

$$\cos x \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

$$x = \pi/6 = 0,52$$

$$n = 3$$

def main ():

    print ("Hola...")

    n = int (input ("Dame la cantidad de términos: "))

    if (n <= 0):

        print ("No te puedo ayudar n>0")

    else:

        x = float (input ("Dame el valor de x: "))

        cont = 1

        sumatoria = 0

        par = 0

        while (cont <= n):

            Calculo  
            Factorial

            { contF = 1  
            n\_fact = 1  
            while (contF <= par):  
                n\_fact = n\_fact \* contF  
                contF = contF + 1  
            if (cont % 2 != 0):

                sumatoria = sumatoria + (x \*\* par / n\_fact)

            else:

                sumatoria = sumatoria - (x \*\* par / n\_fact)

            cont = cont + 1

            par = par + 2

        print ("cos (" , x , ") = " , sumatoria)

        print ("para" , n , "términos de la ST")

        print ("Hasta pronto")

main()

```

def main():
    print("Hola... ")
    c_empleados
    if (c_empleados

```

Variable	Tipo	Valor
F	i	-2 - 10 / 1 - 2 + (-2) <sup>3</sup>
F	F	1 ✓ - 2 - 8
acu	acu	0 - 8 - 9 - 9 - 1% 2 == 0
		-10 (2) 00,5
		-10 + (-1) <sup>3</sup>
		- 1 - 9 + (0)
		- 9 + (1)
		(-8)
Ua	T	V
a		2 9.. - 32 1050
b		3
c		5 - 9 - 21 - 21 1029 - 21
d		4 8 - 50

X " " "

Nombre: Maricá Alejandra Vival Ocuante

Fecha: 7 de Febrero

Descripción: Prueba corta

} Si es el rotulo de  
la plantilla faltaron los  
comillitos pro 2

" " "

def main():

# Saludo

print ("Hola, te ayudaré a saber si el alfíl puede  
pasar de la primer casilla a la segunda en  
un solo movimiento")

print ("Te voy a solicitar 4 números, donde los 2  
primeros indican la posición inicial y los otros  
2 la final.")

# Toma de datos

c1 = int(input("Digita el número de la columna  
inicial: "))

f1 = int(input("Digita el número de la fila inicial: "))

c2 = int(input("Digita el número de la columna final: "))

f2 = int(input("Digita el número de la fila final: "))

# Condicional

if (c1 and f1 and c2 and f2 <= 8 and > 0):

X faltó el llomado de la función main()

```

def Main():
    print ("Bienvenidos... ")
    print ("Es ayudare... ")
    c1 = int(input("c1: "))
    f1 = int(input("f1: "))
    c2 = int(input("c2: "))
    f2 = int(input("f2: "))
    dy = c2 - c1
    dx = f2 - f1

    if (1 <= c2 <= 8 and 1 <= c1 <= 8 and 1 <= f2 <= 8 and 1 <= f1 <= 8):
        if (dy < 0):
            dy = dy * -1
        if (dx < 0):
            dx = dx * -1
        if (dy == dx):
            print ("SI")
        else:
            print ("NO")
    else:
        print ("Error")
    print ("Muchas gracias")

```

main()

Problema 2:

" " "

```

def main():
    # Saludo
    print ("Bueno... ")
    print ("Sección par e impar")
    # petición de datos
    n = int(input("Dame el valor de n: "))
    # condicional

