

EJERCICIOS PYTHON

1.- Programa de astrología:

El usuario debe ingresar el día y mes de su cumpleaños y el programa le debe decir a que signo corresponde.

Aries: 21 de marzo al 20 de abril.

Tauro: 21 de abril al 20 de mayo.

Geminis: 21 de mayo al 21 de junio.

Cancer: 22 de junio al 23 de julio.

Leo: 24 de julio al 23 de agosto.

Virgo: 24 de agosto al 23 de septiembre.

Libra: 24 de septiembre al 22 de octubre.

Escorpio: 23 de octubre al 22 de noviembre.

Sagitario: 23 de noviembre al 21 de diciembre.

Capricornio: 22 de diciembre al 20 de enero.

Acuario: 21 de enero al 19 de febrero.

Piscis: 20 de febrero al 20 de marzo.

2.- Algoritmo de Euclides

a) Escribir el algoritmo de Euclides para calcular el máximo común divisor de dos números n y m , dado por los siguientes pasos.

1. Teniendo n y m , se obtiene r , el resto de la división entera de $m=n$.

2. Si r es cero, n es el mcd de los valores iniciales.

3. Se reemplaza $m \leftarrow n$, $n \leftarrow r$, y se vuelve al primer paso.

b) Hacer la prueba de escritorio del algoritmo implementado para los siguientes pares de números: (15,9); (9,15); (10,8); (12,6).

3.- Escribir en código python una función que reciba dos números como parámetros, y devuelva cuántos múltiplos del primero hay, que sean menores que el segundo.

a) Implementarla utilizando un ciclo for, desde el primer número hasta el segundo.

b) Implementarla utilizando un ciclo while, que multiplique el primer número hasta que sea mayor que el segundo.

c) Comparar ambas implementaciones: ¿Cuál es más clara? ¿Cuál realiza menos operaciones?

4.- El usuario debe ingresar una fecha (día, mes, año) y el programa debe indicar si es válida o no. Considerar los años bisiestos.

5.- El usuario debe ingresar el valor total de una compra, calcular el IVA e indicar al usuario el total que debe cancelar. Después solicitar la forma de pago (Efectivo, TC). Si el pago es en efectivo tiene un 5% de descuento antes del cálculo del IVA (se debe calcular nuevamente el total que debe cancelar), solicitar el valor a cancelar y calcular el vuelto, en el cuál se debe indicar cuántos billetes se debe entregar de \$50, \$20, \$10, \$5, \$1, o monedas de \$0.50, \$0.25, \$0.10, \$0.05 o \$0.01.

6.- Escribir un programa en python que le pida al usuario que ingrese una sucesión de números naturales (primero uno, luego otro, y así hasta que el usuario ingrese '-1' como condición de salida). Al final, el programa debe imprimir cuántos números fueron ingresados, la suma total de los valores y el promedio.

7.- Manejo de contraseñas

Escribir un programa que contenga una contraseña inventada, que le pregunte al

usuario la contraseña, y no le permita continuar hasta que la haya ingresado correctamente. Se debe considerar:

- a) Solo tiene 5 intentos para ingresar la contraseña.
- b) Después de cada intento agregue una pausa cada vez mayor, utilizando la función sleep del módulo time.
- d) Debe ser una función la que devuelva si el usuario ingresó o no la contraseña correctamente, mediante un valor booleano (True o False).

8.- Utilizando la función randrange del módulo random, escribir un programa que obtenga un número aleatorio secreto, y luego permita al usuario ingresar números y le indique si son menores o mayores que el número a adivinar, hasta que el usuario ingrese el número correcto.

9.- Escribir una función que reciba una cadena que contiene un largo número entero y devuelva una cadena con el número y las separaciones de miles. Por ejemplo, si recibe '1234567890', debe devolver '1.234.567.890'.

10.- Escribir funciones que dadas dos cadenas de caracteres:

- a) Indique si la segunda cadena es una subcadena de la primera. Por ejemplo, 'cadena' es una subcadena de 'subcadena'.
- b) Devuelva la que sea anterior en orden alfabético. Por ejemplo, si recibe 'kde' y 'gnome' debe devolver 'gnome'.

11.- Escribir una función que reciba una cadena de unos y ceros (es decir, un número en representación binaria) y devuelva el valor decimal correspondiente.

