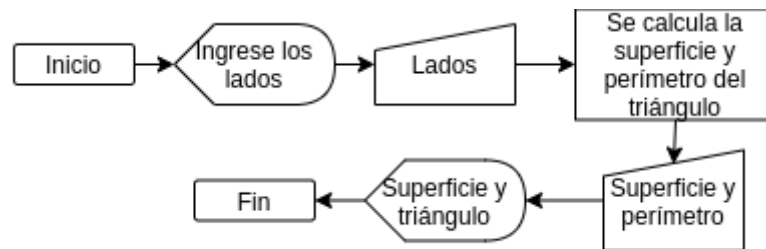


A continuación, detallo el parcial según instrucciones. En primer lugar, doy una breve orientación de como esta planteado el programa, seguido por un diagrama de flujo en los dos primeros, el código, y su ejecución.

Sección I:

4)En este programa: Ingreso los lados del triángulo, y lo calcúlo como se haría matematicamente.



```

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='.')
  
```

Ejecuciones:

Python 3.7.9 (bundled)

```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

(Se pueden ingresar en decimales.)

Ingrese el lado menor del triángulo:

34

Ingrese el lado mayor del triángulo:

56

La superficie del triángulo rectángulo, es 1904.0 y el perímetro es 180.0.

```
>>> |
```

```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

(Se pueden ingresar en decimales.)

Ingrese el lado menor del triángulo:

4

Ingrese el lado mayor del triángulo:

8

La superficie del triángulo rectángulo, es 32.0 y el perímetro es 24.0.

```
>>> |
```

```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

(Se pueden ingresar en decimales.)

Ingrese el lado menor del triángulo:

78

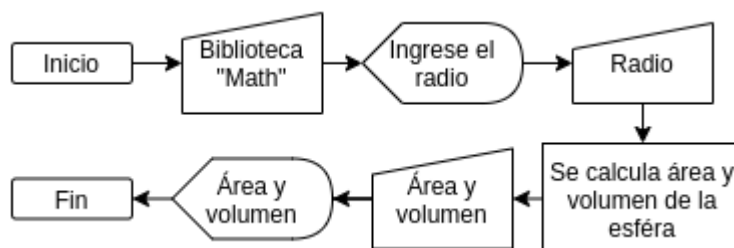
Ingrese el lado mayor del triángulo:

159

La superficie del triángulo rectángulo, es 12402.0 y el perímetro es 474.0.

```
>>> |
```

6)En este programa: Tomo por fuera la biblioteca de Matemática de Python, para hacer el cálculo del área y el volumen, con el radio ingresado por el usuario.



```
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', area, ', y su volumen',
volumen, end='.')
```

Ejecuciones:

```
>>> %Run '#MESA_DE_TRABAJO#.py'
(Se puede ingresar en decimales.)
Ingrese el radio de la esfera:
15
El área de la esfera es 2827.43 , y su volumen 14137.17.
>>> |
```

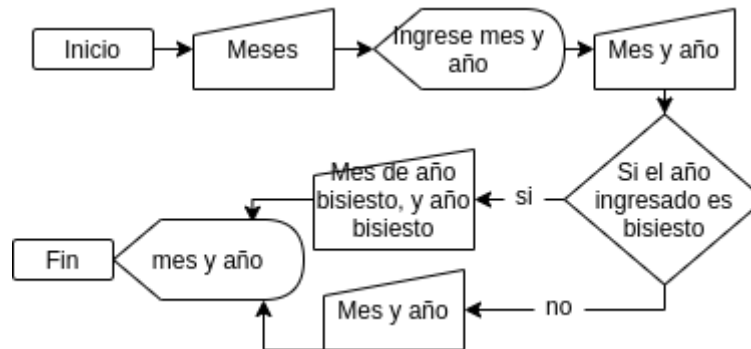
```
>>> %Run '#MESA_DE_TRABAJO#.py'
(Se puede ingresar en decimales.)
Ingrese el radio de la esfera:
67.48
El área de la esfera es 57221.6 , y su volumen 1287104.57.
>>> |
```

