



MANUAL DE PROGRAMACION

TECNOLOGICO JOSE MARIO MOLINA PASQUEL Y HENRIQUEZ
ZAPOTLANEJO



ESTRUCTURA DE DATOS

LESLIE DAREYA RUIZ BARRERA

Informática 3



Índice

REPASO - Programa 1	3
PRACTICA 2	4
PROGRAMA 3	5
PROGRAMA 4 – Tarea	7
PROGRAMA 5	10
PROGRAMA 6	12
PROGRAMA 7	14
PROGRAMA 8	15
PROGRAMA 9	17
REPASO 1	22
REPASO 2	23
REPASO 3	25
PRACTICA REPASO 3	27



REPASO - Programa 1

```
2  a = [10] #arreglo / array
3  b = [] # una lista no tiene tamaño es indefinida / a list has no fixed size, it is undefined
4
5  a[0] = 10
6  a[1] = 10
7
8  b={'hola',10,10.05,False, 'm' ,{1,2,3,4}}
9
10 #operadores logicos / logical operators
11 # + - *
12 # mod
13 # and or
14 # ** potencia o elevar a / power or exponentiation
15 # / exacta, con decimales / exact, with decimals
16 #// sin decimales / without decimals
17 # < > >= <= != not ==
18
19 #ciclos y condiciones / loops and conditions
20 if (len(a) > len(b)):
21     print('A es mayor') # A es mayor / A is greater
22 else:
23     print('B es mayor') # B es mayor / B is greater
24
```

Agregar un numero en cadena y mandarlo a llamar

```
1  print('Escribe un numero')
2  # Muestra un mensaje en pantalla / Displays a message on the screen
3
4  a = input('Escribe un numero')
5  # Pide al usuario que ingrese un valor y lo guarda en la variable 'a'
6  # / Asks the user to enter a value and stores it in the variable 'a'
7
8  print(a)
9  # Imprime en pantalla el valor que se guardó en la variable 'a'
10 # / Prints on the screen the value stored in the variable 'a'
```

```
Escribe un numero
Escribe un numero2
2
```



PRACTICA 2

Arreglos, Listas, Ciclos y condiciones

```
1  a = []
2  # Crea una lista vacía llamada a donde se guardarán los números válidos que el usuario introduce.
3  # Creates an empty list called a where the valid numbers entered by the user will be stored.
4  s = 0
5  # Inicia la variable s en 0. Se usará como acumulador para sumar los números guardados.
6  # Initializes the variable s to 0. It will be used as an accumulator to sum the stored numbers.
7  n = 0
8  # Contador n en 0. Cuenta cuántos números válidos se han guardado en a
9  # Counter n set to 0. Counts how many valid numbers have been stored in a
10 numeros = "0123456789"
11 # Cadena con los caracteres permitidos para considerar una entrada como número.
12 # String with the allowed characters to consider an entry as a number.
13 while(n < 10):
14     # Bucle while que se repite mientras n sea menor que 10
15     # While loop that repeats while n is less than 10
16     b = input('Escribe un numero: ')
17     # Pide al usuario que escriba algo y lo guarda como cadena en b
18     # Asks the user to enter something and stores it as a string in b
19     x = 0
20     # x servirá para contar cuántos caracteres de la cadena b son dígitos
21     # x will be used to count how many characters of the string b are digits
22     for i in b:
23         # RECORRE LOS ELEMENTOS B
24         # ITERATES THROUGH THE ELEMENTS OF b
25         if i in numeros:
26             x += 1
27             # Si el carácter i está en la cadena numeros se incrementa x
28             # If the character i is in the string numeros, x is incremented
29     if len(b) == x:
30         # Si son iguales, se considera la entrada válida como número entero.
31         # If they are equal, the entry is considered valid as an integer number.
32         a.append(int(b))
33         # Convierte la cadena b a entero con int(b) y la añade a la lista a.
34         # Converts the string b to an integer with int(b) and adds it to the list a.
35         n += 1
36         # Incrementa n porque se guardó un número válido.
37         # Increments n because a valid number was stored.
38     else:
39         print('El valor no es numero ')
40         # Si la entrada no está compuesta únicamente por dígitos, muestra el mensaje de error
41         # If the entry is not composed only of digits, displays the error message
```



PROGRAMA 3

Hacer un programa que lea 10 números y los almacene en un arreglo

```
1  a = [0,0,0,0,0,0,0,0,0,0]
2  # Crea una lista de 10 elementos inicializados en 0 / Creates a list of 10 elements initialized to 0
3
4  for i in range(0,10):
5      # Repite las instrucciones 10 veces, con 'i' desde 0 hasta 9
6      # / Repeats the instructions 10 times, with 'i' from 0 to 9
7      a[i] = int(input('Escribe un numero \n'))
8      # Pide al usuario un número, lo convierte a entero y lo guarda en la posición i de la lista
9      # / Asks the user for a number, converts it to integer, and stores it at position i in the list
10
11 for i in a:
12     # Recorre cada elemento de la lista / Iterates through each element in the list
13     print(i)
14     # Imprime el valor de cada elemento / Prints the value of each element
```

```
Escribe un numero
1
Escribe un numero
2
Escribe un numero
3
Escribe un numero
4
Escribe un numero
5
Escribe un numero
6
Escribe un numero
7
Escribe un numero
8
Escribe un numero
9
Escribe un numero
0
```

```
1
2
3
4
5
6
7
8
9
0
```



