

A continuación, detallo el parcial según instrucciones.

Sección I:

4)

```
print('(Se pueden ingresar en decimales.))')
ladoMenor = float(input('Ingrese el lado menor del
triángulo:\n '))
ladoMayor = float(input('Ingrese el lado mayor del
triángulo:\n '))
```

```
#~~~~~#
```

```
superficie = ladoMayor * ladoMenor
perimetro = ladoMayor * 2 + ladoMenor * 2
```

```
#~~~~~#
```

```
superficie = round(superficie, 2)
perimetro = round(perimetro, 2)
```

```
#~~~~~#
```

```
print('La superficie del triángulo rectángulo, es',
superficie, 'y el perímetro es', perimetro, end='.')
```

6)

```
import math
```

```
#~~~~~#
```

```
print('(Se puede ingresar en decimales.))')
radio = float(input('Ingrese el radio de la esfera:\n '))
```

```
#~~~~~#
```

```
area = 4 * math.pi * radio ** 2
volumen = (4 * math.pi * radio ** 3) / 3
```

```
#~~~~~#
```

```
area = round(area, 2)
```

```
volumen = round(volumen, 2)
```

```
#~~~~~#
```

```
print('El área de la esfera es', área, ', y su volumen',  
volumen, end='.')
```

Sección II:

5)

```
#~~~~~//definiciones\\~~~~~#
```

```
fallo = 'Ingresó mal alguno de los datos. Vuelva a  
ingresar.'
```

```
meses = ('Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo',  
'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre',  
'Noviembre', 'Diciembre')
```

```
#~~~~~#
```

```
diasMes = {1 : 31, 2 : 28, 3 : 31, 4 : 30, 5 : 31,  
           6 : 30, 7 : 31, 8 : 31, 9 : 30,  
           10 : 31, 11 : 30, 12 : 31}
```

```
bisiesto = {2 : 29}
```

```
#=====//  
=====
```

```
#~~~~~//funciones\\~~~~~#
```

```
def anioBisiesto(anio):  
    if anio % 4 == 0 and anio % 100 != 0:  
        diasMes.update(bisiesto)
```

```
#=====//  
=====
```

```
try:  
    mes = int(input('Ingrese el mes en número:\n '))  
    anio = int(input('Ingrese el año:\n '))  
    anioBisiesto(anio)
```



```

#~~~~~//funcion de repeticion\\
~~~~~#

def aster(x,y):
    if x>((y-1)//2):
        return ' '
    else:
        return '*'

#=====//Ingreso\\~~~~~#

print('Desde aqui, ingrese tres números...')
for i in range(0, 3):
    numero = int(input('>'))
    numeros.append(numero)

#~~~~~//Ciclador de asteriscos\\
~~~~~#

for i in numeros:
    if i > maximo:
        maximo = i
print(numeros[0], ' ', numeros[1], ' ', numeros[2])

for j in range(maximo):
    p1=aster(j,numeros[0])
    p2=aster(j,numeros[1])
    p3=aster(j,numeros[2])

    print(p1, ' ', p2, ' ', p3)

```

Sección IV:

11)

```

#~~~~~//Definiciones\\~~~~~#

pasar = False
abecedario = ' abcdefghijklmnñopqrstuvwxyz'
ABECEDARIO = abecedario.upper()

```

```
        #~~~~~//Estructuras del programa y
control\\~~~~~#
```

```
def main(frase, abecedario):
    validado = control(frase, abecedario)
    caracter = cesarCaracteres(abecedario,
desplazamiento, validado)
```

```
def fallo():
    pasar = False
    print('Ingresó mal alguno de los datos. Intentelo de
nuevo.')
```

```
def validador(direccion, desplazamiento):
    if direccion == 'D' or direccion == 'I':
        if desplazamiento < 27:
            pasar = True
        else:
            print('Tiene que ser un número dentro del
abecedario, el máximo es 27')
    else:
        fallo()
    return pasar
```

```
def control(frase, abecedario):
    caracter = False
    for f in range(0, len(frase)):
        for a in abecedario:
            if frase[f] == a:
                caracter = True
    return caracter
```

```
def negativo(direccion, desplazamiento):
    if direccion == 'I':
        desplazamiento = 0 - desplazamiento
    return desplazamiento
```

```
        #~~~~~//Transformación a Cesar\\
~~~~~#
```

```
def ciclos(valor):
    ciclos = []

    for i in range(0, 3):
```



```

        for j in range(0, len(valor)):
            ciclos.append(valor[j])
    return ciclos

def formato(abecedario, frase):
    codABC = []
    rastreo = []
    espaciado = []
    for f in range(0, len(frase)):
        espacio = False
        if frase[f] == abecedario[0]:
            espacio = True
        espaciado.append(espacio)

        minuscula = False
        if frase[f].islower():
            minuscula = True
        rastreo.append(minuscula)
        for a in range(0, len(abecedario)):
            if frase[f].lower() == abecedario[a]:
                codABC.append(a)
    return codABC, rastreo, espaciado

def cesarCaracteres(abecedario, desplazamiento,
validado):
    if validado:
        ciclosABC = ciclos(abecedario)

        indices = formato(abecedario, frase)
        caracter = indices[0]
        minuscula = indices[1]
        espacios = indices[2]

        contador = 0
        cesar = []
        for c in caracter:
            if espacios[contador]:
                print(' ',end='')
            elif minuscula[contador]:
                print(ciclosABC[c + 27 +
(desplazamiento)], end='')
            else:
                print(ciclosABC[c + 27 +
(desplazamiento)].upper(), end='')

```

contador += 1

#=====

while not pasar:

try:

 direccion = input('¿Dirección del codificado?
("D" para la derecha, "I" para la izquierda).\n

 desplazamiento = int(input('Ingrese número de
letras a desplazar:\n '))

 frase = input('Ingrese una frase:\n ')

 pasar = validador(direccion, desplazamiento)

 desplazamiento = negativo(direccion,
desplazamiento)

 main(frase, abecedario)

except:

 fallo()

10)

entrada='a'

frases=[]

while entrada!='':

 entrada=input('Ingrese su frase: ')

 frases.append(entrada)

max=0

larga=''

for i in frases:

 if len(i)>max:

 larga=i

 max=len(i)

print (larga)

print(max)

Sección V:

2.

cadena = []

numero = 5

```
agregando = True
```

```
while agregando:  
    numero = int(input('Ingrese un numero:\n '))  
    if numero < 0:  
        agregando = False  
    else:  
        cadena.append(numero)  
print(sorted(cadena))
```

5.

```
alfabeto = []
```

```
frase = input('Ingrese una frase:\n ').lower()
```

```
for x in frase:  
    if x != ' ' and x not in alfabeto:  
        alfabeto.append(x)  
print(sorted(alfabeto))
```