

Departamento de Gestión de Proyectos y Sistemas

Algoritmos y Programación

Profesores: Luis Bello / José Quevedo

Preparador: Isabel Asenjo / Rommel Sanzonetti

GUÍA DE EJERCICIOS

TIPOS DE DATOS Y OPERADORES

1.1. CUENTA BANCARIA

Debes realizar un programa que ejecute las operaciones que se llevan a cabo en una cuenta bancaria.

1. 2020/04/10: El saldo inicial es de \$3480
2. 2020/04/11: Se hace una compra en una tienda de ropa y se gastan \$96
3. 2020/04/17: Se cobran \$1200 de salario
4. 2020/04/30: Se cobra un 3% del saldo de la cuenta en intereses
5. 2020/05/02: Se hace una compra en el supermercado y se gastan \$51

Como output se espera:

- Mostrar la cantidad de dinero recibida/retirada.
- Mostrar el saldo actual de la cuenta y la fecha, al iniciar el programa y cada vez que se realice una operación.

Por ejemplo:

Se han retirado \$96.

Saldo actual a la fecha (2020/04/11): \${saldo_restante}

2.1. REGISTRO DE DATOS BÁSICOS (1)

2.2. PALÍNDROMOS

2.3. CÓDIGO MORSE (1)

2.4. CALCULADORA (1)

- Suma
- Resta

- Multiplicación
- División
- Potencia
- Módulo
- Comparar (mayor o menor que)
- Valor absoluto

Debe preguntarse qué operación se desea hacer y tomarse dos números por teclado (excepto para valor absoluto). Finalmente, imprime el resultado.

2.5. UPPER & LOWER

Realiza un programa que tome por teclado una palabra y verifique que solo contenga letras. Luego, recorriendo el string, cada vez que haya una vocal, debe convertirse a mayúscula. Todas las consonantes deben estar en minúscula.

Ejemplos de output:

hAmbUrgUEsA

AprEndEr

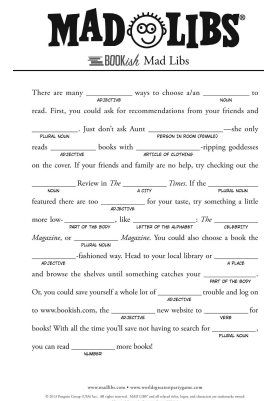
diApOsItivA

2.6. MAD LIBS

Este juego consiste en solicitar varias palabras en inglés por teclado y luego insertarlas en un texto con espacios vacíos para crear una historia.

Las palabras a solicitar son:

- 6 adjetivos (adj)
- 4 sustantivos plurales (plnoun)
- 1 ciudad (city)
- 1 lugar (place)
- 1 celebridad (cel)
- 1 número (num)
- 1 verbo (verb)
- 2 partes del cuerpo (body)
- 1 letra del alfabeto (letter)
- 2 sustantivos singulares (noun)
- 1 nombre femenino (name)
- 1 pieza de ropa (clothes)



El orden de las palabras en el texto es: adj1, noun1, plnoun1, name1, adj2, clothes, noun2, city1, plnoun2, adj3, body1, letter1, cell, plnoun3, adj4, place1, body2, adj5, adj6, verb1, plnoun4, num1.

Como output se espera el texto impreso con las palabras dadas.

```
historia = "There are many __ ways to choose a/an __ to read. First, you could
ask for recommendations from your friends and __. Just don't ask Aunt __ --
```

she only reads __ books with __-ripping goddesses on the cover. If your friends and family are no help, try checking out the __ Review in 'The __ Times'. If the __ featured there are too __ for your taste, try something a little more low-__, like __: 'The __ Magazine', or ' __ Magazine'. You could also choose a book the __-fashioned way. Head to your local library or __ and browse the shelves until something catches your __. Or, you could save yourself a whole lot of __ trouble and log on to www.bookish.com, the __ new website to __ for books! With all the time you'll save not having to search for __, you can read __ more books!"

2.7. CÁLCULO DE ÁREAS

Se requiere un sistema para calcular el área de figuras geométricas sencillas, donde el usuario ingresará en qué figura está interesado. El sistema debe permitir la solicitud de circunferencias, elipses, cuadrados, rectángulos, triángulos y rombos; solicitando para cada uno la información necesaria. Si existe información que sea constante, deben manejarla como tal.

2.8. FECHA (1)

Elabore un programa que se encargue de solicitar 3 números que serán el día, mes y año de una fecha. El programa debe indicar si la fecha es correcta y leerla por consola. El programa debe cumplir las siguientes características:

- Manejar meses de 28, 30 y 31 días.
- No trabajar con años bisiestos.

Por ejemplo, al ingresar día: 45, mes: 1, año: 2018, el programa debe enviar una alerta de que la fecha no es correcta. Por el contrario, si el programa recibe datos adecuados, debe indicar que la fecha es correcta e imprimirla (e.g., 25 de enero de 2018).

2.9. VERIFICACIÓN DE MAYORÍA DE EDAD

Escribir un programa que pregunte al usuario su edad y muestre por pantalla si es mayor de edad o no.

2.10. DIVISIONES

Escribir un programa que pida al usuario dos números y muestre por pantalla su división. Si el divisor es cero el programa debe mostrar un error.

2.11. PARIDAD

Escribir un programa que pida al usuario un número entero y muestre por pantalla si es par o impar.

2.12. CINE

Un cine requiere un programa para calcular el precio de sus entradas y te ha contratado para realizarlo. El programa debe preguntar al usuario la edad del cliente y mostrar el precio de la entrada.

