

Guía 04

1 - Generador de dirección de mail

Se desea un programa que: solicite al usuario un nombre, un apellido y el dominio y luego, proponga una dirección de mail para el nombre y apellido ingresado de acuerdo a las siguientes reglas:

- Componer la dirección de correo de la siguiente manera:
<primera letra del nombre><apellido>@<dominio>
Por ejemplo para Nombre = Felipe, Apellido= Steffolani y Dominio= frc.utn.edu.ar la dirección de mail sería:
fsteffolani@frc.utn.edu.ar
- Pero si la primera letra del nombre y la primera letra del apellido son la misma entonces utilizar:
<nombre>.<apellido>@<dominio>
Por ejemplo para Nombre= Soledad, Apellido= Steffolani y Dominio= colegiorosarito.edu.ar la dirección de mail sería:
soledad.steffolani@colegiorosarito.edu.ar

```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

# Generador de dirección de Mail
# Entradas:
# Nombre
# Apellido
# Dominio
# Salida:
# Dirección de mail propuesta

print('#' * 30)
print('#' + 'Generador de Mails' + \
      (' ' * (30-(len('Generador de Mails')+5))) + '#')
print('#' * 30)

print('\nIngreso de datos:')
nombre = input('\tNombre : ')
apellido = input('\tApellido: ')
dominio = input('\tDominio : ')

#transformar las cadenas ingresadas en minúscula
# independientemente de cómo se ingresaron.
nombre = nombre.lower()
apellido = apellido.lower()
dominio = dominio.lower()
primera_letra_nombre = nombre[0]
primera_letra_apellido = apellido[0]
```

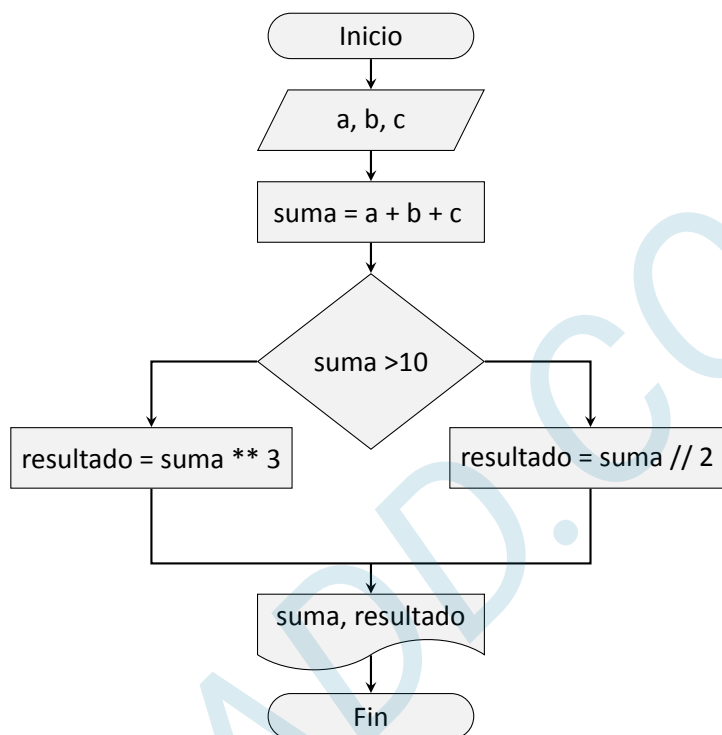
```
if primera_letra_apellido != primera_letra_nombre:
    mail_propuesto = primera_letra_nombre + apellido + '@' + \
    dominio
else:
    mail_propuesto = nombre + '.' + apellido + '@' + dominio

print()
print('Mail propuesto:\n\t', mail_propuesto)
print('#' * 30)

input('\nPresione enter para finalizar...')
```