```

```
if (n % 2 == 0):
```

```
    print ("par")
```

```
else:
```

```
    print ("impar")
```

```
while (n >= 1):
```

```
    print (n ** 2)
```

```
n = n - 2      # También es n -= 2  
                válido
```

```
print ("Gracias")
```

```
main ()
```

```
cont = 1
```

```
while (cont <= n // 2):
```

```
    print (cont)
```

```
    cont = cont + 1
```

```
    print (n)
```

```
    n = n - 1
```

variable	tipo	valor

Nombre: Marí Alejandra Uvias y Luisa Mayerli Mora<sup>①</sup>

Fecha: 14 de febrero

Descripción: Prueba corta

```
/def main ():
```

```
    # Saludo
```

```
    print ("Hola, te ayudaré a calcular la secuencia")
```

```
    # Toma de datos
```

```
    n = int (input ("Dame un numero mayor a cero: "))
```

```
    if (n <= 0):
```

```
        print ("Error")
```

```
    else:
```

```
        cont = 1
```

```
        while (cont <= n):
```

```
            print (cont)
```

```
            if (cont % 3 == 0):
```

```
                print (cont + 3) ??
```

```
            else:
```

```
                print (cont * 2) ??
```

? Por que !=

el algoritmo se quedó en un ciclo infinito  
nunca se actualiza el valor de cont

ej: n = 2

```
        print ("Fue un placer ayudarte.")
```

```
main ()
```

Puntilla

1

4

1

4

1

4

Ciclo inf....

marril

de dato

Variable	tipo	valor
mensaje	caracter (str)	Hola AYPR

print (type (mensaje))

variable

Numeros con punto No comas

○ galon bajo para separar  
ejemplo      suma - numeros

todo en mayuscula

Erros → bugs

Debuguear o debugging → proceso de actualización  
del programa para que no  
tenga errores inesperados

Todos los archivos se guardan con la extensión (-py)

valor quemado o explícito → Herencia

Cálculo

print (200/95)  
↓      ↓  
espacios

→ con un slash (/) y  
el division real  
(incluye decimales)

(//) solo incluye  
enteros

Exponente → (\*\*)  
doble asterisco

% → solo da 1 entero

ejemplo print (29 % 5)

$$\begin{array}{r} 29 \\ \times 5 \\ \hline 45 \\ 29 \\ \hline 5 \end{array} \rightarrow 5 \text{ con un resto de } 4$$

$$9 + 7 \cdot 8 = 36 / 5$$

$$9 + 7 \cdot 8 = 5,2$$

$$9 + 56 = 5,2$$

$$65 - 5,2$$

$$57,8$$

$$\begin{array}{r} 65,2 \\ - 5,2 \\ \hline 59,8 \end{array}$$

$$36 \underline{) 5}$$

$$10 \underline{) 8,2$$

punto  
no coma

Concatenación

Operador (+) para unir caracteres

texto\_saluda = "Hola!"

texto\_pregunta = "¿cómo estás?"

print ( texto\_saluda + texto\_pregunta )  
→ texto\_completo = texto\_saluda +

para ponerlo en un string

Operador mas-igual (=)

- =

\* =

/ =

suma intrínsecamente

numero\_kilometros\_recorridos = 12

numero\_kilometros\_recorridos + = 2

print ( numero\_kilometros\_recorridos )

# Imprime 14.

Operadores relacionales → Comparan dos elementos y devuelven true o false

( == ) igual que

( > ) Mayor que

( >= ) Mayor o igual que

( != ) diferente de

( < ) Menor que

( <= ) Menor o igual que

## Cálculos

+ suma

- resta

\* multiplicación

/ división

\*\* exponente

// división entera (solo toma la parte entera del resultado)

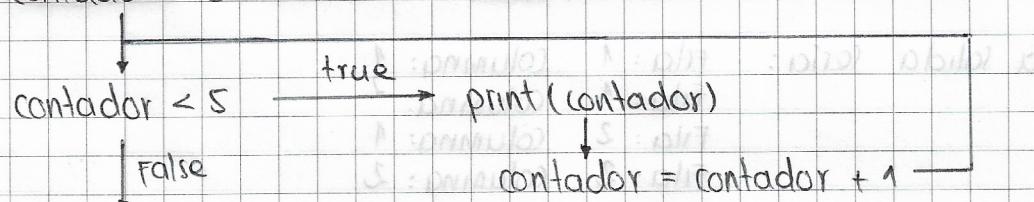
% módulo (toma el residuo de una división)

## Estructuras de control condicional repetitivas (for)

for iterador in range (x):

Es una variable que especifica un valor numérico entero de la lista definida por la función range ()

contador = 0



print ("¡Terminó! ")

For contador in range (5)

    print (contador)

print ("¡Terminó! ")

La función range (n) devuelve una secuencia de números

range (inicio, fin, paso)

Especifica en qué nº empezar

Nº que especifica posición final

Indica si se debe incrementar o decrementar.