- Si el cliente es menor de 4 años puede entrar gratis
- Si tiene entre 4 y 18 años debe pagar \$10
- Si es mayor de 18 años, \$14

Como output se espera:

Imprimir el precio correspondiente a la edad del cliente.

LISTAS Y LOOPS

3.1. SUMA DE DÍGITOS (1)

Realiza un programa que, dado un número entero positivo ingresado por teclado, imprima la suma de los números que lo componen. Si el valor de la suma es menor que 10, la respuesta es el valor obtenido, si no, es la suma de los dígitos de la suma de los dígitos. Por ejemplo: para 7 es 7, para 12 es 3, y para 845 es 8 ($8+4+5 = 17 \Rightarrow 1+7 = 8$).

3.2. CONTADOR DE SÍLABAS

Se tienen tres textos almacenados en tres variables como strings. Para cada texto debes contar cuántas veces se repite la sílaba indicada y qué índice ocupa cada repetición.

- x: "at"
- y: "om"
- z: "in"

Output:

- **x:**
 - **Índices: 13-14, 29-30, 104-105, 112-113**
 - **Repeticiones: 4**
- **y:**
 - **Índices: 12-13, 19-20, 335-336**
 - **Repeticiones: 3**
- **z:**
 - **Índices: 6-7, 104-105, 171-172, 203-204, 237-238, 371-372, 414-415**

- **Repeticiones: 7**

```
x = "Outside of that, Hu argues that the royalties can be more reliable than the entropy of Wall Street. No matter what happens to the stock market, people are always going to be listening to music."
```

```
y = "CBD, which comes from either a marijuana or hemp plant, is non-psychoactive, so it won't get you high like THC (tetrahydrocannabinol). It may, however, offer anti-inflammatory, antioxidant, anti-psychotic, anti-convulsant, and antidepressant properties. Because it can hemp is legal in all 50 states, hemp-derived CBD products are becoming more readily available online and in stores all over the country."
```

```
z = "According to a report published by Fast Company, apparel sales have plummeted by 50 percent. It's a margin that will be difficult to recover from, especially for small businesses that don't have the backing of major conglomerates. These independent companies rely on seasonal buys from retailers to stay afloat. But with the bevy of store closures and restrictions on online sales, these behemoth retailers are facing hard times of their own."
```

3.3. DADOS

Con dos dados, al azar, se pueden obtener números entre 2 y 12 de varias formas. Crea un programa que reciba por teclado un número entre 2 y 12 y retorne las combinaciones posibles de números para que su suma sea igual al número ingresado (no debe repetirse la combinación, por ejemplo, 4-5 y 5-4 debe mostrarse solo una vez).

Ejemplos de output:

- **Combinaciones para 5:**
 - * 1-4
 - * 2-3
- **Combinaciones para 9:**
 - * 3-6
 - * 4-5

3.4. SUCESIÓN DE FIBONACCI (1)

La sucesión de Fibonacci se calcula en base a la siguiente fórmula:

$$f(x) = 0 \text{ si } x = 0$$

$$f(x) = 1 \text{ si } x = 1$$

$$f(x) = f(x-1) + f(x-2) \text{ si } x > 1$$

Escribe un programa que, dado un número n, imprima la sucesión de Fibonacci, separada por comas, hasta el número igual o más cercano que haya por debajo de n.

Ejemplos de output:

- **Si $n = 8$:**
0,1,1,2,3,5,8
- **Si $n = 15$:**
0,1,1,2,3,5,8,13

3.5. TIPOS DE NÚMERO (1)

Realice un algoritmo que, dado un número ingresado por el usuario, determine si este es:

- Primo.
- Compuesto.
- Oblongo.
- Palíndromo.
- Perfecto.
- Abundante.
- Deficiente.
- Cuadrado.
- Libre de cuadrados.

3.6. CALCULADORA (2)

Elabore una calculadora que reciba del usuario sus respectivos operandos, realice la operación correspondiente entre los mismos, almacene este resultado y no lo muestre hasta que el usuario lo desee mostrar en pantalla o eliminarlo para realizar otra operación. La calculadora deberá ser capaz de ofrecerle al usuario las siguientes operaciones:

- Adición.
- Sustracción.
- Multiplicación.
- División.
- Potenciación.

3.7. FACTORIALES (1)

Realice un algoritmo que calcule la factorial de un número.

3.8. TABLA DE MULTIPLICAR

Realice un algoritmo que imprima por pantalla la tabla de multiplicar del 1 al 10.

3.9. SUCESIÓN DE FIBONACCI (2)

Elabore un algoritmo que imprima por pantalla los 10 primeros números de la sucesión Fibonacci.

3.10. NÚMEROS DE DÍGITOS INCREMENTALES (1)

Dado un entero X de tres dígitos diferentes, imprima "SÍ" si sus tres dígitos van en orden ascendente de izquierda a derecha e imprima "NO" de lo contrario.

3.11. SUCESIÓN DE FIBONACCI (3)

Elabore un algoritmo que dado un número introducido por el usuario te permita saber si ese número está o no en la sucesión Fibonacci.

3.12. SUCESIÓN DE FIBONACCI (4)

Elabore un algoritmo que le pida al usuario un número de partida y un número final, para luego imprimir por pantalla los números de la sucesión de Fibonacci contenidos dentro de ese rango de números.

3.13. PRODUCTO DE PRIMOS

Realice un algoritmo que dado un número N cualquiera, encuentre dos números primos A y B que al ser multiplicados resulte el número N. De no existir ningunos números A y B que cumplan la condición, el algoritmo deberá imprimir un mensaje de error.