2 - Suma - División - Potencia

Se necesita desarrollar un programa que permita calcular la suma de tres números. Si el resultado es mayor a 10 dividir por 2 (mostrar su resultado sin decimales), en caso contrario elevar el resultado al cubo.



```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'
# Carga de datos
a = int(input("Ingrese un número"))
b = int(input("Ingrese un número"))
c = int(input("Ingrese un número"))

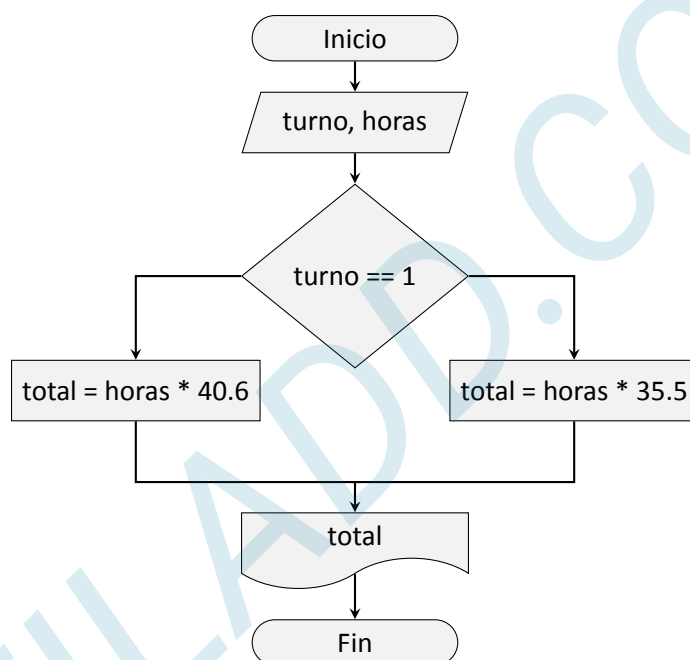
# Proceso
suma = a + b + c
if suma > 10:
    resultado = suma // 2
else:
    resultado = suma ** 3

# Presentación de resultados
print("La suma es:", suma)
print("El resultado es:", resultado)
```

3 - Jornal de un Operario

Se necesita desarrollar un programa para el área de recursos humanos de una empresa que permita informar el jornal de un determinado operario. Usted deberá cargar por teclado el código de turno que el operario trabajó ese día (1- representa Diurno y 2- representa Nocturno) y la cantidad de horas trabajadas.

La política de trabajo en la empresa es que los operarios de la misma pueden trabajar en el turno diurno o nocturno. Si un operario trabaja en el turno nocturno el pago es 40.60 pesos la hora, si lo hace en el turno diurno cobra 35.50 pesos la hora.



```

__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

print('Jornal de Operario')
print('=' * 80, '\n')

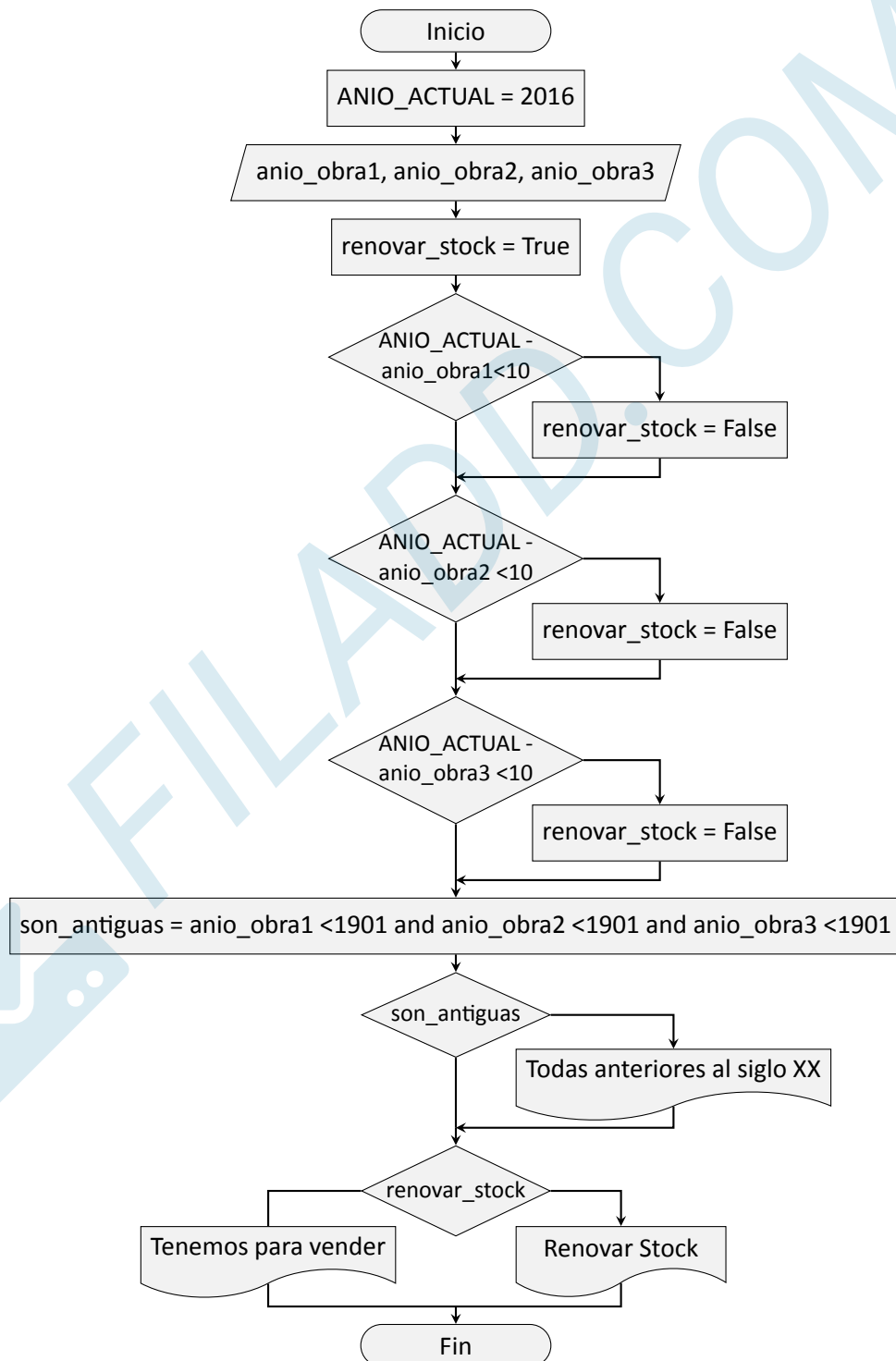
print("Ingreso de datos")
print('-' * 80)
turno = int(input('Ingrese el turno del operario (1- Diurno, 2- Nocturno): '))
horas = int(input('Ingrese la cantidad de horas trabajadas: '))

if turno == 1:
    total = horas * 35.5
else:
    total = horas * 40.60

print('\nPresentacion de resultados')
print('La empresa le debe pagar al operario un jornal de $', total)
  
