Cap3 **Tipos de datos**

1. Escriba un programa en Python que acepte el radio de un círculo y compute su área

(solución).

Entrada: 5

Salida: 78.5

2. Escriba un programa en Python que acepte el radio de una esfera y obtenga su volumen

(solución).

Entrada: 5

Salida: 523.3333333333334

3. Escriba un programa en Python que acepte la base y la altura de un triángulo y

compute su área (solución).

Entrada: base=4; altura=5

Salida: 10.0

4. Escriba un programa en Python que compute el futuro valor de una cantidad de dinero,

a partir del capital inicial, el tipo de interés y el número de años (solución).

Entrada: capital=10000; interés=3.5; años=7

**62 Capítulo 3. Tipos de datos**

**Aprende Python**

Salida: 12722.792627665729

5. Escriba un programa en Python que calcule la distancia euclídea entre dos puntos

(𝑥1, 𝑦1) y (𝑥2, 𝑦2) (solución).

Entrada: (𝑥1 = 3, 𝑦1 = 5); (𝑥2 = −7, 𝑦2 = −4)

Salida: 13.45362404707371

**EJERCICIOS DE REPASO Cadenas de texto**

1. Escriba un programa en Python que acepte el nombre y los apellidos de una persona

y los imprima en orden inverso separados por una coma. Utilice *f-strings* para

implementarlo (solución).

Entrada: nombre=Sergio; apellidos=Delgado Quintero

Salida: Delgado Quintero, Sergio

2. Escriba un programa en Python que acepte una ruta remota de recurso samba, y lo

separe en nombre(IP) del equipo y ruta (solución).

Entrada: //1.1.1.1/eoi/python

Salida: equipo=1.1.1.1; ruta=/eoi/python

3. Escriba un programa en Python que acepte un «string» con los 8 dígitos de un NIF, y

calcule su dígito de control (solución).

Entrada: 12345678

Salida: 12345678Z

4. Escriba un programa en Python que acepte un entero n y compute el valor de n + nn

+ nnn (solución).

Entrada: 5

Salida: 615

5. Escriba un programa en Python que acepte una palabra en castellano y calcule una

métrica que sea el número total de caracteres de la palabra multiplicado por el número

total de vocales que contiene la palabra (solución).

Entrada: ordenador

Salida: 36

Cap4 **Condicionales**

Escriba un programa en Python que acepte la opción de dos jugadoras en

Piedra-Papel-Tijera y decida el resultado (solución).

Entrada: persona1=piedra; persona2=papel

Salida: Gana persona2: El papel envuelve a la piedra

2. Escriba un programa en Python que acepte 3 números y calcule el mínimo (solución).

Entrada: 7, 4, 9

Salida: 4

3. Escriba un programa en Python que acepte un país (como «string») y muestre por

pantalla su bandera (como «emoji»). *Puede restringirlo a un conjunto limitado de*

*países* (solución).

Entrada: Italia

Salida: 􀀀􀀀

4. Escriba un programa en Python que acepte 3 códigos de teclas y muestre por pantalla

la acción que se lleva a cabo en sistemas Ubuntu Linux (solución).

Entrada: tecla1=Ctrl; tecla2=Alt; tecla3=Del;

Salida: Log out

5. Escriba un programa en Python que acepte edad, peso, pulso y plaquetas, y determine

si una persona cumple con estos requisitos para donar sangre.

Entrada: edad=34; peso=81; heartbeat=70; plaquetas=150000

Salida: Apto para donar sangre

**Control de flujo**

Escriba un programa en Python que realice las siguientes 9 multiplicaciones. ¿Nota

algo raro? (solución)

1 ・ 1

11 ・ 11

111 ・ 111

...

111111111 ・ 111111111

2. Escriba un programa en Python que acepte una cadena de texto e indique si todos sus

caracteres son alfabéticos (solución).

Entrada: hello-world

Salida: Se han encontrado caracteres no alfabéticos

3. Escriba un programa en Python que acepte un número entero 𝑛 y realice el siguiente

cálculo de productos sucesivos (solución):

Π︁𝑛

𝑖=1

𝑥2

𝑖

= 𝑥20

・ 𝑥21

・ 𝑥22

・ ・ ・ ・ ・ 𝑥2

𝑛

4. Escriba un programa en Python que acepte dos cadenas de texto y compute el producto

cartesiano letra a letra entre ellas (solución).

Entrada: cadena1=abc; cadena2=123

Salida: a1 a2 a3 b1 b2 b3 c1 c2 c3

5. Escriba un programa en Python que acepte dos valores enteros (𝑥 e 𝑦) que

representarán un punto (objetivo) en el plano. El programa simulará el movimiento de

un «caballo» de ajedrez moviéndose de forma alterna: 2 posiciones en 𝑥 + 1 posición en

𝑦. El siguiente movimiento que toque sería para moverse 1 posición en 𝑥 + 2 posiciones

en 𝑦. El programa deberá ir mostrando los puntos por los que va pasando el «caballo»

hasta llegar al punto objetivo (solución).

Entrada: objetivo\_x=7; objetivo\_y=8;

Salida: (0, 0) (1, 2) (3, 3) (4, 5) (6, 6) (7, 8)

Cap5 **Estructuras de datos**

Escriba un programa en Python que acepte una lista de valores numéricos y obtenga

su valor máximo **sin utilizar** la función «built-in» max() (solución).