```
>>> %Run '#MESA_DE_TRABAJO#.py'
(Se puede ingresar en decimales.)
Ingrese el radio de la esfera:
679.148
El área de la esfera es 5796137.99 , y su volumen 1312145174.34.
>>> |
```

Sección II:

5)En este programa: Comparo los datos ingresados por usuario, con la función “anioBisiesto(anio)” para saber

si es bisiestro, y a esa devolución la comparo con los datos de los meses.



#~~~~~//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)
```

```
    print('El mes de', meses[mes - 1], 'del año', anio,
', tiene en total', diasMes[mes], 'días.')
```

```
except:
    print(fallo)
```

Ejecuciones:

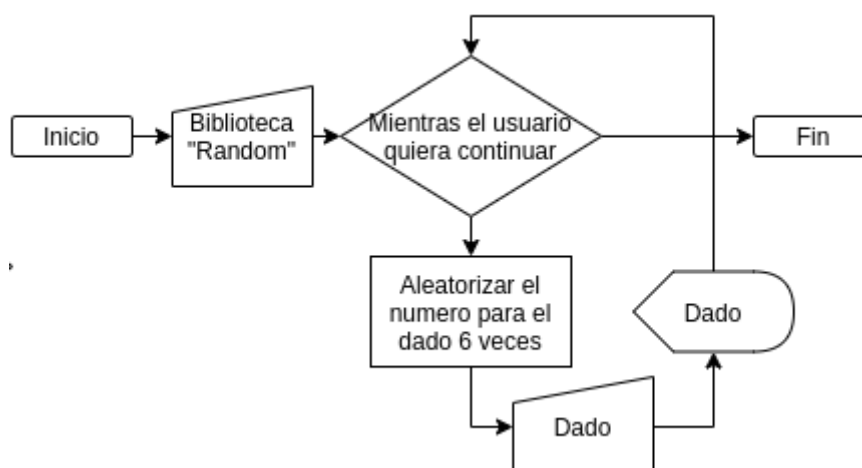
```
>>> %Run '#MESA_DE_TRABAJO#.py'
Ingrese el mes en número:
7
Ingrese el año:
2021
El mes de Julio del año 2021 , tiene en total 31 días.

>>> |
>>> %Run '#MESA_DE_TRABAJO#.py'
Ingrese el mes en número:
2
Ingrese el año:
2024
El mes de Febrero del año 2024 , tiene en total 29 días.

>>> |
>>> %Run '#MESA_DE_TRABAJO#.py'
Ingrese el mes en número:
9
Ingrese el año:
1998
El mes de Septiembre del año 1998 , tiene en total 30 días.

>>> |
```

6)En este programa: Llamo a la función random para aleatorizar los números, los reparto entre 6, y los imprimo.



```

import random

continuidad = ''

while continuidad != '*':
    dado = []
    for i in range(0, 6):
        dado.append(random.randint(1, 6))

    print(' ----- \n|', dado[0], dado[1], '| \n|',
          dado[2], dado[3],
          '| \n|', dado[4], dado[5], '| \n ----- ')

    print('\n\n Siga presionando Enter para seguir dando
tiradas.',
          '\nPresione "*" antes de Enter para salir.')
    continuidad = input(' ')

if continuidad == '*':
    print('\nQue tenga un buen día...')

```

Ejecuciones:

```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

```
-----  
| 1 5 |  
| 3 3 |  
| 3 5 |  
-----
```

Siga presionando Enter para seguir dando tiradas.
Presione "*" antes de Enter para salir.

```
-----  
| 6 5 |  
| 4 4 |  
| 1 6 |  
-----
```

Siga presionando Enter para seguir dando tiradas.
Presione "*" antes de Enter para salir.

```
-----  
| 6 5 |  
| 1 3 |  
| 1 6 |  
-----
```

Siga presionando Enter para seguir dando tiradas.
Presione "*" antes de Enter para salir.

*

Que tenga un buen día...

```
>>> |
```

```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

```
-----  
| 6 4 |  
| 3 4 |  
| 2 1 |  
-----
```

Siga presionando Enter para seguir dando tiradas.
Presione "*" antes de Enter para salir.

```
-----  
| 6 3 |  
| 6 2 |  
| 4 6 |  
-----
```

Siga presionando Enter para seguir dando tiradas.
Presione "*" antes de Enter para salir.

```
-----  
| 4 2 |  
| 5 6 |  
| 6 2 |  
-----
```

Siga presionando Enter para seguir dando tiradas.
Presione "*" antes de Enter para salir.