```

4 - Galería de Arte

Una galería de arte desea preparar un catálogo de sus cuadros más famosos. Se realiza una prueba con tres cuadros y por cada uno se ingresa el año en que fue creado. El programa deberá verificar si todos los cuadros son anteriores al siglo XX (El siglo XX es el siglo pasado. Se inició en el año 1901 y terminó en el año 2000). Determinar cuántos tienen antigüedad inferior a 10 años. Si no hay ninguno, imprimir el mensaje "Renovar stock".



```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

ANIO_ACTUAL = 2016

print('La Galeria de Arte')
print('Lectura de Datos')
print('=' * 80)

anio_obra1 = int(input('Ingrese el año de creacion de la primer obra: '))
anio_obra2 = int(input('Ingrese el año de creacion de la segunda obra: '))
anio_obra3 = int(input('Ingrese el año de creacion de la tercer obra: '))

# Proceso para determinar si hay que renovar stock
renovar_stock = True
if ANIO_ACTUAL - anio_obra1 < 10:
    renovar_stock = False

if ANIO_ACTUAL - anio_obra2 < 10:
    renovar_stock = False

if ANIO_ACTUAL - anio_obra3 < 10:
    renovar_stock = False

son_antiguas = anio_obra1 < 1901 and anio_obra2 < 1901 and anio_obra3 < 1901

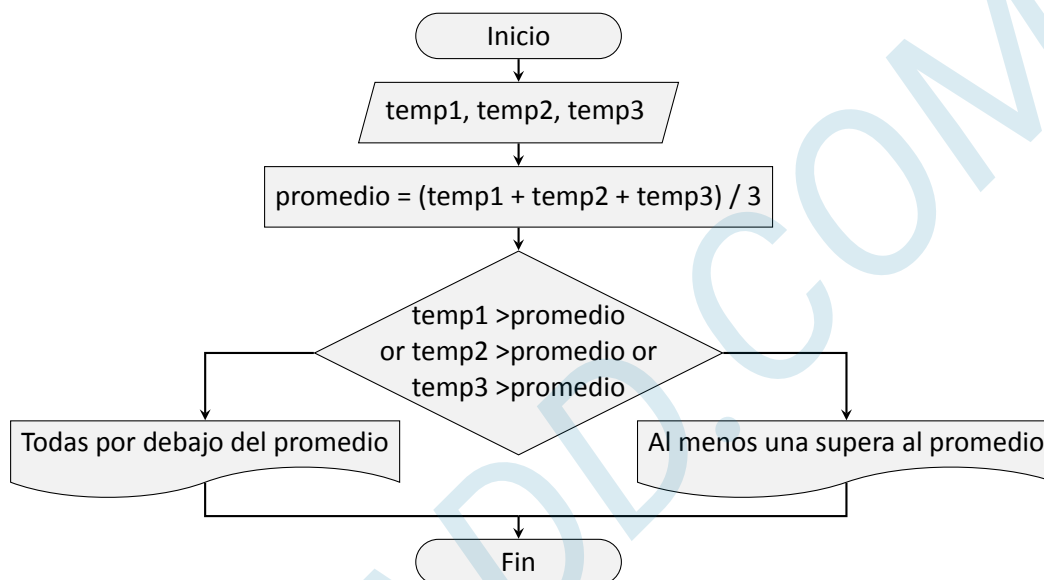
# Mostrar resultados
if son_antiguas:
    print('Todas las obras de arte son anteriores al siglo XX, carisimas!!!')

if renovar_stock:
    stock = 'Renovar Stock'
else:
    stock = 'Tenemos obras disponibles para vender'
```