Opción 2 programa 3

```
1 array = []
2 # Crea una lista vacía llamada 'array'
3 # / Creates an empty list called 'array'
4
5 while True:
6     # Inicia un bucle infinito que seguirá ejecutándose hasta que se cumpla un 'break'
7     # / Starts an infinite loop that will continue until a 'break' is reached
8
9     contador = len(array)
10    # Guarda en 'contador' la cantidad de elementos que hay actualmente en la lista
11    # / Stores in 'contador' the number of elements currently in the list
12
13    entrada_usuario = input("Escribe un numero por favor: ")
14    # Pide al usuario que ingrese un valor y lo guarda en la variable 'entrada_usuario'
15    # / Asks the user to enter a value and stores it in 'entrada_usuario'
16
17    if entrada_usuario.isdigit():
18        # Verifica si lo ingresado por el usuario es un número entero positivo (solo dígitos)
19        # / Checks if the user input is a positive integer (digits only)
20        array.append(entrada_usuario)
21        # Si es un número válido, lo agrega al final de la lista
22        # / If valid, appends it to the end of the list
23    print(f"Lista completa: {array}")
24    # Muestra en pantalla cómo se ve la lista después de la posible adición
25    # / Shows on screen how the list looks after the possible addition
26
27    if contador >= 9:
28        # Si la cantidad de elementos antes de esta iteración era 10 o más, termina el bucle
29        # / If the number of elements before this iteration was 10 or more, it breaks the loop
30        break
```

```
Escribe un numero por favor: 1
Lista completa: ['1']
Escribe un numero por favor: 2
Lista completa: ['1', '2']
Escribe un numero por favor: 3
Lista completa: ['1', '2', '3']
Escribe un numero por favor: 4
Lista completa: ['1', '2', '3', '4']
Escribe un numero por favor: 5
Lista completa: ['1', '2', '3', '4', '5']
Escribe un numero por favor: 6
Lista completa: ['1', '2', '3', '4', '5', '6']
Escribe un numero por favor: 7
Lista completa: ['1', '2', '3', '4', '5', '6', '7']
Escribe un numero por favor: 8
Lista completa: ['1', '2', '3', '4', '5', '6', '7', '8']
Escribe un numero por favor: 9
Lista completa: ['1', '2', '3', '4', '5', '6', '7', '8', '9']
Escribe un numero por favor: 0
Lista completa: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '0']
```



PROGRAMA 4 – Tarea

Hacer un programa que lea 10 números, los almacene en una lista y luego los sume mostrando el resultado final

```
5  a = []
6  # Crea una lista vacía para guardar los números / Creates an empty list to store the numbers
7
8  s = 0
9  # Inicializa la variable 's' para almacenar la suma / Initializes the variable 's' to store the sum
10
11 n = 0
12 # Inicializa la variable 'n' para contar cuántos números válidos se han ingresado
13 # / Initializes 'n' to count how many valid numbers have been entered
14
15 numeros = "0,1,2,3,4,5,6,7,8,9";
16 # Define una cadena con los caracteres numéricos válidos
17 # / Defines a string with the valid numeric characters
18
19 while(n < 10):
20     # Bucle que se repite hasta que se hayan ingresado 10 números válidos
21     # / Loop that repeats until 10 valid numbers have been entered
22
23     b = input('Escribe un numero')
24     # Pide al usuario que ingrese un número y lo guarda en 'b'
25     # / Asks the user to enter a number and stores it in 'b'
26
27     x = 0
28     # Inicializa un contador 'x' para verificar si todos los caracteres son dígitos
29     # / Initializes a counter 'x' to check if all characters are digits
30
31     for i in b:
32         # Recorre cada carácter en la entrada 'b'
33         # / Iterates over each character in the input 'b'
```



```
34
35     # if (ord(i) >= 48 and ord(i) <=57):
36     if i in numeros:
37         # Verifica si el carácter está en la lista de números válidos
38         # / Checks if the character is in the list of valid numbers
39         x += 1
40         # Si es válido, aumenta el contador 'x'
41         # / If valid, increments the counter 'x'
42
43     if len(b) == x:
44         # Si todos los caracteres son dígitos, se considera un número válido
45         # / If all characters are digits, it is considered a valid number
46         a.append(int(b))
47         # Convierte la entrada a entero y la agrega a la lista 'a'
48         # / Converts the input to an integer and appends it to the list 'a'
49         n += 1
50         # Aumenta el contador de números válidos ingresados
51         # / Increments the count of valid numbers entered
52     else:
53         print('el valor no es un numero')
54         # Si algún carácter no es dígito, muestra un mensaje de error
55         # / If any character is not a digit, shows an error message
56
57     for i in a:
58         # Recorre cada número de la lista 'a' / Iterates over each number in the list 'a'
59         print(i)
60         # Imprime el número actual / Prints the current number
61         s += i
62         # Suma el número actual a la variable 's' / Adds the current number to the variable 's'
63
64     print(f'La suma es {s}')
65     # Imprime la suma total de los números ingresados / Prints the total sum of the entered numbers
66
```

```
Escribe un numero1
Escribe un numero2
Escribe un numero3
Escribe un numero4
Escribe un numero5
Escribe un numero6
Escribe un numero7
Escribe un numero8
Escribe un numero9
Escribe un numero0
1
2
3
4
5
6
7
8
9
0
La suma es 45
```

