños aprender las tablas. Crear una función que reciba un número entero e imprima por pantalla la tabla de ese número del 1 al 10.
10. Crear una función que simule un cumpleaños: que dado un entero imprima “Que los cumplas feliz” esa cantidad de veces.
11. En un almacén están buscando la forma de hacer los cobros más automáticamente. Para esto, se nos pide crear una función que reciba un número entero que representa lo que hay que cobrar, le pida al usuario ingresar un monto, y se vaya mostrando por pantalla cuánto falta para completar el pago. Repetir este proceso **hasta** que la deuda sea 0 o menor. Por ejemplo, si se recibe el monto 30:
 > **El importe a pagar es de 30 pesos. Por favor, ingrese un monto.**
 > 10
 > **El importe a pagar es de 20 pesos. Por favor, ingrese un monto.**
 > 15
 > **El importe a pagar es de 5 pesos. Por favor, ingrese un monto.**
 > 5

Estructuras de control condicionales e iterativas

12. Escribir código que recorra los números del 1 al 20 y determine para cada uno si es par o impar, imprimiendo un mensaje por pantalla en cada caso. Es decir, el output esperado sería:
 > **El número 1 es impar.**
 > **El número 2 es par.**
 ...
 > **El número 20 es par.**
13. Se tiene una máquina de sacar juguetes que funciona cuando se ingresa una determinada cantidad de fichas, y se quiere hacer una función que imite ese comportamiento.

- a. Hacer una función que reciba un número que represente el precio de la máquina, e imprima por pantalla “Ingresá x fichas para comenzar” hasta que se hayan ingresado esa cantidad de letras F (que representan una ficha), y luego “¡A jugar!” cuando se hayan ingresado todas las fichas necesarias. Por ejemplo, si la función recibe 3, debería tener el siguiente comportamiento:

```
> Ingresá 3 fichas para comenzar
> F
> Ingresá 3 fichas para comenzar
> F
> Ingresá 3 fichas para comenzar
> B
> Ingresá 3 fichas para comenzar
> F
> ¡A jugar!
```

ATENCIÓN: notar cómo cuando se ingresa una letra distinta de F, esta se ignora a la hora de contar la cantidad de fichas que fueron ingresadas.

- b. Ahora se quiere que se vaya mostrando por pantalla, cuántas fichas FALTAN ingresar para empezar a jugar. Es decir:

```
> Ingresá 3 fichas (F) para comenzar
> F
> Ingresá 2 fichas (F) para comenzar
> F
> Ingresá 1 fichas (F) para comenzar
> B
> Ingresá 1 fichas (F) para comenzar
> F
> ¡A jugar!
```

- c. Agregar a la función el mensaje de error “Por favor solamente ingrese fichas reales (F)” cuando se recibe una letra distinta de F.

14. Crear una función que reciba un número entero e imprima los números primos entre 0 y el número ingresado.

AYUDA: un número es primo cuando solamente es divisible por sí mismo y por 1, es decir que para ver si es primo hay que ver que el módulo (%) de ese número con los anteriores hasta el 1 (sin incluirlo) sea distinto de 0, o sea que no sea divisible.