5 - Temperatura diaria

Se solicita realizar un programa que permita ingresar tres temperaturas correspondientes a diferentes momentos de un día y determinar:

- Cual es el promedio de las temperaturas
- Si existe alguna temperatura que sea mayor al promedio



```

__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

import random

print('\nIngreso de temperaturas')
temp1 = int(input("Ingrese la temperatura 1"))
temp2 = int(input("Ingrese la temperatura 2"))
temp3 = int(input("Ingrese la temperatura 3"))

# Procesos calculo del promedio
promedio = (temp1 + temp2 + temp3) / 3

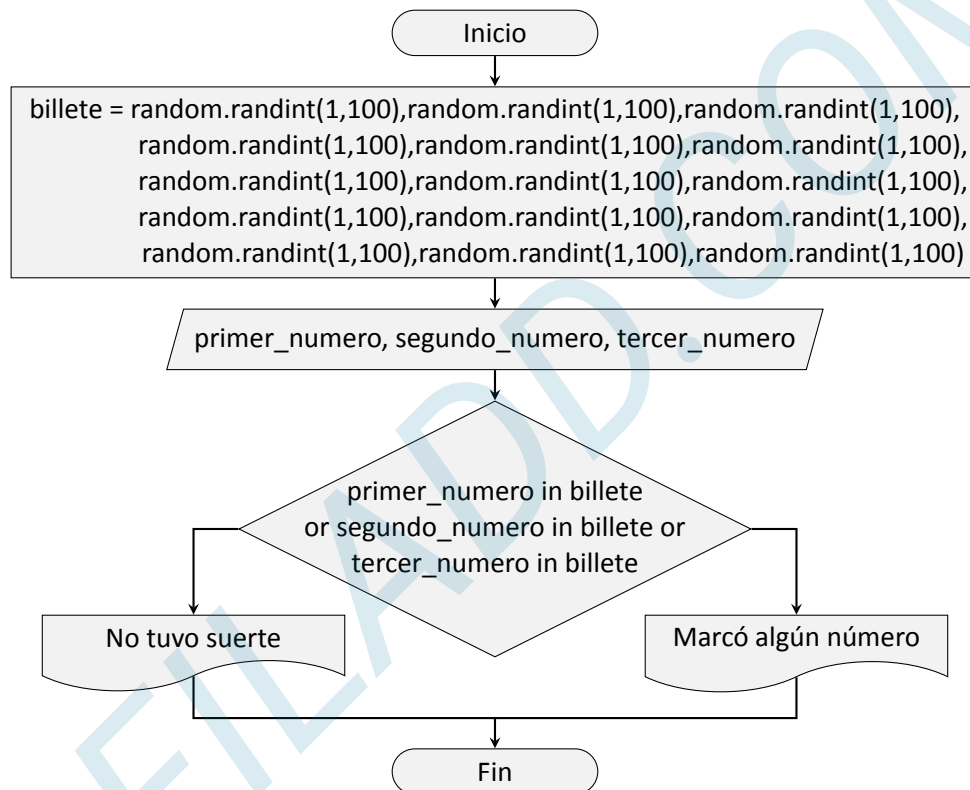
# Salida de resultados
print('El promedio de las temperaturas ingresadas fue ', promedio, ' grados')

if temp1 > promedio or temp2 > promedio or temp3 > promedio:
    print('Existe al menos una temperatura que supera al promedio')
else:
    print('Todas las temperaturas estan por debajo del promedio')
  