**Break** → Con esta instrucción podemos detener el ciclo incluso si la lista de elementos del for no ha terminado de ser iterada

Ej: `for contador in range (5):  
 if (contador == 3):  
 break  
 print (contador)`

**Continue** → Con ella podemos detener (o saltar) la iteración (o iteraciones) que deseamos y continuar con la siguiente

Ej: `for contador in range (5):  
 if (contador == 2 or contador == 3):  
 continue  
 print (contador)`

la salida sería: 0  
1  
4

**Bandera** → Verificadora de eventos

Estructuras de control anidadas

```
for fila in range (1, 3):  
    for columna in range (1, 3):  
        print ("Fila:", fila, "columna:", columna)
```

la salida sería: Fila: 1 columna: 1  
Fila: 1 columna: 2  
Fila: 2 columna: 1  
Fila: 2 columna: 2

Errores y excepciones

Hay 2 tipos de errores distinguibles:

- Errores de sintaxis
- Excepciones

→ parámetro

```
def saludar(nombre):
    print("Hola, " + nombre + " soy un programa")
    Argumento → valor que le doy a la función
    Funciones
```

def nombre\_de\_la\_funcion():
 # las tareas de la función van aquí

Indica el comienzo de una función (también conocida como encabezado de función). Va seguido de un nombre en formato snake-case que describe la tarea que realiza la función

- Los parentesis pueden contener valores de entrada conocidos como **parámetros**.  
Los dos puntos marcan el final del encabezado de la función
- Donde está el comentario tenemos una o más declaraciones válidas de Python que conforman el cuerpo de la función

Ej: def saludo\_de\_bienvenida():
 print("Bienvenido a Bogotá-viajes")
 print("Vamos a llevarte a tu destino")

Esta función se conoce como **función tipo uno** (sin parámetros ni retornos)

**Llamado** (ejecutar una función)

Para llamar a una función en Python, escriba su nombre seguido de paréntesis () .

Ej:

```
def saludo_de_bienvenida():
    print("Bienvenido a Bogotá-viajes")
    print("Vamos a llevarte a tu destino.")
```

def main():

saludo\_de\_bienvenida() — Llamado a la función

main()

## Espacios en blanco y flujo de ejecución

Espacios en blanco → Indica a la computadora que es y que no es parte de una función

flujo de ejecución → El orden en que un programa en Python ejecuta el código

## Parámetros y argumentos

Parámetros → Permiten que nuestra función acepte datos como un valor de entrada

```
def mi_función (parametro_simple):
```

```
    print("Entrada: " + str(parametro_simple))
```

```
    return parametro_simple
```

```
print(mi_función(10))
```

```
Entrada: 10
```

```
10
```

```

def main():
    print("Hola....")
    n = int(input("Dame n: "))
    if (n > 9 or n <= 0):
        print("Error")
    else:
        for f in range(1, n+1):
            for c in range(1, f):
                print(c, end=" ")
            print()

```

Problema 9

↳ Sumatoria de 1 a n  
↳ Sumatoria de las cartas pedidas

la declaración **break** en el **for**

**Break** → nos detiene el ciclo en un punto determinado

Ej.: For contador in range(5): [0, 1, 2, 3, 4]  
 If (contador == 3):  
 break  
 print(contador) → Salida: 0  
 1  
 2

**Continue** → Salta iteraciones

Ej.: for contador in range(5): [0, 1, 2, 3, 4]  
 If contador == 2 or contador == 3:  
 continue  
 print(contador) → Salida: 0  
 1  
 4