3.14. JUEGO DE DADOS (1)

Realice un algoritmo que simule un juego de dados entre 2 jugadores, la idea de este juego obtener más puntos que el oponente, los mismos tendrán 7 turnos cada uno. El sistema de puntuación en los marcadores de cada jugador se verá reflejado por la cara del dado en el siguiente formato, si el dado después de lanzarlo tiene el valor de:

1	+10 puntos
2	+20 puntos
3	Vuelve a lanzar (sin que cuente como turno).
4	Puntuación X2
5	+40 puntos
6	Puntuación /2

Como output se espera:

Al final se deberá reflejar en pantalla el nombre y puntuación de los respectivos jugadores y luego imprimir un mensaje como: "El jugador ganador es: {nombre_del_jugador}".

3.15. CONTADOR DE LETRAS

Escribir un programa en el que se pregunte al usuario por una frase y una letra, y muestre por pantalla el número de veces que aparece la letra en la frase.

3.16. ECO

Escribir un programa que muestre el eco de todo lo que el usuario introduzca hasta que el usuario escriba "salir" que terminará el programa.

3.17. NÚMEROS PREVIOS

Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla la cuenta atrás desde ese número hasta cero separados por comas.

3.18. PALABRAS

Realice un algoritmo que dada una palabra ingresada por el usuario calcule cuántas letras tiene, cuántas consonantes tiene, cuántas vocales tiene y el número de veces en que cada vocal se repite durante la palabra.

3.19. PARES E IMPARES

Elabore un programa que dado 20 números generados de forma aleatoria imprima por pantalla primero los números pares y luego los impares.

3.20. BUSCADOR DE NOMBRES

Realice un programa que dado un array de nombres pida al usuario introducir un nombre y verificar si este está o no en el array.

3.21. OPERACIONES ENTRE MATRICES

Realice un algoritmo que realice las operaciones de suma y resta entre las siguientes matrices y el output del programa devuelva dos matrices, una correspondiente a cada operación.

Matrices:

$$A = \begin{pmatrix} 1 & 4 & 6 \\ 4 & 2 & 5 \\ 6 & 5 & 3 \end{pmatrix} \quad C = \begin{pmatrix} 1.3 & 20 & 12 \\ 1.8 & 28 & 15 \\ 2.5 & 40 & 18 \end{pmatrix}$$

3.22. MULTIPLICACIÓN DE UN ESCALAR POR UNA MATRIZ

Realice un algoritmo que dado un número ingresado por un usuario realice la operación de multiplicación por la matriz A del ejercicio anterior, el programa deberá mostrar como respuesta la matriz resultante de la misma.

TUPLAS Y SETS

4.1. GRÁFICA DE UNA FUNCIÓN

Para graficar una función en el plano cartesiano se ubican los puntos obtenidos a partir de la sustitución de valores de x en f(x).

Diseña un programa que cumpla con los siguientes requerimientos:

Parte 1:

El programa debe conseguir los valores de y correspondientes a los valores de x dados en la siguiente lista: `valores_x = [2,5,9,13]` para $f(x) = 2^x$ (2 elevado a la x), y almacenarlos en una lista de tuplas que deberá imprimirse posteriormente.

Output:

[(2,4),(5,32),(9,512),(13,8192)]

Parte 2:

El programa debe tomar las coordenadas de la siguiente lista: `coord2 = [(4,1), (25,11), (8,5), (16,8), (45,77), (10,10), (4,1), (34,2), (25,11), (8,5)]` y mostrarlos de la siguiente forma, después de eliminar todos los puntos repetidos y ordenar la lista de forma ascendente:

Ejemplo de output:

x	y
4	1
8	5

FUNCIONES

5.1.AGENDA

Deseas crear un programa que te ayude a organizar tus eventos y tareas. Para esto, tu programa debe:

- Solicitar por teclado el nombre del evento a agendar
- Mostrar ordenadamente los meses del año para que el usuario seleccione el mes del evento
- Mostrar ordenadamente los días del mes para que el usuario seleccione el día del evento
- Mostrar ordenadamente las 24 horas del día para que el usuario seleccione la hora del evento (en este programa, cada evento tendrá una duración de una hora)
- Verificar si ese horario ya fue ocupado anteriormente por otro evento
- Notificar que el evento fue creado con éxito o que el horario seleccionado no está disponible. En caso de que se de lo último, deberá solicitarse una nueva fecha
- Debe existir la opción de mostrar todos los eventos agendados

Nota: utilizar funciones

5.2. SORTEO

Una marca de joyería que realiza sus ventas a través de Instagram realizará un sorteo que tendrá tres ganadores. Se sorteará la siguiente mercancía:

- Primer lugar: 1 collar, 2 pulseras, 2 pares de zarcillos y 3 anillos.
- Segundo lugar: 1 collar, 1 pulsera y 2 anillos.
- Tercer lugar: 1 pulsera y 1 par de zarcillos.

Tu labor es diseñar para ellos un programa (con ayuda de la librería "random") que, de todos los usuarios que están participando (de máximo 30 caracteres, recibidos como input y almacenados en una lista) seleccione a los 3 ganadores al azar e indique cuál fue el premio.

Output:

1er lugar: @{primer_ganador} ha ganado 1 collar, 2 pulseras, 2 pares de zarcillos y 3 anillos.

2do lugar: @{segundo_ganador} ha ganado 1 collar, 1 pulsera y 2 anillos.

3er lugar: @{tercer_ganador} ha ganado 1 pulsera y 1 par de zarcillos.

Nota: el módulo de registro de participantes y el de selección del ganador deben estar en dos archivos diferentes, en el main solo deben ejecutarse las funciones.

