

# Desafíos de Programación

**¡Importante!** Resolver en orden dado que va avanzando en grado de complejidad.

**¡Recuerda!** Pueden existir más de una forma de resolver.

1- Escriba un programa que pida al usuario un entero de tres dígitos, y entregue el número con los dígitos en orden inverso:

```
Ingrese numero: 345  
543
```

```
Ingrese numero: 241  
142
```

2- Escriba un programa que pregunte al usuario la hora actual  $t$  del reloj y un número entero de horas  $h$ , que indique qué hora marcará el reloj dentro de  $h$  horas:

```
Hora actual: 11  
Cantidad de horas: 43  
En 43 horas, el reloj marcara las 6
```

3- Escribir un programa que pida al usuario un número entero y muestre por pantalla si es un número primo o no.

```
Ingrese un número: 2  
El número 2 es primo
```

4- **Tiempo de viaje.** Un viajero desea saber cuánto tiempo tomó un viaje que realizó. Él tiene la duración en minutos de cada uno de los tramos del viaje.

Desarrolle un programa que permita ingresar los tiempos de viaje de los tramos y entregue como resultado el tiempo total de viaje en formato horas:minutos.

El programa deja de pedir tiempos de viaje cuando se ingresa un 0.

```
Duración tramo: 15  
Duración tramo: 30  
Duración tramo: 87  
Duración tramo: 0  
Tiempo total de viaje: 2:12 horas
```

---

<sup>1</sup> Recuperado de: <http://progra.usm.cl/apunte/ejercicios/1/tiempo-de-viaje.html>

5- Cuando la Tierra completa una órbita alrededor del Sol, no han transcurrido exactamente 365 rotaciones sobre sí misma, sino un poco más. Más precisamente, la diferencia es de más o menos un cuarto de día.

Para evitar que las estaciones se desfasen con el calendario, el calendario juliano introdujo la regla de introducir un día adicional en los años divisibles por 4 (llamados bisiestos), para tomar en consideración los cuatro cuartos de día acumulados.

Sin embargo, bajo esta regla sigue habiendo un desfase, que es de aproximadamente  $3/400$  de día.

Para corregir este desfase, en el año 1582 el papa Gregorio XIII introdujo un nuevo calendario, en el que el último año de cada siglo dejaba de ser bisiesto, a no ser que fuera divisible por 400.

Escriba un programa que indique si un año es bisiesto o no, teniendo en cuenta cuál era el calendario vigente en ese año:<sup>2</sup>

```
Ingrese el año a evaluar: 2011
```

```
El año 2011 NO es bisiesto
```

```
Ingrese el año a evaluar: 2024
```

```
El año 2024 es bisiesto
```

6- Escribir un programa que pida al usuario un número entero y muestre por pantalla un triángulo rectángulo como el de más abajo con tantos renglones como indique el usuario.

```
Ingrese cantidad de renglones: 5
```

```
2
4 2
6 4 2
8 6 4 2
10 8 6 4 2
```

7- La secuencia de Collatz de un número entero se construye de la siguiente forma:

- si el número es par, se lo divide por dos;
- si es impar, se le multiplica tres y se le suma uno;

<sup>2</sup> Recuperado de <http://progra.usm.cl/apunte/ejercicios/1/bisiestos.html>

- La sucesión termina al llegar a uno.

Desarrolle un programa que entregue la secuencia de Collatz de un número entero:

3

Ingrese un número: 18

18 9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Ingrese un número: 19

19 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Ingrese un número: 20

20 10 5 16 8 4 2 1

8- Este problema apareció en el certamen recuperativo 1 del segundo semestre de 2011 en el campus Vitacura.

Una máquina de alimentos tiene productos de tres tipos, A, B y C, que valen respectivamente \$270, \$340 y \$390. La máquina acepta y da de vuelto monedas de \$10, \$50 y \$100.

