

### **Bisiesto**

*Programa que ve si un año es bisiesto o no. Un año es bisiesto si es divisible por cuatro, con excepción de aquéllos que sean divisibles por 100 siempre que no sean divisibles por 400.*

```
año=int(input("Ingrese año: "))
```

```
if año%4==0:
```

```
    if año%100==0:
```

```
        if año%400==0:
```

```
            print("Si")
```

```
        else:
```

```
            print("No")
```

```
    else:
```

```
        print("Si")
```

```
else:
```

```
    print("No")
```

### **Camel Case**

*Contar cuantas palabras hay. Una nueva palabra empieza con mayúscula.*

```
frase=str(input())
```

```
c_palabras=0
```

```
for c in frase:
```

```
    if c in "QWERTYUIOPASDFGHJKLZXCVBNM":
```

```
        c_palabras+=1
```

```
print(c_palabras)
```

## Índice de Masa Corporal

*A partir de su estatura y peso, se debe determinar el IMC de una persona e indicar si está en un estado Normal, de Delgadez o de Sobrepeso.*

*Utilice la siguiente clasificación como referencia:*

- **Delgadez:**  $<18.50$
- **Normal:**  $18.50-24.99$
- **Sobrepeso:**  $\geq 25.00$

```
estatura=float(input())
```

```
peso=float(input())
```

```
imc=round(peso/(estatura**2),2)
```

```
print(imc)
```

```
if imc<18.50:
```

```
    print("Delgadez")
```

```
elif imc<25.00:
```

```
    print("Normal")
```

```
else:
```

```
    print("Sobrepeso")
```

### Viaje

*El programa debe calcular el costo total de un viaje en auto, considerando como entradas la distancia en Kms, el rendimiento del auto en Kms/litro y el costo de cada litro de combustible.*

```
d=float(input())
```

```
k=float(input())
```

```
c=float(input())
```

```
C=(d/k)*(c)
```

```
print(C)
```

### Tablas de Multiplicar

*Dado un número entero no negativo n, generar la tabla de multiplicar de n, para los factores desde el 0 hasta el 10.*

```
n=int(input())
```

```
i=0
```

```
while i<=10:
```

```
    a=n*i
```

```
    print(n,"x",i,"=",a)
```

```
    i=i+1
```

## Salario

*Una empresa aumenta un 10% a los empleados que ganan hasta \$500.000, un 7% a los que ganan más que eso pero que no superan \$1.000.000, y finalmente un 5% a los que ganan más que ese límite. El programa debe calcular el nuevo salario.*

```
s=float(input())
```

```
if s<=500000:
```

```
    s=s+(s*0.1)
```

```
elif s<=1000000:
```

```
    s=s+(s*0.07)
```

```
else:
```

```
    s=s+(s*0.05)
```

```
print(s)
```

## Dígitos

*Dado un número entero no negativo ingresado como entrada, contar cuántos dígitos tiene y calcular la suma de esos dígitos.*

```
a=str(input())
largo=len(a)
largo_Original=(largo)
suma=0
n=int(len(str(a)))-1
while largo>1:
    a=int(a)
    b=(a//(10**n))
    suma+=b
    a=a%(10**n)
    largo=len(str(a))
    n-=1
suma+=int(a)
print((largo_Original),str(suma))
```

### Factorial

*Dado un número entero  $N$  con  $(0 \leq N \leq 100 \leq N \leq 10)$ , usted debe encontrar e imprimir el factorial de ese número. El factorial de un entero positivo  $N$  se refiere al producto de todos los números en el rango de 1 a  $N$ . Por convención el factorial de 0 es 1.*

```
N=int(input())
```

```
suma=1
```

```
i=2
```

```
while i<=N:
```

```
    suma=suma*i
```

```
    i=i+1
```

```
print(suma)
```

### Novistar

*La empresa telefónica Novistar ha tenido un increíble aumento de clientes en este último año 2019, razón por la cual la empresa quiere premiar a aquellos clientes que llevan más de cinco años siendo parte de la comunidad. Por ello, la empresa Novistar ha guardado toda la información de sus usuarios en un diccionario (no vacío) llamado clientes, en el cual la llave es el número del usuario, y el valor es una tupla con el nombre, apellido, género y los años de servicio de un cliente.*

```
clientes = {
```

```
    123456789: ('Juan', 'Mendoza', 'M', 3),
```

```
    147258369: ('Jane', 'Smith', 'F', 8),
```

```
    145236987: ('Gonzalo', 'Sanchez', 'M', 0),
```

```
    125478963: ('Sarah', 'Jonhson', 'F', 2),
```

```
    951478632: ('Francisco', 'Omar', 'M', 7),
```

```

965478123: ('April', 'Stone', 'F', 13)}
numero=int(input())
_,_,años=clientes[numero]
if años>5:
    print(True)
else:
    print(False)

```

### Promedio

*Haga un programa que pida un número de notas, para luego ingresarlas una por una y calcule el promedio de estas.*

```

n_notas=int(input("Ingrese numero de notas: "))
total=0
for i in range(n_notas):
    nota=float(input("Ingrese nota: "))
    total+=nota
promedio=total/n_notas
print("El promedio es: ",promedio)