5.3. LOTERÍA

Una nueva compañía de lotería te ha contratado para que diseñes un programa que les permita:

- Generar los tickets para vender
- Seleccionar aleatoriamente uno de los tickets para que sea el ganador

Cada ticket tiene 8 dígitos y debe poderse indicar por teclado la cantidad de tickets a generar.

Como output se espera:

Imprimir notificación de cuál fue el ticket ganador.

5.4. NÚMEROS EN VECTORES

Realice una función que dado un array o vector pasada por parámetro pedir un número al usuario y si este no está en el vector agregarlo al final del mismo, si no, imprimir en pantalla un mensaje que diga: "El número introducido ya está en el vector".

5.5. SIMULADOR DE BASE DE DATOS

Elabore un simulador de un sistema de base de datos con un array (arreglo) que guarde nombres de estudiantes y que tenga las siguientes funcionalidades:

- Guardar el nombre de estudiantes ingresados por consola.
- Imprimir el nombre de los estudiantes almacenados en la respectiva base de datos.
- Eliminar el nombre del estudiante.

5.6. SIMULADOR DE BASE DE DATOS - CONTINUACIÓN

Utilizando el ejercicio anterior (5.5) elabore el mismo sistema de base de datos con las mismas funcionalidades con un array para los nombres de los estudiantes, pero esta vez con la posibilidad de guardar además las asignaturas que cursa cada estudiante y sus cédulas. El programa deberá ofrecer las siguientes funcionalidades:

- Guardar el nombre de estudiantes ingresados por consola, su cédula y sus asignaturas cursantes.
- Imprimir el nombre de los estudiantes almacenados en la base de datos, seguido de su respectiva cédula y las asignaturas que cursa. De esta forma:
"Nombre: Rommel Sanzonetti,
Cédula: 28013672,
Asignaturas: Matemáticas 3, Algoritmos, Estructuras de Datos".
- Eliminar el estudiante del simulador de base de datos (eliminando así su respectiva cédula y las respectivas materias cursantes del mismo).

5.7. FACTORIALES (2)

Escribir una función que reciba un número entero positivo y devuelva su factorial.

5.8. FACTURACIÓN

Escribir una función que calcule el total de una factura tras aplicarle el IVA. La función debe recibir la cantidad sin IVA y el porcentaje de IVA a aplicar, y devolver el total de la factura. Si se invoca la función sin pasarle el porcentaje de IVA, deberá aplicar un 21%.

5.9. ÁREA DE UN TRIÁNGULO

Realice una función que calcule el área de un triángulo, cuyos parámetros sean la base y la altura del mismo para su respectivo cálculo.

5.10. TIPOS DE NÚMERO (2)

Realice una función que verifique si un número es:

- Primo.
- Compuesto.
- Oblongo.
- Palíndromo.
- Perfecto.
- Cuadrado.

5.11. CÁLCULO DE CUADRADOS

Escribir una función que reciba una muestra de números en una lista y devuelva otra lista con sus cuadrados.

DICCIONARIOS

6.1. ESTUDIO DEMOGRÁFICO

Se ha realizado un estudio demográfico en una urbanización y se te ha solicitado que, a partir de los datos obtenidos (almacenados en un diccionario de diccionarios), calcules las siguientes estadísticas:

- Cantidad total de residentes de la urbanización.
- Promedio de edad por género.
- Nombre y edad de la persona más joven.
- Nombre y edad de la persona más vieja.
- Cantidad de personas según su ocupación.

Output:

- **Total residentes: 37**
- **Promedio edades:**
 - **H: 44**
 - **M: 45**
- **Persona más joven: Luisana González, 5 año(s).**
- **Persona más vieja: Valerio Fiore, 97 año(s).**
- **Cantidad según ocupación:**