*

Que tenga un buen día...

```
>>> |
```



```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

```
-----  
| 1 6 |  
| 2 6 |  
| 5 5 |  
-----
```

Siga presionando Enter para seguir dando tiradas.
Presione "*" antes de Enter para salir.

```
-----  
| 2 3 |  
| 2 4 |  
| 5 4 |  
-----
```

Siga presionando Enter para seguir dando tiradas.
Presione "*" antes de Enter para salir.

```
-----  
| 4 5 |  
| 6 2 |  
| 4 4 |  
-----
```

Siga presionando Enter para seguir dando tiradas.
Presione "*" antes de Enter para salir.

*

Que tenga un buen día...

```
>>> |
```

Sección III:

5)En este programa: Hago funciones de impresión de emojis, y hago una recorrida que pregunta si el usuario ingresó bien los números, y le muestra emojis hasta que se canse.

```
def feliz():  
    if orden == 1:  
        print('_____)  
        print(' /      \ ')  
        print(' /      \ ')  
        print(' /      \ ')
```



```

        print('|'
        print('|
|')
        print('|
|')
print('\
        \
        /')
    print(' \
    print(' \
    print(' \
    print(' \
    print(' \
#~~~~~#

def fallo():
    print('Ingresó mal sus sensaciones. Intentelo de
nuevo.')
```

```

#=====
=====#
```

```

funcionando = True
```

```

#~~~~~#
```

```

while funcionando:
    try:
        orden = input('\n\n¿Como te sientes?\n1- Feliz\
n2- Triste\n3- Canchero\n4- Esceptico\n\n(Ingrese "*"
para salir).\n\n')
        if orden == '*':
            funcionando = False
            print('Mantenga esa alegría, que no existen
motivos de tristeza. Nunca...')
        elif len(orden) == 1:
            orden = int(orden)
            emojis = (feliz(), triste(), guinio(),
esceptico())
        else:
            fallo()

    except:
        fallo()
```

Ejecuciones:

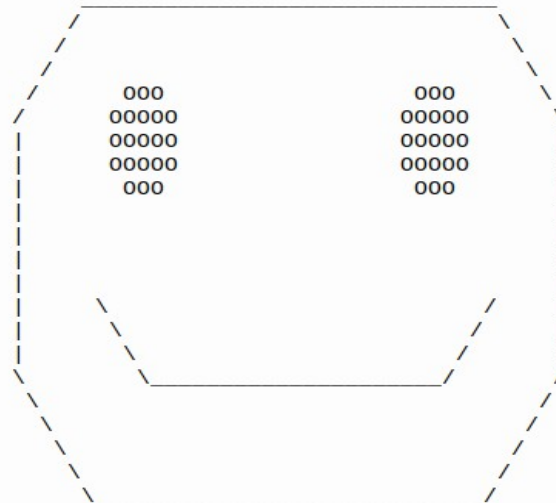
```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

```
¿Como te sientes?
```

- 1- Feliz
- 2- Triste
- 3- Canchero
- 4- Esceptico

```
(Ingrese "*" para salir).
```

```
1
```

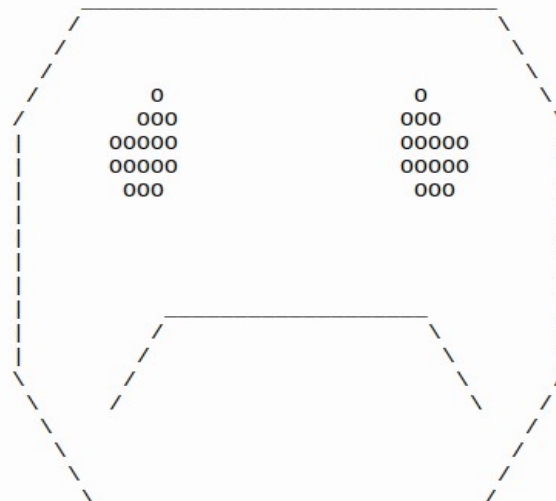


```
¿Como te sientes?
```

- 1- Feliz
- 2- Triste
- 3- Canchero
- 4- Esceptico

```
(Ingrese "*" para salir).
```

```
2
```



```
¿Como te sientes?
```

- 1- Feliz
- 2- Triste
- 3- Canchero
- 4- Esceptico

```
(Ingrese "*" para salir).
```

```
*
```

```
Mantenga esa alegría, que no existen motivos de tristeza. Nunca...
```

```
>>> |
```