```

6 - Tarjeta de Bingo

Realizar un programa que genere 15 numero aleatorios enteros en el rango del 1 al 100, que representaria la tarjeta de bingo de una persona. Una vez generado los numeros aleatorios solicitar al usuario que ingrese 3 numeros enteros, a partir de alli mostrar los siguientes mensajes:

- Si el usuario no marco ninguno de los numeros indicarlo diciendo “El jugador tiene mala suerte, no marco ninguna casilla”.
- Caso contrario mostrar “El jugador marco algun numero de la tarjeta”.




```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

import random

print('Tarjeta de Bingo')
print("=" * 80)

print("\nCarga de Datos")
print('-' * 80)

billete = random.randint(1,100),random.randint(1,100),random.randint(1,100),\
    random.randint(1,100),random.randint(1,100),random.randint(1,100),\
    random.randint(1,100),random.randint(1,100),random.randint(1,100),\
    random.randint(1,100),random.randint(1,100),random.randint(1,100),\
    random.randint(1,100),random.randint(1,100),random.randint(1,100)

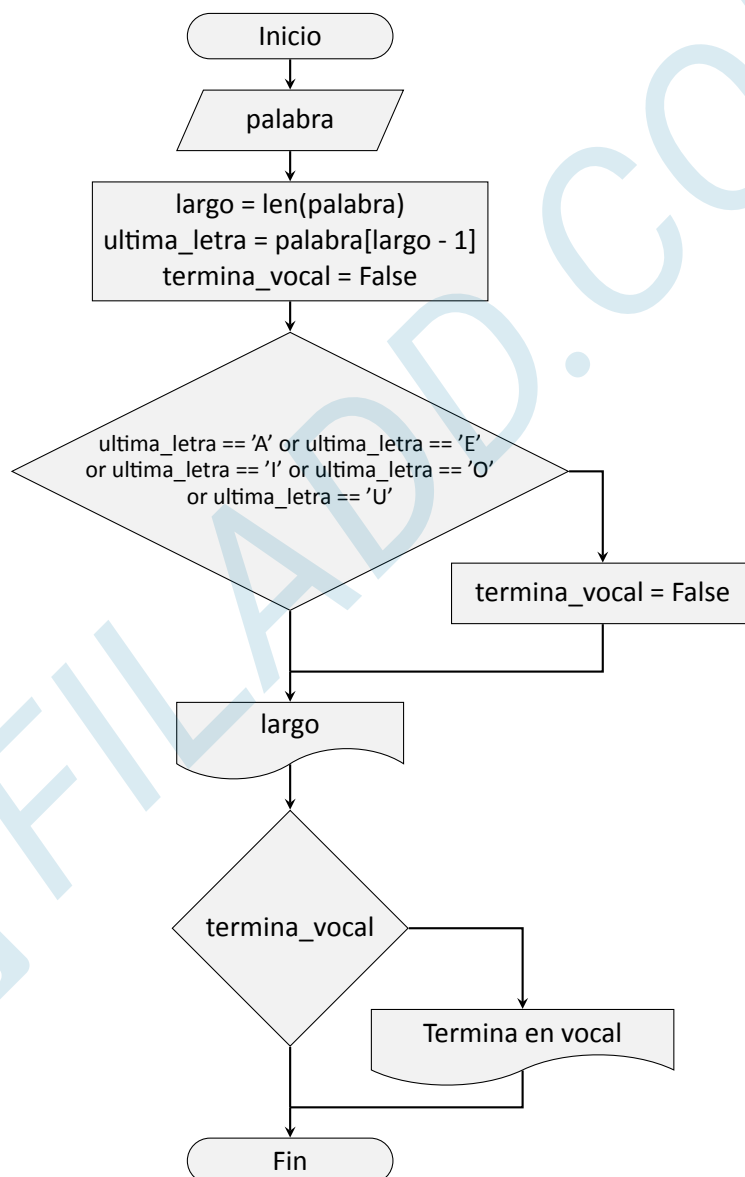
primer_numero = int(input('Ingrese la bolilla que salio (entre 1 y 100): '))
segundo_numero = int(input('Ingrese la bolilla que salio entre 1 y 100): '))
tercer_numero = int(input('Ingrese la bolilla que salio entre 1 y 100): '))

# Proceso y salida de resultados
print('La tarjeta del usuario es: ', billete)
if primer_numero in billete or \
    segundo_numero in billete or \
    tercer_numero in billete:
    print('El jugador marco algun numero de la tarjeta')
else:
    print('El jugador tiene mala suerte, no marco ninguna casilla')
```