12.- Escribir funciones que dada una cadena de caracteres:

- a) Devuelva solamente las letras consonantes. Por ejemplo, si recibe 'algoritmos' o 'logaritmos' debe devolver 'lgrtms'.
- b) Devuelva solamente las letras vocales. Por ejemplo, si recibe 'sin consonantes' debe devolver 'i ooae'.
- c) Reemplace cada vocal por su siguiente vocal. Por ejemplo, si recibe 'vestuario' debe devolver 'vistaerou'.
- d) Indique si se trata de un palíndromo. Por ejemplo, 'anita lava la tina' es un palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda).

SOLUCIÓN:

EJERCICIO 1

```
print 'PROGRAMA DE ASTROLOGIA'
print 'Por favor ingrese el dia y mes de su nacimiento'
dia=int(raw_input('ingrese su dia de nacimiento:'))
mes=int(raw_input('ingrese su mes de nacimiento:'))
#declaro las restricciones para que no se ingrese una fecha invalida
if mes<1 or mes>12:
    print 'mes invalido'
#se asigna las fechas correspondientes a cada signo,segun la tabla dada
#sin cionsiderar si es anio bisciesto
if mes==1:
    if dia>=1 and dia<=20:
        print "Ud. es capricornio"
    elif dia>=21 and dia<=31:
        print "Ud. es Acuario"
    elif dia>31 or dia<1:
        print 'fecha invalida'
if mes==2:
```

```
if dia>=1 and dia<=19:
    print "Ud. es Acuario"
elif dia>=20 and dia<=29:
    print "Ud. es Piscis "
elif dia>29 or dia<1:
```

EJERCICIO 2:

```
print'PROGRAMA PARA DETERMINAR EL MAXIMO COMUN DIVISOR (ALGORITMO DE EUCLIDES)'
m=int(raw_input('ingrese un numero:'))
n=int(raw_input('ingrese un numero:'))
#ingreso la restriccion para que los numeros sean difernetes de 0
if m != 0 and n != 0 :
    #se analiza m y n con el fin de saber cual es el mayor, y por lo tanto
    #cual sera el dividendo y cual el divisor
    if(m > n):
        dividendo = m
        divisor = n
    else:
        dividendo = n
        divisor = m
    try:
        #se procede a aplicar el algoritmo de Euclides en
        #codificacion de python
        while dividendo % divisor != 0 :
            resto = dividendo % divisor
            dividendo = divisor
            divisor = resto
        print 'Maximo de', m, 'y', n, 'es',divisor
    except:
```

```
print 'los dos numeros deben ser diferentes de 0'
```

```
else:
```

```
print 'los dos numeros deben ser diferentes de 0'
```

EJERCICIO 3 CON FOR:

```
print 'PROGRAMA PARA DETERMINAR LOS MULTIPLOS DE UN NUMERO X MENORES A UN NUMERO Y'
```

```
x=int(raw_input("ingrese el numero x:"))
```

```
y=int(raw_input("ingrese el numero y:"))
```

```
#programa con for
```

```
def multi(x,y):
```

```
    if x>y: #x debe ser menor a y para que existan multiplos
```

```
        print 'no hay multiplos de',x,'menores a',y
```

```
        print 'porque',x,'es mayor a',y
```

```
    else:
```

```
        cadena=[] #en este conjunto vacio se van a almacenar los multiplos
```

```
        n=0 # n es el contador de la cantidad de multiplos menores a y
```

```
        for i in range(x,y,x):
```

```
            n+=1
```

```
            cadena.append(i) #agrego todos mis numeros i a cadena
```

```
        cadena=str(cadena).replace('[',",") #elimino el los corchetes para que no
```

```
        cadena=str(cadena).replace(']',",") #me salgan en la pantalla
```

```
        print 'hay',n,'numero(s) multiplo(s) de',x,'menor(es) a',y,':'
```

```
        print cadena
```

```
multi(x,y)
```

```
#este programa resulta mas fácil hacerlo con el for, porque utiliza menos codificación
```

```
#que el while
```