- Médicos: 6	- Psicólogos: 3
- Profesores: 4	- Abogados: 5
- Ingenieros: 4	- Desempleados: 4
- Administradores: 4	- Estudiantes: 7

```
residentes = {"Julia Rodríguez": {"Edad": 21, "Género": "M", "Ocupación":  
"Estudiante"}, "Santiago López": {"Edad": 33, "Género": "H", "Ocupación":  
"Abogado"}, "Luis González": {"Edad": 8, "Género": "H", "Ocupación":  
"Estudiante"}, "Luisana González": {"Edad": 5, "Género": "M", "Ocupación":  
"Estudiante"}, "Martina Reverón de González": {"Edad": 37, "Género": "M",  
"Ocupación": "Ingeniero"}, "Sergio González": {"Edad": 37, "Género": "H",  
"Ocupación": "Ingeniero"}, "Abel Araujo": {"Edad": 81, "Género": "H",  
"Ocupación": "Médico"}, "Roberto Mendoza": {"Edad": 56, "Género": "H",  
"Ocupación": "Profesor"}, "Bárbara Zapatero de Piñera": {"Edad": 32, "Género":  
"M", "Ocupación": "Psicólogo"}, "Leonardo Piñera": {"Edad": 31, "Género": "H",  
"Ocupación": "Administrador"}, "Gustavo Álvarez": {"Edad": 14, "Género": "H",  
"Ocupación": "Estudiante"}, "Guillermo Álvarez": {"Edad": 38, "Género": "H",  
"Ocupación": "Médico"}, "Laura Paz de Álvarez": {"Edad": 38, "Género": "M",  
"Ocupación": "Desempleado"}, "Ignacio Salcedo": {"Edad": 19, "Género": "H",  
"Ocupación": "Estudiante"}, "Saúl Salcedo": {"Edad": 22, "Género": "H",  
"Ocupación": "Estudiante"}, "Romina Salcedo": {"Edad": 25, "Género": "M",  
"Ocupación": "Administrador"}, "Elena Pinto de Salcedo": {"Edad": 52,  
"Género": "M", "Ocupación": "Abogado"}, "Mauricio Salcedo": {"Edad": 52,  
"Género": "H", "Ocupación": "Ingeniero"}, "Tatiana Echeverría": {"Edad": 68,  
"Género": "M", "Ocupación": "Médico"}, "Marcelo Gonçalves": {"Edad": 27,  
"Género": "H", "Ocupación": "Administrador"}, "Jessica Franco de Gonçalves":  
{"Edad": 26, "Género": "M", "Ocupación": "Profesor"}, "Valerio Fiore":  
{"Edad": 97, "Género": "H", "Ocupación": "Desempleado"}, "Giuliana Rossi de  
Fiore": {"Edad": 95, "Género": "M", "Ocupación": "Desempleado"}, "José  
Castro": {"Edad": 35, "Género": "H", "Ocupación": "Abogado"}, "Samantha Correa  
de Castro": {"Edad": 35, "Género": "M", "Ocupación": "Abogado"}, "Ángel  
Castro": {"Edad": 7, "Género": "H", "Ocupación": "Estudiante"}, "Antonietta  
Marín": {"Edad": 58, "Género": "M", "Ocupación": "Profesor"}, "Lorenzo  
Blanco": {"Edad": 77, "Género": "H", "Ocupación": "Abogado"}, "Vanessa Blanco  
de Bustamante": {"Edad": 37, "Género": "M", "Ocupación": "Médico"}, "Raúl  
Bustamante": {"Edad": 39, "Género": "H", "Ocupación": "Médico"}, "Carolina  
Alcalá": {"Edad": 24, "Género": "M", "Ocupación": "Ingeniero"}, "Juan Alcalá":  
{"Edad": 60, "Género": "H", "Ocupación": "Psicólogo"}, "Ingrid Gil de Alcalá":  
{"Edad": 60, "Género": "M", "Ocupación": "Profesor"}, "Francesca Costa de  
Gil": {"Edad": 88, "Género": "M", "Ocupación": "Médico"}, "Giancarlo Gil":  
{ 'Edad': 89, "Género": "H", "Ocupación": "Psicólogo"}, "César Lovera":  
{"Edad": 64, "Género": "H", "Ocupación": "Administrador"}, "Natalia Lovera":
```

```
{"Edad": 64, "Género": "M", "Ocupación": "Desempleado"}
```

6.2. CÓDIGO MORSE (2)

Este programa se hizo anteriormente pero usando una cadena de condicionales. Ahora, se pide el mismo programa pero utilizando **diccionarios**.

A continuación se te presentan 5 códigos postales escritos en código Morse, todos almacenados en un mismo string y separados por “*”. Tu labor consiste en descifrar cuáles son los números que componen cada código utilizando un diccionario que contenga lo mostrado en la tabla.

```
codigos_postales = "..._... ..*_...*_...  
...*_...*_...*_...*_..."
```

Output:

4752

1880

6236

5711

8499

1 ● — — — —	6 — ● ● ● ●
2 ● ● — — —	7 — — ● ● ●
3 ● ● ● — —	8 — — — ● ●
4 ● ● ● ● —	9 — — — — ●
5 ● ● ● ● ●	0 — — — — —

6.3. DIVISAS

Escribir un programa que guarde en una variable el diccionario `{'Euro':'€', 'Dollar':'$', 'Yen':'¥'}`, pregunte al usuario por una divisa y muestre su símbolo o un mensaje de aviso si la divisa no está en el diccionario.

6.4. REGISTRO DE DATOS BÁSICOS (2)

Escribir un programa que pregunte al usuario su nombre, edad, dirección y teléfono y lo guarde en un diccionario. Después debe mostrar por pantalla la información

6.5. PRECIOS

Escribir un programa que guarde en un diccionario los precios de las frutas de la tabla, pregunte al usuario por una fruta, un número de kilos y muestre por pantalla el precio de ese número de kilos de fruta. Si la fruta no está en el diccionario debe mostrar un mensaje informando de ello.

Fruta	Precio \$
Uva	1.35
Manzana	1.00
Pera	0.85
Naranja	0.70

6.6. FECHA (2)

Escribir un programa que pregunte una fecha en formato `dd/mm/aaaa` y muestre por pantalla la misma fecha en formato `dd de <mes> de aaaa` donde `<mes>` es el nombre del mes.

Ejemplo de output:

19 de mayo de 2020

6.7. FACTURAS

Escribir un programa que gestione las facturas pendientes de cobro de una empresa. Las facturas se almacenarán en un diccionario donde la clave de cada factura será el número de factura y el valor el coste de la factura. El programa debe preguntar al usuario si quiere añadir una nueva factura, pagar una existente o terminar. Si desea añadir una nueva factura se preguntará por el número de factura y su coste y se añadirá al diccionario. Si se desea pagar una factura se preguntará por el número de factura y se eliminará del diccionario. Después de cada operación el programa debe mostrar por pantalla la cantidad cobrada hasta el momento y la cantidad pendiente de cobro.

6.8. COMPRA

Escribir un programa que cree un diccionario simulando una cesta de la compra. El programa debe preguntar el artículo y su precio y añadir el par al diccionario, hasta que el usuario decida terminar. Después se debe mostrar por pantalla la lista de la compra y el coste total, con el siguiente formato:

Lista de la compra	
Artículo 1	Precio

Artículo 2	Precio
Artículo 3	Precio
...	...
Total:	Precio total

6.9. ASIGNATURAS

Escribir un programa que almacene el diccionario con los créditos de las asignaturas de un curso `{'Matemáticas': 6, 'Física': 4, 'Química': 5}` y después muestre por pantalla los créditos de cada asignatura en el formato `"<asignatura> tiene <créditos>"` créditos, donde `<asignatura>` es cada una de las asignaturas del curso, y `<créditos>` son sus créditos. Al final debe mostrar también el número total de créditos del curso.

