



PROBLEMA

1. Generar los N términos de la siguiente serie: 3, 5, 8, 13, 21, 34, 55,...

ENTRADA	SALIDA
N=5	3, 5, 8, 13, 21

SOURCE IN PYTHON

```
# _____ function leer _____

def leer():
    n=int(input("Un valor="))
    return n

# _____ function fifonacci _____

def serie(x):
    a=1
    b=2
    if n<0:
        print("Incorrect input")
    elif n==0:
        print(0)
    elif n==1:
        print(a+b)
    else:
        for i in range(1,x+1):
            c=a+b
            a=b
            b=c
            print(b,end=', ')

# _____ function main _____

n=leer()
serie(n)
```

DESK CHECK

Un valor=5
3, 5, 8, 13, 21,

PROBLEMA

2. Utilizando programación modular definir un módulo que tome un número como entrada y determine si es positivo, negativo o cero.

ENTRADA	SALIDA
Introduzca un número 3	Positivo
Introduzca un número -6	Negativo

SOURCE IN PYTHON

```
# _____ function leer _____

def leer():
    n=int(input("Un valor="))
    return n

# _____ function verificar _____

def verifica(x):
    a=1
    b=2
    if n<0:
        print("Valor negativo")
    elif n==0:
        print("Valor Nulo")
    else:
        print("Valor Posetivo")

# _____ function main _____

n=leer()
verifica(n)
```

DESK CHECK

Un valor=1
Valor Posetivo

PROBLEMA

3. Realiza las funciones: suma, resta, multiplicación, división, potencia y raíz cuadrada.

ENTRADA	SALIDA
Introduzca un número a=3 Introduzca un número b=8	Suma=11 Resta=5 Multiplicación=24 División=0.375 Potencia=6561

SOURCE IN PYTHON

```
# _____ function leer _____

def leer():
    n=int(input("Un valor="))
    m=int(input("Un valor="))
    return n,m

# _____ function suma, resta.... _____

def suma(x,y):
    return x+y
def resta(x,y):
    return x-y
def multiplica(x,y):
    return x*y
def division(x,y):
    if y == 0:
        print(" Math Eerror")
    else:
        return x/y
def raizCua(x,y):
    return (x**(0.5)),(y**(0.5))
def potencia(x,y):
    return x**y
def mostrar(x,y):
    print (x,y)

# _____ function main _____

n,m = leer()
res = suma(n,m)
mostrar(" suma:",res)
res = resta(n,m)
mostrar(" Resta:",res)
res = multiplica(n,m)
mostrar(" multiplicacion:",res)
res = division(n,m)
mostrar(" division:",res)
res = potencia(n,m)
mostrar(" potencia:",res)

res1,res = raizCua(n,m)
mostrar( f" raiz={n}:",res1)
mostrar( f" raiz={m}:",res)
```

DESK CHECK

Un valor=3
Un valor=8
suma: 11
Resta: -5
multiplicacion: 24
division: 0.375
potencia: 6561
raiz=3: 1.7320508075688772
raiz=8: 2.8284271247461903



PROBLEMA

4. Definir un módulo que imprima la tabla de multiplicar del 1 al 10, de un número dado.

ENTRADA	SALIDA
Introduzca un número x=4	1 x 4 = 4 2 x 4 = 8 3 x 4 = 12 4 x 4 = 16 5 x 4 = 20

```
#_____ function leer_____
def leer():
    n=int(input("Un valor="))
    return n

#_____ function generar tabla...._____
def tabla(x):
    for i in range(1,11):
        print(f"{i}*{n}={i*n}")

#_____ function main_____
n = leer()
tabla(n)
```

DESK CHECK

```
Un valor=4
1*4= 4
2*4= 8
3*4= 12
4*4= 16
5*4= 20
6*4= 24
7*4= 28
8*4= 32
9*4= 36
10*4= 40
```



PROBLEMA

5. Introducir tres números enteros positivos, calcule el factorial de cada número introducido y halle la suma de los factoriales.