Concepto de bandera

- Es una variable cuyo contenido permite saber si algo sucedió en el algoritmo
- Una variable que cambia de estado en un proceso cíclico y luego la uso para tomar una decisión

Estructuras de control anidadas

Un ciclo anidado son ciclos dentro de ciclos

Ej.: for fila in range(1, 3): [1, 2]  
 for columna in range(1, 3): [1, 2]  
 print("Fila:", fila, "Columna:", columna)

```

" " "
" " "
def main():
    print ("Hola... ")
    n = int (input ("Dame n: "))
    if (n <= 0):
        print ("Error")
    else:
        cont = 1
        acum = 1
        print (acum, end = " ")
        while (cont <= n):
            if (cont % 3 == 0):
                acum = acum + 3
            else:
                acum = acum * 2
            print (acum, end = " ")
            cont = cont + 1
main()

```

Variable	Tipo	Vibr	Pantalla
n	int	2 6	Holaod ob ofgma)
acum	int	2 12 4 7 128 1 2 1 7 14 28	(contando ob ofgma) -

Corrección prueba corta

" " "

" " "

def patron (n):

For f in range (n, 0, -1):

For c in range (n, f-1, -1):

print (c, end = " ")

print ()

para que tenga espacio

→ Para que pase a otro renglón

4

4 3

4 3 2

4 3 2 1

def patron (n):

If (n % 2 == 0): # Par  
par = 2

For f in range (n, 0, -1):

for c in range (par):

print (f, end = " ")

par = par + 2 # par += 2

print ()

else:

impar = 1

for f in range (n, 0, -1):

for c in range (impar):

print (f, end = " ")

impar += 2

print ()

4 4

3 3 3 3

2 2 2 2 2 2

1 1 1 1 1 1 1 1

Calcular sumatoria de  $(k! + \sum_{v=valin}^{valsu} (v-2)^p)$

def factorial (k):

fact = 1

for f in range (1, k+1):

fact \*= f

return fact

def sumatoria (valin, valsu):

acum = 0

for v in range (valin, valsu + 1):

acum += (v-2)

return acum

def sum\_completa (k, valin, valsu, p):

s = (factorial (k) + sumatoria (valin, valsu)) \*\* p

return s

- def factorial (n):
 

```
Fact = 1
FOR F in range (1, n+1):
    Fact *= F
return Fact
```
- def sumatoria (n-1, n-2)
 

```
acum = 0
FOR v in range (n-1, n-2+1):
    acum += (valor que aparece en sumatoria)
return acum
```
- For contador in range (5):
 

```
if (contador == 3):
    break
print (contador)
```

0  
1  
2
- For contador in range (5):
 

```
if (contador == 2 or contador == 3):
    continue
print (contador)
```

0  
1  
4
- def num\_entero (mensaje):
 

```
while (True):
    try:
        n = int (input (mensaje))
    break
    except (ValueError):
        print ("Error")
return n
```
- def num\_real (mensaje):
 

```
while (True):
    try:
        n = float (input (mensaje))
    break
    except (ValueError):
        print ("Error")
return n
```
- def piramide (n):
 

```
cont = 1
FOR F in range (1, n+1):
    FOR C in range (cont):
        print (" ", end="")
    cont += 1
    print ("#")
print () # opcional
```

4	-	1
33	-	12
222	-	123
1111	-	1234

## Arrays

1D

3	5	7	9	4	3	1
0	1	2	3	4	5	6

elementos

indice

2D

	0	1	2
0	00	01	02
1	10	11	12
2	20	21	22

columnas

Indices

↓  
Filas

## listas

alturas = [ 171, 170, 179, 184 ]

Se usan corchetes para definirlo []

(Cada elemento se separa por una coma ,)

Lista vacía → no tiene elementos pero se pueden dar con un append

ej Alturas = [] → lista vacía

• append () → lo usa para agregar 1 elemento a una lista

Ej: jardín = []

jardín.append ("Tomates")

print (jardín)