```

### Países

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 que dadas dos personas del diccionario, retorne el número de países que ambos han visitado.*

```
def cuantos_en_comun(persona_1, persona_2):  
    comunes=0  
    for pais in paises[persona_1]:  
        if pais in paises[persona_2]:  
            comunes+=1  
    return comunes
```

### **Restricción Vehicular**

*Pitonia tiene una alta congestión de vehículos circulando por sus calles. Las autoridades han decidido aplicar restricción vehicular para descongestionar la ciudad en base a las patentes de los vehículos.*

*La patente de un vehículo es un string de cuatro letras y dos dígitos, y la restricción depende sólo del penúltimo dígito. Por ejemplo, para la patente GEEA78, el dígito 7 es el utilizado para evaluar la restricción.*

*La restricción vehicular de los días hábiles de la semana se encuentra ingresada en el diccionario dígitos, cuyas llaves son los días de la semana, y cuyos valores son tuplas de cuatro enteros que representan los dígitos con restricción de ese día.*

- *Implemente una función que indique si el vehículo está o no con restricción.*

```
def esta_con_restriccion(digitos, dia, patente):  
    d1,d2,d3,d4=digitos[dia]  
    v=patente[-2]  
    if v==d1 or v==d2 or v==d3 or v==d4:  
        return True  
    else:
```



```
return False
```

- *Implemente una función que retorne una lista de los días en que el vehículo no puede circular.*

```
def dias_con_restriccion(digitos, patente):
```

```
    dias=list()
```

```
    v=int(patente[-2])
```

```
    for dia in digitos:
```

```
        if v in digitos[dia]:
```

```
            dias.append(dia)
```

```
    return días
```

- *Implemente una función que retorne una lista con los días en que el vehículo sí puede circular.*

```
def dias_sin_restriccion(digitos, patente):
```

```
    dias=dias_con_restriccion(digitos,patente)
```

```
    puedes=[]
```

```
    for llave in digitos:
```

```
        if llave not in dias:
```

```
            puedes.append(llave)
```

```
    return puedes
```

### **Asistencias a clases**

*En una escuela se lleva registro de las asistencias de los estudiantes a sus clases a través de una lista de listas. Cada lista dentro de la primera corresponde a la asistencia de un estudiante, donde “True” indica que asistió a esa clase y “False” que no.*

```
asistencia = [ [True, True, True, False, False, False, False],
```

```
                [True, True, True, False, True, False, True],
```

[True, True, True, True, True, True, True],

[True, True, True, False, True, True, True]]

- *Escriba una función que reciba como parámetro la tabla de asistencia y retorne una lista con el número de clases a las que asistió cada alumno.*

```
def total_por_alumno(tabla):
```

```
    asistencias=list()
```

```
    for alumno in tabla:
```

```
        registro=0
```

```
        for clase in alumno:
```

```
            if clase==True:
```

```
                registro+=1
```

```
        asistencias.append(registro)
```

```
    return asistencias
```

- *Escriba una función que reciba como parámetro la tabla de asistencia y retorne una lista con el número de alumnos que asistió a cada clase.*

```
def total_por_clase(tabla):
```

```
    asistencias=[0]*len(tabla[0])
```

```
    for dia in range(len(tabla[0])):
```

```
        for alumno in tabla:
```

```
            if alumno[dia]==True:
```

```
                asistencias[dia]+=1
```

```
    return asistencias
```

### **Puente**

*El puente py py es ruta más directa entre la ciudad de Pythonia y el campo.*

*El puente tiene un sistema controlador, el cual identifica los vehículos terrestres con un 0 y los acuáticos con un 1 y luego levanta o baja el puente*

*según corresponda. Además, muestra en la pantalla si el puente está subiendo o bajando.*

- *Escriba una función que permita al controlador levantar o bajar el puente dependiendo del vehículo que se acerque.*

```
def levantar_puente(tipo):
```

```
    if tipo==1:
```

```
        levantar=True
```

```
    else:
```

```
        levantar=False
```

```
    return levantar
```

- *Haga un programa que pregunte por el número de vehículos que pasarán por el puente. Con esto, deberá hacer un ciclo en el cual usar la función anterior para poder mover el puente hasta que pasen todos los vehículos.*

```
limite=int(input("Vehiculos que van a pasar: "))
```

```
i=0
```

```
while i<limite:
```

```
    tipo_vehiculo=int(input("Tipo de vehiculo: "))
```

```
    estado=levantar_puente(tipo_vehiculo)
```

```
    if estado==True:
```

```
        print("Levantando puente")
```

```
    if estado==False:
```

```
        print("Bajando puente")
```

```
    i+=1
```

## Tienda

*Hay una tienda en el campo que tiene un sistema el cual pregunta por el producto a comprar y su precio, también pregunta al cliente si desea algo más que llevar. Cuando el cliente no desea llevar nada más, el programa muestra el precio total que hay que pagar.*

```
producto=input("Nombre del producto: ")
precio_producto=int(input("Precio del producto: "))
total=0
flag=True
total+=precio_producto
algo_mas=input("¿Desea algo más?: [si/no] ")
while flag:
    if algo_mas=="si":
        producto=input("Nombre del producto: ")
        precio_producto=int(input("Precio del producto: "))
        total+=precio_producto
        algo_mas=input("¿Desea algo más?: [si/no] ")
    else:
        flag=False
print("El total a pagar es:",total)
```