ESTRUCTURAS DE DATOS COMBINADAS

7.1.FAST FOOD CHAIN

Una cadena de comida rápida te necesita para la creación de un programa con el que se pueda llevar una base de datos de los productos que ofrecen en su menú con su precio. Para ello, deberás aplicar lo que has aprendido sobre las diferentes estructuras de datos de Python. Este negocio ofrece hamburguesas, nuggets, papas fritas, helados y bebidas.

- Las hamburguesas pueden ser: clásica (\$4.00), doble carne (\$4.80), BBQ (\$4.40) y de pollo (\$4.20),.
- Los nuggets pueden ser: de 15 piezas (\$6.20) o de 30 piezas (\$10.50).
- Las papas fritas pueden ser: clásicas (\$2.80), con queso (\$3.10) o con queso y tocineta (\$3.90).
- Los helados pueden ser: de vainilla (\$1.30), de fresa (\$1.50) o de chocolate (\$1.50).
- Las bebidas pueden ser: agua (\$0.50), té frío (de durazno o limón) (\$1.10) o refresco (Coca-Cola, 7-Up o Freskolita) (\$0.90).

Como output se espera:

Deberá mostrarse de forma ordenada cada elemento del menú, con sus respectivas variaciones y precios. También se pide cuál es el producto más caro (nuggets de 30) y cuál es el más barato (agua) con su precio.

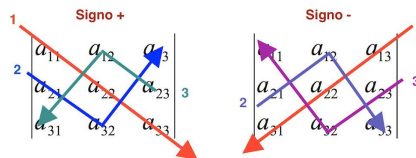
```
menu_lista = [("Hamburguesas","Clásica", 4.00), ("Hamburguesas","Doble Carne",
4.80),      ("Hamburguesas","BBQ",    4.40),      ("Hamburguesas","Pollo",    4.20),
("Nuggets","15  piezas",  5.20),  ("Nuggets","30  piezas",  9.50),  ("Papas
```

```
fritas","Clásicas", 2.80), ("Papas fritas","Queso", 3.10), ("Papas
fritas","Queso y tocineta", 3.90), ("Helados","Vainilla", 1.30),
("Helados","Fresa", 1.50), ("Helados","Chocolate", 1.50), ("Bebidas","Agua",
0.50), ("Bebidas","Té","Limón", 1.10), ("Bebidas","Té","Durazno", 1.10),
("Bebidas","Refresco","Coca-Cola", 0.90), ("Bebidas","Refresco","7-Up", 0.90),
("Bebidas","Refresco","Freskolita", 0.90)]
```

7.2. DETERMINANTE DE UNA MATRIZ

Las *matrices* podrían definirse como distribuciones rectangulares o cuadradas de números o funciones, están compuestas por filas y columnas.

Para las matrices cuadradas (mismo número de filas y columnas) existe lo que se conoce como *determinante*. El determinante de una matriz cuadrada es la suma de todos los productos que pueden formarse de tal forma que en cada producto haya solo un elemento de cada fila y solo uno de cada columna, por ejemplo:



Entonces, el cálculo del determinante de esta matriz de 3x3 sería:

$$\Delta = (a_{11} \cdot a_{22} \cdot a_{33}) + (a_{21} \cdot a_{32} \cdot a_{13}) + (a_{31} \cdot a_{12} \cdot a_{23}) - (a_{13} \cdot a_{22} \cdot a_{31}) - (a_{12} \cdot a_{21} \cdot a_{33}) - (a_{23} \cdot a_{32} \cdot a_{11})$$

Los tres productos que resultan de la diagonal principal ($a_{11} \cdot a_{22} \cdot a_{33}$) y las diagonales paralelas a ella, se suman, los otros tres productos se restan.

Realiza un programa que, a partir de la siguiente matriz de 3x3, retorne su determinante:

```
matriz = [[1, 3, 2], [4, 5, 7], [8, 6, 9]]
```

Output:

Determinante = 31

7.3. DESCUENTOS

Una panadería te ha contratado para que le diseñes un programa que tome los pedidos de sus clientes, a partir de la información de una base de datos dada, y generes una factura de su compra. Debes asignarle a cada cliente nuevo un número de factura, además de solicitar su nombre y su cédula.

Además de esto, esta panadería tiene un sistema de descuentos que funciona de la siguiente forma:

- Si el total de la compra es múltiplo de 3, se aplica un 10% de descuento a la compra
- Si es múltiplo de 7, se aplica un 12%
- Si es múltiplo de ambos, 3 y 7, se aplica un descuento de 15%

Como output se espera:

Imprimir la factura del cliente (nombre, cédula, items comprados con su precio y cantidad, y total a pagar después de aplicados los descuentos).

```
menu = {"Cachito": 4.00, "Empanada": 3.00, "Pasatelito": 3.50, "Sandwich": 2.50, "Pan tradicional (1 barra)": 1.00, "Pan especial (1 barra)": 1.75, "Café": 1.25, "Jugo": 2.00, "Agua": 0.75, "Dulces (por kilo)": 6.00, "Galletas (por kilo)": 5.75, "Torta": 10.25}
```

MÉTODOS DE ORDENAMIENTO

8.1. SORTING ALGORITHMS

Realice un programa que genere 10 números de forma aleatoria de modo que estos se almacenen en un array. Seguidamente el programa deberá preguntarle por cuál algoritmo de ordenamiento desea someter al respectivo array (método de la burbuja, selección o inserción) y de qué forma quiere que se genere el ordenamiento (descendente o ascendente), finalmente imprimir en pantalla dicho array.