Entrada: [6, 3, 9, 2, 10, 31, 15, 7]

Salida: 2

2. Escriba un programa en Python que acepte una lista y elimine sus elementos duplicados

(solución).

Entrada: [“this”, “is”, “a”, “real”, “real”, “real”, “story”]

Salida: [“this”, “is”, “a”, “real”, “story”]

3. Escriba un programa en Python que acepte una lista – que puede contener sublistas

(sólo en 1 nivel de anidamiento) – y genere otra lista «aplanada» (solución).

Entrada: [0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100, 110, 120]]

Salida: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]

4. Escriba un programa en Python que acepte una lista y genere otra lista eliminando los

elementos duplicados consecutivos (solución).

Entrada: [0, 0, 1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 8, 9, 4, 4]

Salida: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 4]

5. Escriba un programa en Python que acepte una lista de listas representando una matriz

numérica y compute la suma de los elementos de la diagonal principal (solución).

Entrada: [[4, 6, 1], [2, 9, 3], [1, 7, 7]]

Salida: 20

**Estructuras de datos**

Escriba un programa en Python que acepte una lista de palabras y las agrupe por su

letra inicial usando un diccionario (solución).

Entrada: [ “mesa”, “móvil”, “barco”, “coche”, “avión”, “bandeja”, “casa”,

“monitor”, “carretera”, “arco”]

Salida: {“m”: [“mesa”, “móvil”, “monitor”], “b”: [“barco”, “bandeja”], “c”:

[“coche”, “casa”, “carretera”], “a”: [“avión”, “arco”]}

2. Escriba un programa en Python que acepte un diccionario y determine si todos los

valores son iguales o no (solución).

Entrada: {“Juan”: 5, “Antonio”: 5, “Inma”: 5, “Ana”: 5, “Esteban”: 5}

Salida: Same values

3. Escriba un programa en Python que acepte una lista de listas con varios elementos y

obtenga un diccionario donde las claves serán los primeros elementos de las sublistas

y los valores serán los restantes – como listas – (solución).

Entrada: [[“Episode IV - A New Hope”, “May 25”, 1977], [“Episode V -

The Empire Strikes Back”, “May 21”, 1980], [“Episode VI - Return of the

Jedi”, “May 25”, 1983]]

Salida: {“Episode IV - A New Hope”: [“May 25”, 1977], “Episode V - The

Empire Strikes Back”: [“May 21”, 1980], “Episode VI - Return of the Jedi”:

[“May 25”, 1983]}

4. Escriba un programa en Python que acepte un diccionario cuyos valores son listas y

borre el contenido de dichas listas (solución).

Entrada: {“C1”: [10, 20, 30], “C2”: [20, 30, 40], “C3”: [12, 34]}

Salida: {“C1”: [], “C2”: [], “C3”: []}

5. Escriba un programa en Python que acepte un diccionario y elimine los espacios de

sus claves respetando los valores correspondientes (solución).

Entrada: {“S 001”: [“Math”, “Science”], “S 002”: [“Math”, “English”]}

Salida: {“S001”: [“Math”, “Science”], “S002”: [“Math”, “English”]}

Cap 6 **Modularidad**

Escriba una función en Python que indique si un número está en un determinado

intervalo (solución).

Entrada: valor=3; lim\_inferior=2; lim\_superior=5

Salida: True

2. Escriba una función en Python que reciba una lista de valores enteros y devuelva otra

lista sólo con aquellos valores pares (solución).

Entrada: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Salida: [2, 4, 6, 8]

3. Escriba una función en Python que indique si un número es perfecto. *Utilice una*

*función auxiliar que calcule los divisores propios* (solución).

Entrada: 8128

Salida: True

4. Escriba una función en Python que determine si una cadena de texto es un palíndromo

(solución).

Entrada: ana lava lana

Salida: True

5. Escriba una función en Python que determine si una cadena de texto es un pangrama

(solución)

Entrada: The quick brown fox jumps over the lazy dog

Salida: True

**Objetos y Clases**

Escriba una clase en Python para representar una secuencia de ADN. De momento, la

clase sólo contendrá los siguientes elementos:

• 4 atributos de clase, cada uno representando una base nitrogenada con su valor

como un carácter.

• Constructor que recibe una secuencia de caracteres (bases).

• Método para representar el objeto en formato «string».

2. Continúe con el ejercicio anterior, y añada a la clase 4 propiedades que calculen el

número total de cada una de las bases presentes en la secuencia.

3. Continúe con el ejercicio anterior, y añada a la clase un método de instancia para sumar

dos secuencias de ADN. La suma se hará base a base y el resultado será el máximo de

cada letra(base).

4. Continúe con el ejercicio anterior, y añada a la clase un método de instancia para

obtener el porcentaje de aparición de cada base (usando las propiedades definidas en

ejercicios anteriores).

5. Continúe con el ejercicio anterior, y añada a la clase un método de instancia para

multiplicar dos secuencias de ADN. La multiplicación consiste en dar como salida una

nueva secuencia que contenga sólo aquellas bases que coincidan en posición en ambas

secuencias de entrada