EJERCICIO 3 CON WHILE:


```

print 'PROGRAMA PARA DETERMINAR LOS MULTIPLOS DE UN NUMERO X MENORES A UN NUMERO Y'

x=int(raw_input("ingrese el numero x:"))
y=int(raw_input("ingrese el numero y:"))

def multiplos(x):

    if x>y: #x debe ser menor a y para que existan multiplos

        print 'no hay multiplos de',x,'menores a',y

        print 'porque',x,'es mayor a',y

    else:

        i=x # i va a iniciar en x

        n=0 # n es el contador de la cantidad de multiplos menores a y

        cadena=[]#en este conjunto vacio se van a almacenar los multiplos

        while i < y:

            cadena.append(i)

            n+=1

            i+=x

        cadena=str(cadena).replace('[','') #elimino el los corchetes para que no

        cadena=str(cadena).replace(']', '') #me salgan en la pantalla

        print'hay',n,'numero(s) multiplo(s) de',x,'menor(es) a',y

        print cadena

multiplos(x)

```

EJERCICIO 4:

```

print 'PROGRAMA PARA DETERMINAR SI UNA FECHA INGRESADA EN FORMATO dia/mes/año ES VALIDA'

dia=int(raw_input('ingrese un dia:'))

mes=int(raw_input('ingese un mes:'))

año=int(raw_input('ingrese un año:'))

if mes<1 or mes>12: # restriccion de los meses invalidos

```

```

    print 'fecha invalida'

#condiciones para que un anio sea bisiesto
if (anio%4==0 and not anio%100==0) or anio%400==0:
    # en el caso de que sea anio bisiesto
    if mes==1 or mes==3 or mes==5 or mes==7 or mes==8 or mes==10 or mes==12:
        if dia>=1 and dia<=31:
            print 'el',dia,'/',mes,'/',anio,'es una fecha valida'
        else:
            print 'el',dia,'/',mes,'/',anio,'es una fecha invalida'
    if mes==2:
        if dia>=1 and dia<=29:
            print 'el',dia,'/',mes,'/',anio,'es una fecha valida'
        else:
            print 'el',dia,'/',mes,'/',anio,'es una fecha invalida'
    if mes==4 or mes==6 or mes==9 or mes==11:
        if dia>=1 and dia<=30:
            print 'el',dia,'/',mes,'/',anio,'es una fecha valida'
        else:
            print 'el',dia,'/',mes,'/',anio,'es una fecha invalida'
else:
    if mes==1 or mes==3 or mes==5 or mes==7 or mes==8 or mes==10 or mes==12:
        if dia>=1 and dia<=31:
            print 'el',dia,'/',mes,'/',anio,'es una fecha valida'
        else:
            print 'el',dia,'/',mes,'/',anio,'es una fecha invalida'
    if mes==2:
        if dia>=1 and dia<=28:
            print 'el',dia,'/',mes,'/',anio,'es una fecha valida'
        else:

```



```

    print 'el',dia,'/',mes,'/',anio,'es una fecha invalida'
if mes==4 or mes==6 or mes==9 or mes==11:
    if dia>=1 and dia<=30:
        print 'el',dia,'/',mes,'/',anio,'es una fecha valida'
    else:
        print 'el',dia,'/',mes,'/',anio,'es una fecha invalida'

```

EJERCICIO 5:

```

print 'PROGRAMA PARA DETERMINAR EL VALOR TOTAL A PAGAR EN UNA COMPRA Y DAR VUELTO (DE
SER EL CASO)'

#funcion que va a permitir redondear valores luego
def redondear(n):
    return (int(n*100))/100.0

x=float(raw_input('ingrese el valor total de la compra:'))
iva=x*0.12
total=redondear(x+ iva)

print 'valor a cancelar: $', total #total a cancelar con iva

from time import sleep
sleep(1)

print 'Por favor seleccione una forma de pago'
print '1). Efectivo'
print '2). Tarjeta de credito \n'

y=int(raw_input('ingrese la opcion:'))

while y!=1 and y!=2:
    print 'opcion incorrecta'
    y=int(raw_input('ingrese de nuevo la opcion:'))

if y==1:
    print 'Ha seleccionado pagar en efectivo, por lo que tiene'
    print 'un descuento del 5% en el valor ingresado de la compra (antes del calculo del iva)'
    descuento=x-(x*0.05) #descuento antes de calcular el iva