Alumno: Maria Alejandra Vivas Duarte

Nota:

17

(5.0 puntos) Demostrando sólidos conocimientos en la solución de problemas mediante la implementación de algoritmos y del lenguaje de programación Python, solucione el (los) siguiente (s) problema (s) como el profesor le indique (papel y lápiz, computador o los dos):

0.7 ✓ Problema 1 (3.0 puntos)

Calcula el valor aproximado de la función  $\cos(x)$ ,  $x \in \mathbb{R}$ , por medio de la serie de Taylor. El usuario indicará también con cuantos términos  $n$  quiere que se haga el cálculo con  $n > 0$ .

$$\cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

Caso

$$x = \frac{\pi}{6} \approx 0.52$$

Cantidad de términos: 3

Resultado

$\cos(0.52) \approx 0.8678465066666666$ , para 3 términos de la serie de Taylor.

Explicación

$$\cos(0.52) \approx 1 - \frac{0.52^2}{2!} + \frac{0.52^4}{4!} = 0.8678465066666666$$

18 → 100  
V → -

1.1 ✓ Problema 2 (2.0 puntos)

Pedir un texto al usuario (carácter por carácter) y decir a qué porcentaje de la longitud del texto equivale la cantidad de vocales que éste tiene. Escriba también la cantidad de vocales encontradas y la longitud del texto, recuerde que los espacios en blanco también son caracteres. Cada carácter se ingresará en minúsculas, recuerde que puede haber acentos en las vocales. El fin del texto es identificado mediante el carácter # (numeral).

<u>Caso</u>	<u>Caso</u>
Texto ingresado: <i>la casa es bonita.#</i>	Texto ingresado: <i>jholail esta es otra oración.#</i>
<u>Resultado</u>	<u>Resultado</u>
La cantidad de vocales (7) equivale al 38.8888% de la longitud del texto de (18) caracteres.	La cantidad de vocales (11) equivale al 37.9310% de la longitud del texto de (29) caracteres.

**NOTA IMPORTANTE:** Valide todo lo que considere necesario para que los algoritmos funcionen acorde a los enunciados, recuerden que pueden existir otros casos de prueba y los algoritmos deben ser consistentes y evitar fallos tanto de ejecución como de lógica. Haga uso de todos los conceptos vistos en el aula de clase debidamente practicados durante las semanas previas a este parcial, solo hasta las estructuras de control repetitivas (ciclo while). Si hace uso de estructuras, librerías, módulos matemáticos o estrategias de programación no vistas aún en el aula de clase, el punto no será tenido en cuenta. No se exponga.

Maria Alejandra Vivas Duarte

Nombre : Maria Alejandra Vivas

Fecha: 21-02-2023

Descripción: Parcial del primer tercio (Problema 1)

def Main () :

print ("Hola, te ayudaré a calcular el valor aproximado de  
la función cos(x)")

x = int(input ("Dame un valor para x que sea real: " ))

n = ? ?

1.5

Nombre: María Alejandra Vivas

Fecha: 21-02-2023

Descripción: Problema 2

def main():

print ("Hola, te diré a que porcentaje de longitud de tu texto equivalen las vocales que tenga.")

n = 1

l = input ("Dame 1 carácter en minúscula: ")

v = 0

while (l != "#")

" eran or

~~if ( l == "a" and l == "e" and l == "i" and l == "o" and l == "u" and l == "á" and l == "é" and l == "í" and l == "ó" and l == "ú" )~~

v = v + 1

else: } No habrá resultado.

v = v

n = n + 1

→ muestra error por el mensaje del input.

l = input ("Dame ", n, " caracteres en minúscula: ")

p = ((v \* 100) / n)

Porque solo la parte entera ??  
 print ("La cantidad de vocales (", v, ") equivale al ", p, "% de la longitud del texto de (", n, ")")  
 (caracteres)")

main()