7 - Análisis de palabra

Se pide un programa que le solicite al usuario que ingrese una palabra. Con esa palabra calcular los siguientes puntos:

- Determinar la cantidad de letras que tiene la palabra
- Mostrar un mensaje que informe si la palabra termina en vocal



```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

print('Análisis de Palabra')
print('=' * 80)

# Lectura de datos
palabra = input('Ingrese la palabra a analizar en mayúsculas: ')

# Proceso
largo = len(palabra)
ultima_letra = palabra[largo - 1]

termina_vocal = False
if ultima_letra == 'A' or ultima_letra == 'E' or ultima_letra == 'I' or \
    ultima_letra == 'O' or ultima_letra == 'U':
    termina_vocal = True

# Salida
print('La palabra ingresada tiene una longitud de ', largo, ' letras')
if termina_vocal:
    print('La palabra ingresada termina en vocal')
```

8 - Tirada de moneda

Programar una tirada de una moneda (opciones: cara o cruz) aleatoriamente. Permitir que un jugador apueste a cara o cruz y luego informar si acertó o no con su apuesta.

```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

import random

print('Tirada de la moneda')
print('=' * 80)

caras = 'cara', 'cruz'

apuesta = int(input('Selección que cara desea apostar (0 Cara 1 Cruz): '))
jugada = random.choice(caras)

if jugada == caras[apuesta]:
    print('El jugador ha ganado el juego, acerto, salio', jugada)
else:
    print('El jugador ha perdido el juego, " \
        "salio', jugada, 'y el jugador apostó a', caras[apuesta])
```

9 - Lanzamiento de dados

Simular un juego en el que se lanzan dos dados.

Si ambos dados son iguales o la suma entre ellos es impar, gana el usuario. En caso contrario, gana la máquina.

```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

import random

print("Juego de Dados\n")

dado1 = random.randrange(1, 7)
dado2 = random.randint(1, 6)

print("Dado 1:", dado1, "- Dado 2:", dado2)

suma = dado1 + dado2

if dado1 == dado2 or (suma%2) != 0:
    res = "Ganaste!"
else:
    res = "Perdiste!"

print("Resultado:", res)
```

10 - Edad mínima

Ingresar por teclado las edades de 3 participantes de un concurso.

Informar si todos cumplen con la edad mínima establecida para el mismo, también ingresada por teclado.

```
__author__ = 'Catedra de Algoritmos y Estructuras de Datos'

#Datos
edad1 = int(input('Participante 1 - Ingrese edad: '))
edad2 = int(input('Participante 2 - Ingrese edad: '))
edad3 = int(input('Participante 3 - Ingrese edad: '))

minimo = int(input('Ingrese edad minima para participar: '))

#Proceso y resultado
if edad1 >= minimo and edad2 >= minimo and edad3 >= minimo:
    print('TODOs los participantes cumplen con la edad mínima')
else:
    print('NO TODOs los participantes cumplen con la edad mínima')
```

11 - Terreno

Se ingresan las medidas de frente y fondo de un terreno.
Determinar si es cuadrado o rectangular y calcular su superficie.

```
fondo = float(input('Ingrese fondo del terreno (en metros): '))

#Proceso
if frente == fondo:
    forma = "CUADRADA"
else:
    forma = "RECTANGULAR"

superficie = round(frente * fondo,2)

#Resultados
print('El terreno tiene forma',forma,'y',superficie,'metros cuadrados de superficie')
```