ENTRADA	SALIDA
Introduzca un número a=1 Introduzca un número b=3 Introduzca un número c=5	Factorial de a: 0 Factorial de a: 6 Factorial de a: 120 Suma=126

SOURCE IN PYTHON

```
#_____ function leer_____
def leer():
    x = int(input("Un valor="))
    y = int(input("Un valor="))
    z = int(input("Un valor="))
    return x,y,z

#_____ function generar tabla...._____
def facto(x):
    fa=1
    for i in range(1,x+1):
        fa = fa*i
    return fa

def mostrar(x,y):
    print (x,y)

#_____ function main_____
res=0
suma = 0
x,y,z = leer()
res = facto(x)
mostrar(f"factorial={x}",facto(x))
suma = suma+ res

res = facto(y)
mostrar(f"factorial={y}",facto(y))

suma =suma+ res
res = facto(z)
mostrar(f"factorial={z}",facto(z))
suma = suma+ res
print("la suma=",suma)
```

DESK CHECK

```
Un valor=1
Un valor=3
Un valor=5
factorial=1 1
factorial=3 6
factorial=5 120
la suma= 127
```





PROBLEMA

6. Introducir tres números enteros positivos, calcule la suma de los números primos.

ENTRADA	SALIDA
Introduzca un número a=9 Introduzca un número b=5 Introduzca un número c=11	Suma=16

SOURCE IN PYTHON

```
#_____ function leer_____

def leer():
    n = int(input("Un valor="))
    return n

#_____ function generarar primo...._____

def primos():
    conta=0
    suma =0
    while 3>conta:
        n = leer()
        c=0
        for i in range(1,n+1):
            if n % i==0:
                c = c + 1
        if c==2:
            #return True
            suma=suma + n
            conta = conta + 1
    return suma

#_____ function main_____
r=primos()
print("la suma es =", r)
```

DESK CHECK

```
Un valor=5
Un valor=11
Un valor=9
la suma es = 16
```

PROBLEMA

7. Realice la función, para calcular el valor de aproximado de π , según la serie:

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} + \dots$$

ENTRADA	SALIDA
Introduzca un número impar n=5	resultado=3.3396

SOURCE IN PYTHON

```
#_____ function leer_____

def leer():
    n = int(input("Un valor="))
    return n

#_____ function generar ...._____

def pi():
    signo= (-1)*1
    cons=4
    conta=0
    suma =0
    impares=1
    n = leer()
    while n>conta:
        #print(impares)
        #print(f"{cons}/{impares}")
        if conta %2==0:
            suma=suma+(cons/impares)
        else:
            suma=suma -(cons/impares)
        impares= impares+2
        conta =conta +1
    return suma

#_____ function main_____
r = pi()
print("el pi es =", r)
```

DESK CHECK

```
Un valor=5
el pi es = 3.3396825396825403
```





PROBLEMA

8. Realice las funciones necesarias para calcular los N términos de la siguiente sucesión de términos:

$$\frac{2^3}{2} - \frac{3^4}{3} + \frac{5^5}{2} - \frac{8^6}{3} + \frac{12^7}{2} - \frac{17^8}{3} + \dots$$

ENTRADA	SALIDA
Introduzca un número n=3	resultado=1539.5

SOURCE IN PYTHON

```
# _____ function leer _____

def leer():
    n = int(input("Un valor="))
    return n

# _____ function generar ...._____

def sumar():
    expo=3
    cons1=2
    conta=0
    suma=0
    n = leer()
    a=1
    b=1
    c=0
    while n>conta:
        c=a+b
        a=b
        b=c
        if cons1 %2==0:
            suma=suma+(b**expo/cons1)
            cons1=cons1+1
        else:
            suma=suma+(b**expo/cons1)
            cons1=cons1-1

        expo =expo+1
        conta =conta +1

    return suma

# _____ function main _____
r = sumar()
print("la suma es=", r)
```

DESK CHECK

```
Un valor=3
el pi es = 1593.5
```

DESK CHECK

```
Un valor=12
1,2,3,4,6,12,La suma es: 28
```

PROBLEMA

9. Realice las funciones necesarias (mínimo 3), para obtener el resultado de la siguiente ecuación.

$$res = \sqrt{(a^3 + b^3 + c^3)}$$

ENTRADA	SALIDA
Introduzca un número a=1	res=11.5758
Introduzca un número b=2	
Introduzca un número c=5	