```

```

i=redondear(descuento + descuento*0.12) #total a pagar con el 5% de descuento

print 'su total a pagar es:',i

z=float(raw_input('ingrese el dinero: $'))#pago del cliente

while z<i: # en el caso de que pague menos del total

    print 'Cantidad insuficiente'

    z=float(raw_input('ingrese el dinero: $'))

if z==i:

    print 'Gracias por su compra, regrese pronto...'

if z>i:

    vuelto=(z-i)

    print 'su vuelto es $',vuelto,'desglosado en:'

    print 'espere...'

    sleep(0.1)

    #con esta funcion se calcula el vuelto para cada denominacion

    def vuelto(vuelto):

        denominaciones = [50, 20, 10, 5,1]

        for denominacion in denominaciones:

            sleep(0.5)

            #segun el valor del vuelto se calcula la cantidad de billetes y monedas para

            #cada denominacion

            print '%d billete(s) de $%d' % ((vuelto / denominacion), denominacion)

            vuelto = vuelto % denominacion

            centavos=((vuelto)-int(vuelto))*100 #multiplico por 100 para poner valores como 50 ctvs

            cent=[50,25,10,5,1]          #y no 0.50 ctvs

            for monedas in cent:

                sleep(0.5)

                print '%d moneda(s) de %d' % ((centavos/monedas),monedas),'ctv(s)'

                centavos= centavos % monedas

                sleep(0.2)

```

```

        print 'Gracias por su compra, regrese pronto...'

    vuelto(vuelto)

if y==2:

    print 'Gracias por su compra, regrese pronto...'

```

EJERCICIO 6:

```

print 'PROGRAMA QUE PERMITE AL USUARIO INGRESAR n NUMEROS HASTA QUE INGRESE -1'

print 'Y DESPUES CALCULAR CUANTOS NUMEROS FUERON INGRESADOS, SU SUMA Y SU PROMEDIO'

x=int(raw_input('ingrese un numero:'))

n=0 # n es la cantidad de numeros ingresados

suma=x #la suma inicia con el primer numero ingresado

while x!=-1:

    x=int(raw_input('ingrese un numero:'))

    n+=1

    suma+=x

    #para que el -1 no altere a la suma y promedio a la suma le agrego +1

    promedio=(suma+1)/(float(n))

print 'ha ingresado',n,'numeros'

print 'la suma de los',n,'numeros','es:',(suma+1)

print 'el promedio de los',n,'numeros','es:',promedio

```

EJERCICIO 7:

```

print 'PROGRAMA QUE PIDA LA CONTRASENIA AL USUARIO, Y NO LE PERMITA CONTINUAR'

print 'HASTA QUE ESCRIBA LA CONTRASENIA CORRECTA(TIENE SOLO CINCO INTENTOS)'

usuario=raw_input('Ingrese su nombre de usuario:')

clave=raw_input('Ingrese su contrasenia: ')

def contrasenias(clave,usuario):

    contrasenia = '1994'

    intentos=1

    tiempo=0.1

```

```

while clave!=contrasenia:
    if intentos<5: #intentos permitidos
        tiempo+=1 # el tiempo se ira aumentando 1 segundo a cada intento
        intentos+=1
        print clave == contrasenia #valor booleano
        print 'Contrasenia incorrecta...'
        print 'espere para que pueda continuar...'
        from time import sleep
        sleep(tiempo)
        clave=raw_input('Ingrese nuevamente su contrasenia: ')
    if intentos==5:
        sleep(1)
        print 'Han terminado sus 5 intentos... no puede continuar'
        print 'vuelva a intentarlo mas tarde'
        break
if clave == contrasenia:
    print clave == contrasenia #valor booleano
    print 'Contrasenia correcta...Bienvenido al sistema',usuario
contrasenias(clave,usuario)

```

EJERCICIO 8:

```

print 'PROGRAMA QUE PERMITA AL USUARIO ADIVINAR UN NUMERO GENERADO AL AZAR'
import random
numero = random.randint(1,10) #genero el numero aleatorio
x=0
while x!=numero: #genero las condiciones para poder adivinar el numero
    x=int(raw_input('ingrese un numero:'))
    if x < numero:
        print 'el numero ingresado es muy bajo'

```

```

if x > numero:

    print 'el numero ingresado es muy alto'

if x==numero:

    print 'Felicitaciones has adivinado el numero que era el',numero

