

NOTAS PARA ESTRUCTURA :D

Este es un repaso donde voy a poner mis códigos hechos en estos días para no ir sin nada jejeje

Es que ya me puse a estudiar porfin wuuuuuu :D

ESTE CODIGO CONSISTE EN CONTAR LOS DIGITOS SIN CONTAR LOS CEROS QUE ESTEN EN LA IZQUIERDA

```
c = str(input())
c2 = c.lstrip('0') #lstrip borra los digitos de la izquierda
print(len(c2))
```

ESTE CODIGO IMITA LA FUNCION XOR

```
def xor(n,n2):
    if n == 1 and n2 == 0:
        return print(True)
    elif n == 0 and n2 == 1:
        return print(True)
    elif n == 1 and n2 == 1:
        return print(False)
    elif n == 0 and n2 == 0:
        return print(True)

if __name__=='__main__':
    n = int(input())
    n2 = int(input())

    xor(n,n2)

    '''compara los casos, si es 11 es falso,
    si es 10,01,00 es verdadero'''
```

ESTE CODIGO ENSEÑA EL NUMERO MAYOR DE UNA LISTA

```
def mayorlista (n,lista,mayor):
    for i in range(n):
        i = i+1
        lista.append(int(input())) #se agregan los numeros a la lista

    for n in range(0,len(lista)): #va desde el primer al ultimo
        if lista[n] > mayor: #si el primero numero agregado es mayor al
anterior se mete
            mayor = lista[n] #aqui lo mismo

    print (mayor)

if __name__ == '__main__':

    n = int(input()) #determina el tamaño de la lista
    lista = []
    mayor = 0
    mayorlista(n,lista,mayor)
```

ESTE CODIGO MULTIPLICA TODOS LOS NUMEROS AGREGADOS

```
def producto(p,n,multi):

    for i in range(n): #termina en el numero max de agregar
        p.append(int(input())) #se agregan a la lista
        multi = multi* p[i] #i es el numero que va siendo agregado el cual
se multiplica por el anterior agregado a el
    print (multi)

if __name__=='__main__':
    p = []
    multi = 1
    n = int(input()) #cuantos numeros se van a agregar

    producto(p,n,multi)
```

ESTE CODIGO PONE EN MAYUSCULAS LA PRIMERA LETRA DE LOS NOMBRES EXCEPTO SI ES “DE o LA”

```
prohibido = ["de", "la"] #las palabras que no se deben de hacer mayus
c = [] # donde se entrega todo
t = str(input())

t2 = t.split() #se divide en partes el texto

for i in t2:
    if i in prohibido: # si i tiene las palabras de PROHIBIDO
        c.append(i) #entonces se agrega i a la lista c como esta
    else:
        c.append(i.capitalize()) #i capitaliza las palabras de prohibido y
se agrega a c

c2= ' '.join(c) #c se une con un espacio a c2 para volverse cadena

print (c2)
```

CODIGO DE UN PALINDROMO

```
def pal (p,c1,c2):

    for i in p:

        c1 = c1+i # si la variable la pongo a la izquierda se pasa normal

    for i in p:
        c2 = i + c2 #si la variable la pongo a la derecha se invierte

    if c1 == c2: #compara las variables
        print(True)
    else:
        print(False)

if __name__=='__main__':
    c1= str()
    c2= str()
    p = str(input())
# variables utilizadas

    pal (p,c1,c2) # manda a llamar a la función
```

ESTE CODIGO AGREGA EL PRIMER VALOR DE UNA LISTA, DESPUES LE PRIMER DE LA SIGUIENTE Y ASI EN UNA TERCERA LISTA DONDE SE CONCATENAN LOS 2

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

n1 = []
n2 = []
n3 = []
a = 0 #se declara en 0 para que sean acumuladores mas adelante
b = 0 #se declara en 0 para que sean acumuladores mas adelante
for i in range(n):
    n1.append(int(input())) #agrega los valores a las listas

for i in range(m):
    n2.append(int(input())) #agrega los valores a las listas

while a < len(n1) or b < len(n2): #mientras a sea menor a len n1 o b sea
    menor a len n2
    if a < len(n1):
        n3.append(n1[a]) #se agrega el primer valor (seria 0 primeramente )
        a = a+1 #se suma para que ese 0 se vuelva uno y asi arriba se
    agregue ahora el valor 1 y asi hasta que llegue al final
    if b < len(n2):
        n3.append(n2[b]) #cuando se agrego el a pasó a ser 1, entonces como
    b sigue siendo 0 se agrega este de segundas
        b =b+1 #se suma por 1 para volverse uno y asi hasta llegar al final
    de su lista

print(n3)
```

CODIGOS DE RECURSIVIDAD

```
'''  
  
def ejemploFAC(n):  
    r=1  
    i=2  
    while i <=n:  
        r *= i  
        i += 1  
    return print(r)  
  
if __name__=='__main__':  
    n = int(input())  
    ejemploFAC(n)  
  