SOURCE IN PYTHON

```
# _____ function leer _____

def leer():
    n = int(input("Un valor="))
    m = int(input("Un valor="))
    nn=int(input("Un valor="))
    return n,m,nn

# _____ function generar ..._____

def cubo(a,b,c):
    a1 = a*a*a
    b1 = b*b*b
    c1 = c*c*c
    raiz( a,b1,c1)

def raiz( a,b,c):
    suma=0
    suma= (a+b+c)**0.5
    print("la raiz",suma)

# _____ function main _____
x,y,z = leer()
cubo(x,y,z)
```

DESK CHECK

```
Un valor=1
Un valor=2
Un valor=5
la raiz 11.575836902790225
```

PROBLEMA

10. Leer un numero entero positivo y determinar la suma de sus divisores.

ENTRADA	SALIDA
Introduzca un número n=12	Divisores: 1, 2, 3, 4, 6, 12 Suma: 28

SOURCE IN PYTHON

```
# _____ function leer _____

def leer():
    n = int(input("Un valor="))

    return n

# _____ function generar..._____

def divisores(n):
    suma= 0
    for i in range(1,n+1):
        if n % i==0:
            print(i,end=",")
            suma=suma+i

    return suma

def mostrar(a,b):
    print(a,b)

# _____ function main _____
x = leer()
r=divisores(x)
mostrar("La suma es:",r)
```



PROBLEMA

11 Leer un numero entero positivo a y determinar la suma de sus n múltiplos.

ENTRADA	SALIDA
Introduzca un número a=5 Cúantos múltiplos desea sumar ? n=3	Multiplos de a: 5,10,15 Suma: 20

SOURCE IN PYTHON

```
# _____ function leer _____
def leer():
    n = int(input("Un valor="))
    return n
# _____ function generar... _____
def multiplo(n):
    suma= 0
    mult=0
    for i in range(1,n+1):
        print(i*5,end=";")
        mult=mult+(i*5)
        suma=suma+mult
    return mult
def mostrar(a,b):
    print(a,b)
# _____ function main _____
n = leer()
r =multiplo(n)
mostrar("La sumaes:",r)
```

DESK CHECK

```
Un valor=3
5,10,15,La sumaes: 30
```

PROBLEMA

12. Utilizando programación modular, determinar si un número es palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda).

ENTRADA	SALIDA
Introduzca un número n=12321	Es un palíndromo
Introduzca un número n=35023	No es un palíndromo

SOURCE IN PYTHON

```
# _____ function leer _____
def leer():
    n = int(input("Un valor="))
    return n
# _____ function generar... _____
def palindromo(n):
    nn = n
    res = 0
    while n > 0:
        dig = n % 10
        res = (res * 10) + dig
        n = n // 10
    if nn == res:
        print(f"El número {nn} es un palíndromo. =={res}")
    else:
        print(f"El número {nn} no es un palíndromo. =={res}")
# _____ function main _____
n= leer()
palindromo(n)
```

DESK CHECK

```
Un valor=12321
El número 12321 es un palíndromo. ==12321
```

PROBLEMA

13. Utilizando programación modular, descomponer un número entero en sus dígitos y luego Calcular la suma de los dígitos.

ENTRADA	SALIDA
Introduzca un número n=841	La suma de sus dígitos es 13

SOURCE IN PYTHON

```
# _____ function leer _____
def leer():
    n = int(input("Un valor="))
    return n
# _____ function generar... _____
def sumaDig(n):
    res = 0
    while n > 0:
        dig = n % 10
        res = res + dig
        print(dig,end=";")
        n = n // 10
    print(f"la suma es= {res}")
# _____ function main _____
n= leer()
sumaDig(n)
```

DESK CHECK

```
Un valor=841
1,4,8,la suma es= 13
```



PROBLEMA

14. Leer 3 números que sean mayores a cero y conformar un nuevo número con los dígitos que sean múltiplos de 3 y

ENTRADA	SALIDA
Introduzca un número a=3168	res=3696
Introduzca un número b=9214	
Introduzca un número c=126	

SOURCE IN PYTHON

```
#_____function leer_____
def leer():
    n = int(input("Un valor="))
    return n

#_____function generar..._____
def sumaDig():

    res = 0
    contar = 1
    while 3>=contar: # pide 3 numero verificamos
        n = leer()
        if n>0:
            while n > 0:
                dig = n % 10
                if dig % 3==0:
                    res = (res * 10) + dig
                    print(dig,end = ",")
                    n = n // 10
                print (f"la suma es= {res}")
            else:
                print ( "el numero debe ser mayor de cero")
                contar = contar + 1
        else:
            print ( "el numero debe ser mayor de cero")
            contar = contar + 1

#_____function main_____
sumaDig()
```