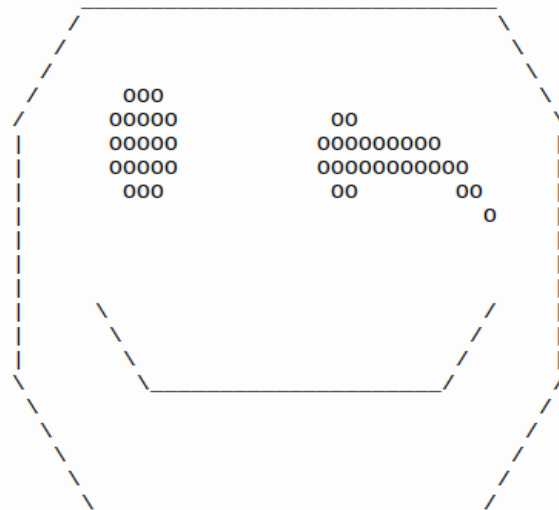
```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

```
¿Como te sientes?
```

- 1- Feliz
- 2- Triste
- 3- Canchero
- 4- Esceptico

```
(Ingrese "*" para salir).
```

```
3
```

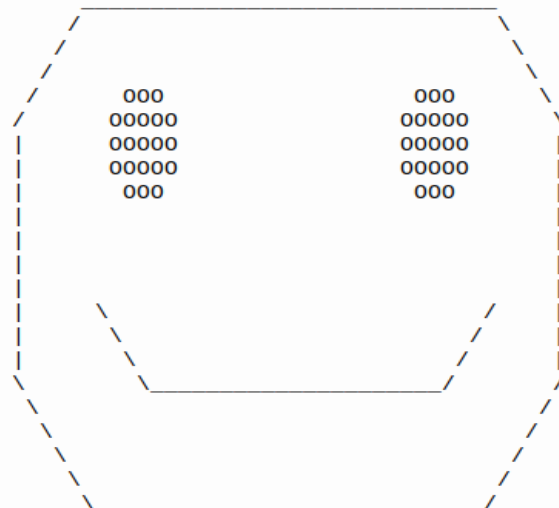


```
¿Como te sientes?
```

- 1- Feliz
- 2- Triste
- 3- Canchero
- 4- Esceptico

```
(Ingrese "*" para salir).
```

```
1
```



```
¿Como te sientes?
```

- 1- Feliz
- 2- Triste
- 3- Canchero
- 4- Esceptico

```
(Ingrese "*" para salir).
```

```
*
```

```
Mantenga esa alegría, que no existen motivos de tristeza. Nunca...
```

```
>>> |
```

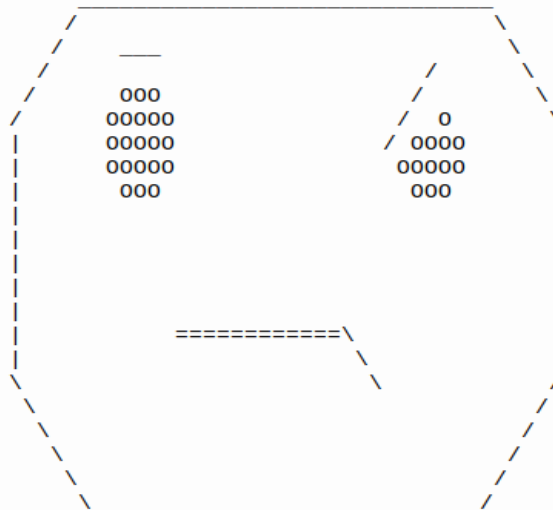
```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

¿Como te sientes?

- 1- Feliz
- 2- Triste
- 3- Canchero
- 4- Esceptico

(Ingrese "*" para salir).

4

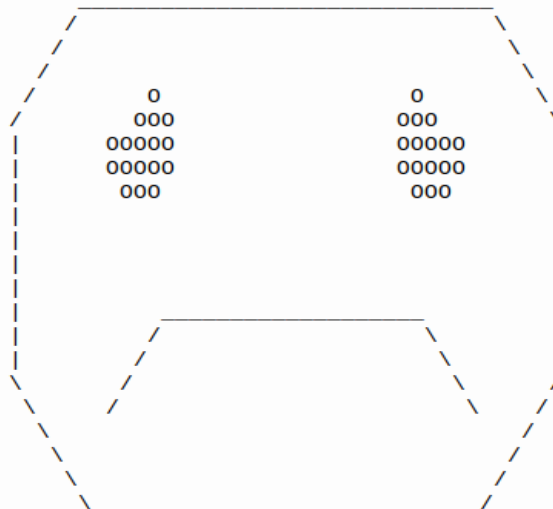


¿Como te sientes?