8.2. SORTING ALGORITHMS - PARES E IMPARES

Elabore un programa que, dado 20 números generados de forma aleatoria, almacene en un array los números pares y en otro los impares; para luego ordenarlos de forma ascendente y seguidamente imprimir por pantalla los arrays ordenados.

PROGRAMACIÓN ORIENTADA A OBJETOS

9.1. SCHOOL DATABASE

Un colegio te ha contratado para que diseñes un programa que los ayude a almacenar de una manera más sencilla toda la información de sus alumnos. Cada alumno tiene nombre, grado que cursa, promedio, dirección de habitación, nombre del representante, teléfono del representante y la condición de becado o no.

Si el promedio del alumno es menor a 18, el alumno no tendrá beca; de lo contrario, sí la tendrá.

Luego de almacenar esto en su base de datos, debe existir una función que permita ver la

información organizada de cada alumno de la institución.

Como requerimientos adicionales pidieron:

- Mostrar los nombres, grados y promedios de las 5 personas con mejores promedios del colegio
- Mostrar un promedio general de todos los promedios de los alumnos del plantel
- Mostrar cuántos alumnos tienen promedios menores a 10, cuántos entre 10 y 15 (o sea, hasta 15.9) y cuántos entre 16 y 20.

Nota: trabajar con POO.

9.2. MALL DATABASE

Un centro comercial te ha contratado para que diseñes un programa que los ayude a almacenar de una manera más sencilla toda la información de sus tiendas. Cada tienda tiene nombre, tipo de tienda, piso en el que está (hay PB, 1, 2 y 3), RIF, nombre del propietario y teléfono del propietario.

Luego de almacenar esto en su base de datos, debe existir una función que permita ver la información organizada de cada local del centro comercial.

Como requerimientos adicionales pidieron:

- Al mostrar todas las tiendas registradas, deben aparecer ordenadas por nombre, piso o tipo, según se solicite.

Nota: trabajar con POO.

```
tipos = ["Women's Apparel", "Men's Apparel", "Children's Apparel", "Footwear",  
"Health & Beauty", "Jewelry & Watches", "Sporting Goods & Apparel", "Luggage &  
Handbags", "Optical/Eyewear", "Electronics", "Food", "Entertainment &  
Attractions", "Toys & Hobbies", "Home Décor", "Department Store", "Services",  
"Other"]
```

9.3. LIBRERÍA

Se desea implementar un sistema para el manejo de los libros de una librería, en donde cada libro tiene un título, un autor, número de páginas, número de capítulos, una editorial, número de edición, precio y cantidad disponible. En el caso de los autores, cada uno tendrá nombre, apellido, edad, cédula y nacionalidad. Para las editoriales, se maneja su nombre y dirección.

9.4. JUEGO DE DADOS (2)

Dado el problema 3.14 de la guía (Juego de dados (1)), plantee una simulación del respectivo juego con programación orientada a objetos.

NOTA: utilice la clase Jugador y plantee una solución al agregarle un atributo “puntuación” a la respectiva clase.

9.5. CARRERA

Desarrolle un programa que simule una carrera de 6 automóviles y el respectivo lugar de llegada de cada auto se verá dado aleatoriamente. Cada automóvil posee por lo menos dos características (atributos) que le permiten ser diferenciado con otros:

- Marca del automóvil.
- Conductor.

Al final el programa deberá ser capaz de imprimir en pantalla la lista de llegada de los automóviles de forma ordenada.

HERENCIA Y POLIMORFISMO

10.1. REGISTRO EN CONCURSO DE TALENTOS

Te han contratado para que diseñes un programa en Python para llevar a cabo el registro de los participantes de un concurso de talentos a realizarse próximamente. Deberás realizarlo utilizando tus conocimientos sobre **POO, herencia y polimorfismo**.

Tu labor consiste en:

- Diseñar una clase padre llamada *Participante*, y sus clases hijas, de acuerdo a las categorías del concurso (*Canto*, *Baile*, *Música*, *Libre*).
- La clase padre debe almacenar: nombre del concursante (o grupo), si es un solo o un grupo, categoría, número de teléfono y número de registro.
- Si participará en *Canto*, debe tener también el nombre de la canción a interpretar y el nombre de su autor.
- Si participará en *Baile*, debe incluirse el estilo de baile que interpretará.
- Si participará en *Música*, debe indicarse el (o los) instrumentos a tocar.
- Si participará en *Libre*, debe solicitarse además cuál será el talento a mostrar.
- La clase padre debe tener un método que imprima organizadamente los atributos del concursante.

Como output se espera:

Deberá mostrarse de forma ordenada, luego de registrar a todos los participantes, una lista con todos los atributos de cada uno. También se pide mostrar el total de participantes y cuántos hay por categoría.

10.2. MIEMBROS UNIVERSIDAD

Se desea desarrollar un sistema para el manejo de las clases, preparadurías, estudiantes,

preparadores y profesores. Cada profesor cuenta con su nombre, apellido, cédula y las secciones que dictan, las cuales son 3. Los estudiantes tienen su nombre, apellido, carnet y las secciones que cursan. Cada estudiante cursa 5 secciones. Las secciones tienen el nombre y código de la asignatura, el profesor que las dicta, salón asignado y horario.