Escriba un programa que pida al usuario elegir el producto y luego le pida ingresar las monedas hasta alcanzar el monto a pagar. Si el monto ingresado es mayor que el precio del producto, el programa debe entregar las monedas de vuelto, una por una.<sup>4</sup>

Elija producto: A

Ingrese monedas:

100

10

50

100

100

Su vuelto:

50

10

10

10

10

<sup>3</sup> Recuperado de: <http://progra.usm.cl/apunte/ejercicios/1/collatz.html>

<sup>4</sup> Recuperado de <http://progra.usm.cl/apunte/ejercicios/1/maquina-alimentos.html>

Elija producto: B

Ingrese monedas:

100

100

100

100

Su vuelto:

50

10

Elija producto: C

Ingrese monedas:

100

100

50

10

100

10

10

10

9 - **Anagrama.** Escribe una función que reciba dos palabras y retorne verdadero o falso según sean o no anagramas.

- Un Anagrama consiste en formar una palabra reordenando TODAS las letras de otra palabra inicial.
- NO hace falta comprobar que ambas palabras existen.
- Dos palabras exactamente iguales no son anagrama.

palabra1: Sergio

palabra2: Riesgo

true

10- **Torre y Alfil.** Un tablero de ajedrez es una grilla de ocho filas y ocho columnas, numeradas de 1 a 8. Dos de las piezas del juego de ajedrez son el alfil y la torre. El alfil se desplaza en diagonal, mientras que la torre se desplaza horizontal o verticalmente. Una pieza puede ser capturada por otra si está en una casilla a la cual la otra puede desplazarse:

Escriba un programa que reciba como entrada las posiciones en el tablero de un alfil y de una torre, e indique cuál pieza captura a la otra:

Fila alfil: 7  
Columna alfil: 6  
Fila torre: 4  
Columna torre: 3  
Alfil captura

Fila alfil: 3  
Columna alfil: 4  
Fila torre: 7  
Columna torre: 4  
Torre captura

Fila alfil: 3  
Columna alfil: 3  
Fila torre: 8  
Columna torre: 5  
Ninguna captura

11- **Piedra, papel y tijera.** En cada ronda del juego del cachipún, los dos competidores deben elegir entre jugar tijera, papel o piedra.

Las reglas para decidir quién gana la ronda son: tijera le gana a papel, papel le gana a piedra, piedra le gana a tijera, y todas las demás combinaciones son empates.

El ganador del juego es el primero que gane tres rondas.

Escriba un programa que pregunte a cada jugador cuál es su jugada, muestre cuál es el marcador después de cada ronda, y termine cuando uno de ellos haya ganado tres rondas. Los jugadores deben indicar su jugada escribiendo tijera, papel o piedra.<sup>6</sup>

A: tijera  
B: papel  
1 - 0  
  
A: tijera  
B: tijera  
1 - 0

<sup>5</sup> Recuperado de: <http://progra.usm.cl/apunte/ejercicios/1/torre-y-alfil.html>

<sup>6</sup> Recuperado de: <https://progra-utfsm.readthedocs.io/en/latest/ejercicios/1/cachipun.html>

```
A: piedra
B: papel
1 - 1

A: piedra
B: tijera
2 - 1

A: papel
B: papel
2 - 1

A: papel
B: piedra
3 - 1

A es el ganador
```

12- **Torneo de Tenis.** Escriba un programa para simular un campeonato de tenis.

Primero, debe pedir al usuario que ingrese los nombres de ocho tenistas. A continuación, debe pedir los resultados de los partidos juntando los jugadores de dos en dos. El ganador de cada partido avanza a la ronda siguiente.

El programa debe continuar preguntando ganadores de partidos hasta que quede un único jugador, que es el campeón del torneo.<sup>7</sup>

```
Jugador 1: Nadal
Jugador 2: Melzer
Jugador 3: Murray
Jugador 4: Soderling
Jugador 5: Djokovic
Jugador 6: Berdych
Jugador 7: Federer
Jugador 8: Ferrer

Ronda 1
a.Nadal - b.Melzer: a
a.Murray - b.Soderling: b
a.Djokovic - b.Berdych: a
```

<sup>7</sup> Recuperado de: <https://progra-utfsm.readthedocs.io/en/latest/ejercicios/2/torneo-tenis.html>

a.Federer - b.Ferrer: a

Ronda 2

a.Nadal - b.Soderling: a

a.Djokovic - b.Federer: a

Ronda 3

a.Nadal - b.Djokovic: b

Campeon: Djokovic

13- El diccionario países asocia cada persona con el conjunto de los países que ha visitado:

```
países = {
    'Pepito': {'Chile', 'Argentina'},
    'Yayita': {'Francia', 'Suiza', 'Chile'},
    'John': {'Chile', 'Italia', 'Francia', 'Peru'},
}
```

Escriba una función `cuantos_en_comun(a, b)`, que indique cuántos países en común han visitado la persona a y la persona b:<sup>8</sup>

```
>>> cuantos_en_comun('Pepito', 'John')
1
>>> cuantos_en_comun('John', 'Yayita')
2
```

14- **Signo zodiacal.** El signo zodiacal de una persona está determinado por su día de nacimiento.