- 1- Feliz
- 2- Triste
- 3- Canchero
- 4- Esceptico

(Ingrese "*" para salir).

2



¿Como te sientes?

- 1- Feliz
- 2- Triste
- 3- Canchero
- 4- Esceptico

(Ingrese "*" para salir).

*

Mantenga esa alegría, que no existen motivos de tristeza. Nunca...

```
>>> |
```

11)En este programa: Se ingresan tres números, y el programa compara cual de ellos va siendo mas grande, y de esta forma va imprimiendo asteriscos, hasta que llegue al número límite.

```
#~~~~~//definiciones\\~~~~~#
maximo = 0
numeros = []

#~~~~~//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)
```

Ejecuciones:


```
%Run '#MESA_DE_TRABAJO#.py'
```

```
Desde aqui, ingrese tres números...
```

```
>3
```

```
>7
```

```
>4
```

```
3      7      4
*      *      *
*      *      *
      *
      *
```

```
>>> |
```

```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

```
Desde aqui, ingrese tres números...
```

```
>10
```

```
>15
```

```
>20
```

```
10      15      20
*      *      *
*      *      *
*      *      *
*      *      *
*      *      *
      *      *
      *      *
      *      *
```

```
>>> %Run '#MESA_DE_TRABAJO#.py'
```

```
Desde aqui, ingrese tres números...
```

```
>20
```

```
>13
```

```
>12
```

```
20      13      12
*      *      *
*      *      *
*      *      *
*      *      *
*      *      *
*      *      *
*      *
*
*
*
```

```
>>> |
```

Sección IV:

```
>>> |
```

11)En este programa: El programa del cifrado cesar lo hice bastante largo, puede procesar las frases de izquierda a derecha, y comprobar caracteres mal ingresados.

Después del ingreso, se toma validez de los datos, y pasa a una función central, que dicha función central, hace también una evaluación técnica, básicamente, de que el número no exceda el intervalo del abecedario, tanto para la izquierda como para la derecha.

Por último, y no menos importante, se hace el desplazamiento, que es tan simple como sumar un número, ya sea este negativo, o positivo.

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

```
pasar = False
abecedario = ' abcdefghijklmnñopqrstuvwxyz'
```

```
#~~~~~//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]
minusculta = indices[1]
espacios = indices[2]

contador = 0
cesar = []
for c in caracter:
    if espacios[contador]:
        print(' ',end='')
    elif minusculta[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
').upper()
        desplazamiento = int(input('Ingresa número de
letras a desplazar:\n '))
        frase = input('Ingresa una frase:\n ')
        pasar = validador(direccion, desplazamiento)
        desplazamiento = negativo(direccion,
desplazamiento)
        main(frase, abecedario)

    except:
        fallo()

```

Ejecuciones:

```
>>> %Run '#MESA_DE_TRABAJO#.py'
¿Dirección del codificado? ("D" para la derecha, "I" para la izquierda).
d
Ingrese número de letras a desplazar:
7
Ingrese una frase:
Wikipedia La Enciclopedia Libre
Bñpñvkjñg Qg Ksiñiquvkjñg Qñhxxk
>>> |
```

```
>>> %Run '#MESA_DE_TRABAJO#.py'
¿Dirección del codificado? ("D" para la derecha, "I" para la izquierda).
i
Ingrese número de letras a desplazar:
20
Ingrese una frase:
Agradezco todos los dias a la obra Don Bosco por enseñarme lo que tanto me gusta
Hnyhklfjv vkvz rvz kohz h rh viyh Kvt Ivzjv wvy ltzluhysl rv xal ht v sl naz h
>>> |
```