'''  
'''  
  
def ejemplo2(n):  
  
    if n==1:  
        return 1  
    else:  
  
        return n * ejemplo2(n-1)  
#lo que hace esto es que si 5 es diferente a 1 entonces  
#se multiplica por su anterior hasta llegar a 1  
  
if __name__=='__main__':  
    n = int(input())  
    print(ejemplo2(n)) #retorna el resultado  
  
'''  
  
def impares(n):  
  
    if n == 0:  
        return 0  
    else:  
        if n % 2 != 0: # esto identifica los que no son pares  
            return n + impares(n - 1) # aqui suma los numeros impares para  
mostrar el resultado  
        else:  
            return impares(n - 1) #aqui llama al siguiente impar
```

```

if __name__=='__main__':

    n = int(input())

    print (impares(n))

```

CODIGO QUE ENCUENTRA LA MITAD DE UNA CADENA

```

def mitad(c,c2,md,md2): #funcion

    c2 = list(c) #convierte la cadena C en una lista

    if len(c2)%2 == 0: # si el numero de carcteres entre 2 da de residuo 0
se ejecuta
        md=len(c2)//2 #se divide el n. de carcteres, y como tiene '/' no
deja decimal
        md2= c2[md-1]+c2[md] #se meten a una cadena
        print (md2)

    elif len(c2)%2 != 0: #si el residuo es diferente de 0 se ejecuta
        ''!= significa diferente''
        md=len(c2)//2
        md2 = c2[md]
        print (md2)

if __name__=='__main__':
    c = str(input())
    c2 = str()
    md = 0
    md2 = ""
    mitad(c,c2,md,md2)

```

MATRICEZ

```

n = int(input())
c2 = []
c3 = str

```

```

for i in range (n): #las filas de la matriz
    c = [] #donde se va a aguardar
    for j in range (n): #columnas
        c.append(int(input())) #se agregan los datos
    c2.append(c) #se meten los datos a otra lista

if len(c)%2 != 0: #si la matriz es impar entonces se consigue el numero de
en medio

    m = len(c)//2
    n = len(c)//2
    c3 = c2[m][n] #se vuelve str
    print (c3)

```

CONCATENA 2 LISTAS Y LAS ORDENA DE MENOR A MAYOR

```

def unirL(l1,l2,l3,n1,n2):
    for i in range(n1):
        l1.append(int(input())) #se definen los numeros de la lista 1

    for i in range (n2):
        l2.append(int(input())) #se definen los numeros de la lista 2

    l3 = l1+l2

    for i in range (len(l3)): #primer for para recorrer la lista de inicio
a fin
        for j in range (0,len(l3)-i -1): #segundo for para ordenar, se usa
el -i -1 para evitar comparar los ya acomodados
            if l3[j]> l3[j+1]: # se comparan
                l3[j], l3[j + 1] = l3[j + 1], l3[j] #se determina el mayor

    print(l3)

if __name__=='__main__':
    l1 = []
    l2 = []
    l3 = []

    n1 = int(input())
    n2 = int(input())
    unirL(l1,l2,l3,n1,n2)

```

CALCULA EL NUMERO QUE TU PIDES SUMANDO LOS NUMEROS ANTERIORES HASTA LLEGAR A ESE RESULTADO

```
'''def infinito(inf, n, r):
    r = inf[n-1]
    return r

if __name__=='__main__':
    inf= [1, 1, 2, 3, 5, 8, 13, 21]
    n = int(input())
    r = 0

    infinito(inf,n,r)'''

def infinito(inf, n, r, n2):
    for i in range(2, n): # Inicia en el tercer número y va hasta el n-ésimo
número
        r = inf[i - 1] + inf[i - 2] # Calcula la suma de los dos números
anteriores
        inf.append(r) # Añade el resultado a la lista
        print(inf[n2]) # Imprime el término n2 de la lista

if __name__ == '__main__':
    r = 0
    inf = [1, 1] # Inicializa la lista con los dos primeros términos de
Fibonacci
    n = int(input()) # Lee el valor de n del usuario
    n2 = n - 1 # Ajusta para obtener el índice correcto
    infinito(inf, n, r, n2) # Llama a la función
```