DESK CHECK

```
Un valor=3168
8,6,1,3,la suma es= 63
Un valor=9214
4,1,2,9,la suma es= 639
Un valor=126
6,2,1,la suma es= 6396
```

PROBLEMA

15. Leer dos números que tengan 2 o más dígitos necesariamente, a y b, valide la entrada, y busque la cantidad de veces que el primer dígito de a se encuentre en b y muestre el resultado obtenido.

ENTRADA	SALIDA
Introduzca un número a=653	El dígito 6 se encuentra 2 veces en 3646
Introduzca un número b=3646	

SOURCE IN PYTHON

```
#_____función leer _____
def leer():
    n = int(input("Un valor="))
    return n

#_____función _____
def sumaDig():
    contarIgual = 0

    n = leer()
    nOriginal = n
    while nOriginal >= 10:
        nOriginal = nOriginal // 10
        primer_digito = nOriginal
        print(f"El primer dígito es: {primer_digito}")

    digito = leer()
    nAux = digito

    while digito > 0:
        nn = digito % 10
        if nn == primer_digito:
            contarIgual = contarIgual + 1
        digito = digito // 10

    print(f"El dígito {primer_digito} se encuentra {contarIgual} veces en {nAux}")

#_____función main _____
sumaDig()
```

DESK CHECK

```
Un valor=653
El primer dígito es: 6
Un valor=3646
El dígito 6 se encuentra 2 veces en 3646
```





VECTORES

PROBLEMA

16. Mostrar la suma de los números primos del vector.
Determinar si la suma es primo o no.

SOURCE IN PYTHON

```
def leer():
    n = int(input("Un valor="))
    return n

#_____ function para generar vector _____
def llenar(n):
    V = []
    for i in range(n):
        x=leer()
        V.append(x)
    return V

def verPrimo(V,n):
    suma = 0
    c=0
    for i in range(n):
        primo = V[i]
        c=0
        for j in range(1,primo+1):
            if primo % j == 0:
                c = c + 1
        if c == 2:
            suma = suma + primo
        #V[i] = suma
    return suma

def mostrar(V, n):
    for i in range(n):
        print(V[i],end=";")
    print()

def verSumaPrimo(nvector):
    c=0
    for j in range(1,nvector +1):
        if nvector % j == 0:
            c = c + 1
    if c == 2:
        print(f"La suma es un numero primo")
    else:
        print(f"La suma no es un numero primo")

#_____ function main _____

print("dimension de vector")
n = leer()
print("Ingrese valores para vector : ")
vector = llenar(n)
nvector = verPrimo(vector,n)
print("la suma de los primos en el vector es : ",nvector)

verSumaPrimo(nvector)

mostrar(vector,n)
```

ENTRADA	SALIDA
Ingrese la dimensión del vector: 5	Primos: 3,3,7
Ingrese los elementos:	Suma: 13
3	"La suma es un número primo"
4	
3	
7	
2	

NOTA: la suma total es
15 no 13 ya que el
número 2 es primo
 $\Rightarrow 3+3+7+2 = 15$

DESK CHECK

```
dimension de vector
Un valor=5
Ingrese valores para vector :
Un valor=3
Un valor=4
Un valor=3
Un valor=7
Un valor=2
la suma de los primos en el vector es : 15
La suma no es un numero primo
- - - - -
```

PROBLEMA

17 2. Reemplazar los valores de un vector por la suma de los dígitos de cada valor. mostrar el nuevo número que se conformó,

ENTRADA	SALIDA
Ingrese la dimensión del vector: 5	
Ingrese los elementos:	Vector inicial: 33, 4, 13, 7, 20
33	
4	
13	
7	
20	Vector final: 6, 4, 4, 7, 2