El diccionario de signos asocia a cada signo el período del año que le corresponde. Cada período es una tupla con la fecha de inicio y la fecha de término, y cada fecha es una tupla (mes, día):

```
signos = {
    'aries': (( 3, 21), ( 4, 20)),
    'tauro': (( 4, 21), ( 5, 21)),
    'geminis': (( 5, 22), ( 6, 21)),
    'cancer': (( 6, 22), ( 7, 23)),
    'leo': (( 7, 24), ( 8, 23)),
    'virgo': (( 8, 24), ( 9, 23)),
    'libra': (( 9, 24), (10, 23)),
    'escorpio': ((10, 24), (11, 22)),
    'sagitario': ((11, 23), (12, 21)),
    'capricornio': ((12, 22), ( 1, 20)),
    'acuuario': (( 1, 21), ( 2, 19)),
    'piscis': (( 2, 20), ( 3, 20)),
}
```

<sup>8</sup> Recuperado de <https://progra-utfsm.readthedocs.io/en/latest/ejercicios/2/paises.html>

Por ejemplo, para que una persona sea de signo libra debe haber nacido entre el 24 de septiembre y el 23 de octubre.

Escriba la función **determinar\_signo(fecha\_de\_nacimiento)** que reciba como parámetro la fecha de nacimiento de una persona, representada como una tupla (año, mes, día), y que retorne el signo zodiacal de la persona:

```
>>> determinar_signo((1990, 5, 7))
'tauro'
>>> determinar_signo((1904, 11, 24))
'sagitario'
>>> determinar_signo((1998, 12, 28))
'capricornio'
>>> determinar_signo((1999, 1, 11))
'capricornio'
```

15- **Autores de Libros.** Este problema apareció en el certamen 2 del segundo semestre de 2011 en el campus Vitacura.

Escriba las funciones necesarias para que el siguiente programa funcione:

```
9 libros = [
    ('Papelucho programador', 'Marcela Paz', 1983),
    ('Don Python de la Mancha', 'Miguel de Cervantes', 1615),
    ('Raw_input y Julieta', 'William Shakespeare', 1597),
    ('La tuplamorfosis', 'Franz Kafka', 1915),
    # ...
]

autores = {
    # autor: nacimiento, defunción, idioma
    'William Shakespeare': ((1564, 4, 26), (1616, 5, 3), 'inglés'),
    'Franz Kafka': ((1883, 7, 3), (1924, 6, 3), 'alemán'),
    'Marcela Paz': ((1902, 2, 28), (1985, 6, 12), 'español'),
    'Miguel de Cervantes': ((1547, 9, 29), (1616, 4, 22), 'español'),
    # ...
}

titulo = input('Ingrese título del libro: ')
print 'El libro fue escrito en', obtener_idioma(titulo),
print 'por', obtener_autor(titulo)
print 'El autor falleció', calcular_años_antes_de_morir(titulo), 'años',
print 'después de haber escrito el libro'
```

16- **Código Morse.** Crea un programa que sea capaz de transformar texto natural a código morse y viceversa.

<sup>9</sup> Recuperado de: <https://progra-utfsm.readthedocs.io/en/latest/ejercicios/2/autores-libros.html>



- Debe detectar automáticamente de qué tipo se trata y realizar la conversión.
- En morse se soporta raya "—", punto ".", un espacio " " entre letras o símbolos y dos espacios entre palabras " " .
- El alfabeto morse soportado será el mostrado en [https://es.wikipedia.org/wiki/Código\\_morse](https://es.wikipedia.org/wiki/Código_morse)<sup>10</sup>

---

<sup>10</sup> Recuperado de: <https://retosdeprogramacion.com/ejercicios/>