LECTURA Y ESCRITURA DE ARCHIVOS

11.1. CRUD

Aplicando el método CRUD, crea un programa que maneje la base de datos de trabajadores de una tienda. Debe tener las opciones de:

- Registrar persona
 - Modificar persona
 - Ver personas
 - Eliminar persona
-

RECURSIVIDAD

12.1. NÚMEROS DE DÍGITOS INCREMENTALES (2)

Se le llama “número de dígitos incrementales” a aquel número para el que cada uno de sus dígitos es menor o igual al que le sigue. Por ejemplo, 1338 es un número de dígitos incrementales; porque $1 < 3$, $3 = 3$ y $3 < 8$. Por otro lado, 597 no lo es; porque $5 < 9$ pero $9 > 7$.

Diseña un programa que, tomando un número de dos o más dígitos por teclado, verifique si se trata de un número de dígitos incrementales o no.

Nota: aplicar recursividad.

12.2. SUMA DE DÍGITOS (2)

Este programa se hizo anteriormente pero usando un while loop. Ahora, se pide el mismo programa pero utilizando **recursividad**.

Realiza un programa que, dado un número entero positivo ingresado por teclado, imprima la

suma de los números que lo componen. Si el valor de la suma es menor que 10, la respuesta es el valor obtenido, si no, es la suma de los dígitos de la suma de los dígitos. Por ejemplo: para 7 es 7, para 12 es 3, y para 845 es 8 ($8+4+5 = 17 \Rightarrow 1+7 = 8$).

12.3. FACTORIALES (3)

Realice un algoritmo que calcule el factorial de un número.

12.4. SUCESIÓN DE FIBONACCI (5)

Realice un algoritmo que realice la sucesión Fibonacci.

12.5. LABERINTO

Un multimillonario desea saber qué entradas de su nuevo laberinto de arbustos llevan a la salida del mismo, pero solo dispone de una matriz con direcciones y un vector con las posiciones de las entradas; por ello, el multimillonario le ha contratado para que realice un algoritmo RECURSIVO que determine qué entradas llevan a la salida.

Esta es la matriz de la cual dispone el multimillonario:

```
matriz=[
["DOWN", "RIGHT", "RIGHT", "RIGHT", "RIGHT", "RIGHT", "DOWN"],
["DOWN", "DOWN", "LEFT", "LEFT", "LEFT", "LEFT", "LEFT"],
["DOWN", "WALL", "RIGHT", "DOWN", "DOWN", "LEFT", "LEFT"],
["RIGHT", "RIGHT", "UP", "DOWN", "DOWN", "RIGHT", "DOWN"],
["WALL", "DOWN", "LEFT", "LEFT", "RIGHT", "UP", "DOWN"],
["UP", "RIGHT", "RIGHT", "RIGHT", "RIGHT", "RIGHT", "DOWN"],
["UP", "LEFT", "LEFT", "LEFT", "LEFT", "LEFT", "EXIT"],
]
```

Y este es el vector: `vector = [[0,0],[6,5],[2,6],[0,1]]`

12.6. NERVIOSO

Desarrolla una función recursiva que simule una ronda del juego de cartas “Nervioso”. Para esto, partirás de una lista con todos los números de las cartas y usarás la librería “random” para la selección aleatoria de las cartas. A medida que corra el programa, deberá mostrarse en la consola por qué número va el conteo y qué carta aleatoria tiene el jugador actual. En el momento en el que la carta del jugador actual y el número del conteo coincidan, debe notificarse al usuario y culminar la función recursiva. De lo contrario, la función deberá seguir ejecutándose con una nueva carta aleatoria y el siguiente número del conteo.

Definición:

Nervioso consiste en que se ha de repartir toda la baraja (sin comodines) a partes iguales entre todos los jugadores boca abajo, los jugadores no deben mirar sus

cartas. Seguidamente se elige un jugador que inicie la partida. Este jugador comienza soltando la primera carta de su mazo pronunciando As. Cada jugador debe ir soltando sucesivamente su primera carta de su mazo pronunciando el número que sigue (dos, tres, cuatro, cinco, seis, siete, ocho, nueve, 10, J, Q, K) si un jugador al soltar su carta coincide con el número que le toca pronunciar, los otros jugadores no deben lanzar naipes o se llevarán todas las cartas, el último en colocar la mano debe llevarse todas las cartas de la mesa. Un jugador gana si no tiene cartas en su mazo pero puede volver a obtener cartas al perder si otros jugadores siguen soltando cartas.

TAREAS DE REDACCIÓN

SEMANA 4

Redacte un problema elaborado por usted de programación y plantee una respectiva respuesta a el mismo donde se involucren:

- Condicionales if- elif-else.
- Loops.

SEMANA 7

Redacte un problema elaborado por usted de programación y plantea una respectiva respuesta a el mismo donde involucre por lo menos:

- El manejo de vectores (arrays).
- Algún método de ordenamiento.

SEMANA 9

Redacte un problema elaborado por usted de programación y plantea una respectiva respuesta a el mismo donde involucre por lo menos:

- La programación orientada a objetos.
- Vectores.

Envía las respuestas las tareas de redacción a tu preparador vía correo:

- Sección 1***: isabel.asenjo@correo.unimet.edu.ve

***también recibo por Slack, pero el enunciado debe enviarse como imagen o PDF o algo así, no como un mensaje normal

- Sección 2: rommel.sanzonetti@correo.unimet.edu.ve

Asunto: "Tarea de redacción semana #X"

Cuerpo del mensaje o correo: "ZIP, RAR o URL (en caso de haberlo hecho en Repl.it)" con el enunciado y anexada la respectiva solución.