```
>>> %Run '#MESA_DE_TRABAJO#.py'
¿Dirección del codificado? ("D" para la derecha, "I" para la izquierda).
dshnsfhdfjndf
Ingrese número de letras a desplazar:
Cuarenta y cinco
Ingresó mal alguno de los datos. Intentelo de nuevo.
¿Dirección del codificado? ("D" para la derecha, "I" para la izquierda).
d
Ingrese número de letras a desplazar:
115
Ingrese una frase:
Desde el CFP hasta el infinito y mas alla
Tiene que ser un número dentro del abecedario, el máximo es 27
Ingresó mal alguno de los datos. Intentelo de nuevo.
¿Dirección del codificado? ("D" para la derecha, "I" para la izquierda).
d
Ingrese número de letras a desplazar:
15
Ingrese una frase:
Desde el CFP hasta el infinito y mas alla haciendo lo que mas me gusta
Qrfqr ry PSC uñfgñ ry v sv vgb l zñf ñyyñ uñpvr qb yb dhr zñf zr thfgñ
>>> |
```

10)En este programa: El usuario ingresa frases, y si la frase siguiente, es mas larga que la anterior, incrementa un valor máximo, comparando así al final el largo de la frase con el máximo.

```
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)

```

Ejecuciones:

```

>>> %Run '#MESA_DE_TRABAJO#.py'

Ingrese su frase: La casa esta pintada de rosa
Ingrese su frase: El perro es muy lindo
Ingrese su frase: El colectivo esta en malas condiciones
Ingrese su frase:
_ _ _ _ _
>>> %Run '#MESA_DE_TRABAJO#.py'

Ingrese su frase: Hay que ser muy perceptivo para tomar liderazgo
Ingrese su frase: Nunca haber aprendido tanto, como con Coco, Diego, y Esteban
Ingrese su frase:
Nunca haber aprendido tanto, como con Coco, Diego, y Esteban
60
>>> |

```

```

>>> %Run '#MESA_DE_TRABAJO#.py'

Ingrese su frase: Mi hermana tiene el mismo primer y segundo nombre que mi cuñada, del lado de la pareja de ella, tremendo.
Ingrese su frase: Un auto es viejo y se deteriora si no se mantiene, pero si si se le hace, puede llegar a ser mas atractivo, y hasta mas eficiente en algunos aspectos que los coches de ultima gama
Ingrese su frase:
Un auto es viejo y se deteriora si no se mantiene, pero si si se le hace, puede llegar a ser mas atractivo, y hasta mas eficiente en algunos aspectos que los coches de ultima gama
179
>>> |

```

Sección V:

2.En este programa: el usuario ingresa una cadena de números, y se ordenan con “sorted()”, terminando esta lista, si cuando un número es menor que 0.

```

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))

```

Ejecuciones:

```

>>> %Run '#MESA_DE_TRABAJO#.py'
Ingrese un numero:
5
Ingrese un numero:
3
Ingrese un numero:
7
Ingrese un numero:
56
Ingrese un numero:
2
Ingrese un numero:
123
Ingrese un numero:
-456
[2, 3, 5, 7, 56, 123]
>>> |

>>> %Run '#MESA_DE_TRABAJO#.py'
Ingrese un numero:
2
Ingrese un numero:
15
Ingrese un numero:
34
Ingrese un numero:
-5
[2, 15, 34]
>>> |

>>> %Run '#MESA_DE_TRABAJO#.py'
Ingrese un numero:
56
Ingrese un numero:
3456789
Ingrese un numero:
1
Ingrese un numero:
-10
[1, 56, 3456789]
>>> |

```

5.En este programa: El usuario ingresa una frase, y esta es medida en no tener espacios ni letras repetidas, y se muestra las letras que usa en una lista ordenada.

```
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))

```

Ejecuciones:

```

>>> %Run '#MESA_DE_TRABAJO#.py'

Ingrese una frase:
La casa en el arbol
['a', 'b', 'c', 'e', 'l', 'n', 'o', 'r', 's']

>>> |

```

```

>>> %Run '#MESA_DE_TRABAJO#.py'

Ingrese una frase:
Además de ganar plata, la vida en el desarrollo tiene otras facetas mucho mas importantes, uno se mantiene lógico, en constante dinamismo, y dispuesto a aprender más.
['.', 'a', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'y', 'á', 'ó']

>>> |

>>> %Run '#MESA_DE_TRABAJO#.py'

Ingrese una frase:
Además de mantenerte lógico, uno aprende el verdadero valor de una vida mas "Modularizada".
['.', 'a', 'c', 'd', 'e', 'g', 'i', 'l', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'z', 'á', 'ó']

>>> |

```