```

EJERCICIO 9:

```

print 'PROGRAMA QUE INGRESADO UN NUMERO LO MUESTRA CON SU SEPARACION POR MILES '
n=int(raw_input('ingrese el numero:'))

def separadigitos(n):

    x= "{:,.}" .format(n) #"{:,.}" separa los digitos con coma por lo que

    print x.replace(',','.')#reemplazo la coma por el punto para

separadigitos(n)#obtener el resultado correcto

```

EJERCICIO 10:

```

print 'PROGRAMA QUE INGRESADAS DOS CADENAS DE TEXTO DETERMINA SI LA SEGUNDA '
print 'CADENA ES SUBCADENA DE LA PRIMERA Y ADEMAS MOSTRARLAS SEGUN EL ORDEN ALFABETICO'
x=str(raw_input('ingrese una palabra:'))
y=str(raw_input('ingrese otra palabra:'))

if x.find(y)>=0: #con el find analizo si hay coincidencia

    print y,'es subcadena de',x#entre las dos cadenas

if x.find(y)<0:#para que y sea subcadena de x, x debe ser una cadena

    print y,'no es subcadena de',x#mas extensa

if x<y:

    print 'El orden de las cadenas segun el orden alfabetico es:',x,',',y

else:

    print 'El orden de las cadenas segun el orden alfabetico es:',y,',',x

```

EJERCICIO 11:

```
print 'PROGRAMA PARA TRANSFORMAR UN NUMERO BINARIO EN DECIMAL'

binario=raw_input('Ingrese el numero binario: ')

def bindecimal(binario):

    n=len(binario)#cantidad de digitos del binario

    valor=0

    for bit in binario:

        if bit == '1':

            valor = valor+2***(n-1)#formula para la transformacion

            n -=1#cuando termine de analizar un digito pasara al otro hasta terminar

    print 'el valor decimal de',binario,'es', valor

bindecimal(binario)
```

EJERCICIO 12:

```
print 'PROGRAMA QUE DADA UNA CADENA DE CARACTERES DEVUELVA SOLO LAS LETRAS CONSONANTES,'

print 'SOLO LAS LETRAS VOCALES, REEMPLACE LAS VOCALES POR SU SIGUIENTE VOCAL Y ADEMAS'

print 'DETERMINE SI ES PALINDROMO '

from time import sleep

sleep(1)

texto = raw_input('Inserte un texto: ')

def consonat(texto):

    vocales = 'aeiouAEIOU'

    for letra in vocales:

        texto = texto.replace(letra,"")#reemplazo las vocales por "",con lo que se eliminan

    print '>solo las letras consonantes son:', texto

consonat(texto)

sleep(1)

def vocal(texto):
```



```

consonantes = ('bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTUVWXYZ')
for letra in consonantes:#tambien elimino las consonantes al reemplazarlas
    texto = texto.replace(letra,")#por comillas
print '>>solo las vocales son:',texto
vocal(texto)
sleep(1)
def siguienteVocal(texto):
    vocales = 'aeiouAEIOU'
    for i in range(9,-1,-1):#el rango de la cantidad total de vocales,tomando en cuenta que inicia en 0
        if vocales[i]=='u':#al intentar cambiar 'u'por 'a' y 'U' por 'A' lo que hace es cambiarlo por 'e'
            texto = texto.replace(vocales[i],'1')# y 'E'ya que esa es la equivalencia de a
        elif vocales[i]=='U':#por lo que primero le cambio por un valor numerico,para que no altere las
            texto = texto.replace(vocales[i],'2')#letras de la cadena y luego procedo a cambiarle nuevamente
        else: #por 'a' y 'A'
            texto = texto.replace(vocales[i],vocales[i+1])# para las demas vocales si me funciona la formula
    texto = texto.replace('1','a')
    texto = texto.replace('2','A')
    print '>>el texto con las vocales intercambiadas es:',texto
siguienteVocal(texto)
sleep(1)
def palindromo(texto):
    # para determinar si es un palindromo primero invierto el texto para compararlo
    alrevez=texto[::-1]
    # luego elimino los espacios en ambas cadenas para ver si esque son iguales
    if alrevez.replace(' ','')== texto.replace(' ',''):
        print '>>La cadena si es un palindromo'
    else:
        print '>>La cadena no es un palindromo'
palindromo(texto)

```