SOURCE IN PYTHON

```
#def leer():
    n = int(input("Un valor="))
    return n

#_____ function para generar vector _____
def llenar(n):
    V = []
    for i in range(0, n):
        x=leer()
        V.append(x)
    return V

def primoVer(V,n):
    suma=0
    var=0
    for i in range(0,n):
        #
        V[i]+=30
        suma=0
        var=0
        var=V[i]
        while var>0:
            nn=var%10
            suma=suma+nn
            var=var//10
        V[i]=suma
    return V

def mostrar(V, n):
    for i in range(0,n):
        print(V[i],end=" ")
    print()

#_____ function main _____

print("dimension de vector")
n=leer()
print("Ingrese valores para vector : ")
vector=llenar(n)
nvector=primoVer(vector,n)
print("la suma del nuevo vectors en cada posicion : ")
mostrar(nvector,n)
```

DESK CHECK

```
diminsion de vector
Un valor=5
Ingrese valores para vector :
Un valor=33
Un valor=4
Un valor=13
Un valor=7
Un valor=20
la suma del nuevo vectors en cada posicion :
6,4,4,7,2,
```




PROBLEMA

18.Utilizando programación modular realizar un programa para leer N elementos en un vector de números enteros. Luego se pide imprimir dichos elementos y finalmente duplicar los elementos de los subíndices impares.mostrar el nuevo número que se conformó,

ENTRADA	SALIDA
Ingrese la dimensión del vector: 5	
Ingrese los elementos:	Vector inicial: 3, 1, 3, 7, 2
3	
1	
3	
7	
2	Vector final: 3, 2, 3, 14, 2

SOURCE IN PYTHON

```
def leer():
    n = int(input("Un valor="))
    return n

#_____ function para generar vector _____
def llenar(n):
    V = []
    for i in range(n):
        x=leer()
        V.append(x)
    return V

def duplicar(V,n):
    suma=0
    for i in range(n):
        if i % 2 == 0:
            pass
        else:
            doble=V[i]
            suma=doble+doble
            V[i]=suma

    return V

def mostrar(V, n):
    for i in range(n):
        print(V[i],end=";")
    print()

#_____ function main _____

print("dimension de vector")
n = leer()
print("Ingrese valores para vector : ")
vector = llenar(n)
nvector = duplicar(vector,n)
print("el doble del posiciones imparessdel vector : ")
mostrar(nvector,n)
```

DESK CHECK

```
diminsion de vector
Un valor=5
Ingrese valores para vector :
Un valor=3
Un valor=1
Un valor=3
Un valor=7
Un valor=2
el doble del posiciones imparessdel vector :
3,2,3,14,2,
```



PROBLEMA

19. Cargar un vector V con N elementos y generar un vector W con el cálculo del factorial del último dígito de cada elemento de V. Mostrar el vector W generado.

ENTRADA		SALIDA	
Ingrese la dimensión del vector: 6			
Ingrese los elementos:			
V	23 10 5 146 93 32	W	6 1 120 720 6 2
=		=	

SOURCE IN PYTHON

```
def leer():
    n = int(input("Un valor="))
    return n

#_____ function para generar vector _____
def llenar(n):
    V = []
    for i in range(n):
        x=leer()
        V.append(x)
    return V

def factorial(V,n):
    suma = 0
    for i in range(n):

        fact = V[i]

        fa =1
        res= fact % 10

        for j in range(1,res+1):
            fa = fa *j
        V[i] = fa

    return V
def mostrar(V, n):
    for i in range(n):
        print(V[i],end=";")
    print()

#_____ function main _____

print("dimension de vector")
n = leer()
print("Ingrese valores para vector : ")
vector = llenar(n)
nvector = factorial(vector,n)
print("el factorial de l ultimo digto es : ")
mostrar(nvector,n)
```

DESK CHECK

```
dimension de vector
Un valor=6
Ingrese valores para vector :
Un valor=23
Un valor=10
Un valor=5
Un valor=146
Un valor=93
Un valor=32
el factorial de l ultimo digto es :
6,1,120,720,6,2,
```



PRÁCTICA GENERAL I

MATERIA: PROGRAMACIÓN I SIS-211
NOMBRE: PORFIEIO ELIAS QUISPE QUISPE
PARALELO: "2A"
TURNO : MAÑANA
SEMESTRE: SEGUNDO
DOCENTE: ING. KATYA PEREZ M.

GESTION II-2024

UNIVERSIDAD PÚBLICA DE EL ALTO