Opción 2 programa 4



```
3  lista = []
4  # Crea una lista vacía para almacenar los números ingresados
5  # / Creates an empty list to store the entered numbers
6
7  for e in range(0,9):
8      # Bucle que se repite 9 veces / Loop that repeats 9 times
9      numeros = input("Ingresa un numero: ")
10     # Pide al usuario que ingrese un número y lo guarda en 'numeros'
11     # / Asks the user to enter a number and stores it in 'numeros'
12     x = numeros
13     # Guarda el valor de 'numeros' en la variable 'x'
14     # / Stores the value of 'numeros' in the variable 'x'
15     lista.append(int(x))
16     # Convierte 'x' a entero y lo agrega a la lista
17     # / Converts 'x' to integer and appends it to the list
18     print(f"La lista ahora tiene estos elementos: {lista}")
19     # Muestra cómo se ve la lista después de agregar el número
20     # / Shows how the list looks after adding the number
21
22 suma_total = 0
23 # Inicializa la variable para almacenar la suma de los números
24 # / Initializes the variable to store the sum of the numbers
25
26 for e in lista:
27     # Recorre cada elemento de la lista / Iterates over each element in the list
28     suma_total = suma_total + e
29     # Suma cada elemento al total / Adds each element to the total
30
31 print(f"La suma total de todos los numeros es: {suma_total}")
32 # Imprime la suma total de los números ingresados / Prints the total sum of the entered numbers
```

```
Ingresa un numero: 1
La lista ahora tiene estos elementos: [1]
Ingresa un numero: 2
La lista ahora tiene estos elementos: [1, 2]
Ingresa un numero: 3
La lista ahora tiene estos elementos: [1, 2, 3]
Ingresa un numero: 4
La lista ahora tiene estos elementos: [1, 2, 3, 4]
Ingresa un numero: 5
La lista ahora tiene estos elementos: [1, 2, 3, 4, 5]
Ingresa un numero: 6
La lista ahora tiene estos elementos: [1, 2, 3, 4, 5, 6]
Ingresa un numero: 7
La lista ahora tiene estos elementos: [1, 2, 3, 4, 5, 6, 7]
Ingresa un numero: 8
La lista ahora tiene estos elementos: [1, 2, 3, 4, 5, 6, 7, 8]
Ingresa un numero: 9
La lista ahora tiene estos elementos: [1, 2, 3, 4, 5, 6, 7, 8, 9]
La suma total de todos los numeros es: 45
```



PROGRAMA 5

Hacer un programa que lea 10 datos, si el dato es un número se almacenará en un arreglo, si es un carácter o caracteres se meterá a una lista, cuando finalice el programa nos mostrará cuántos elementos numéricos y cuántos caracteres hay en cada estructura

```
6 arr = [-1,-1,-1,-1,-1,-1,-1,-1,-1,-1] # arreglo para almacenar números / array to store numbers
7 car = [] # lista para almacenar caracteres / list to store characters
8 c = 0 # contador para el arreglo / counter for the array
9 c2 = 0 # contador para números en el arreglo / counter for numbers in the array
10
11 while(True):
12     a = input('Escribe un dato o valor\n') # pedir al usuario que ingrese un valor / ask the user to enter a value
13     if a.isdigit(): # verificar si la entrada es un número / check if the input is a digit
14         arr[c] = int(a) # almacenar el número en el arreglo / store the number in the array
15     elif a.isalpha(): # verificar si la entrada es una letra o palabra / check if the input is alphabetic
16         car.append(a) # agregar la entrada a la lista de caracteres / add the input to the character list
17     c += 1 # incrementar el contador / increment the counter
18     if c >= 10: # verificar si el arreglo está lleno / check if the array is full
19         break # salir del bucle / exit the loop
20
21 print(f'la lista tiene {len(car)}') # imprimir cuántos caracteres hay en la lista
22 # / print how many characters are in the list
23
24 for i in arr: # recorrer el arreglo / loop through the array
25     if i != -1: # verificar si el valor es un número válido / check if the value is a valid number
26         c2 += 1 # incrementar el contador de números / increment the number counter
27
28 print(f'el arreglo tiene {c2}') # imprimir cuántos números hay en el arreglo / print how many numbers are in the array
29 print(car) # imprimir la lista de caracteres / print the list of characters
30 print(arr) # imprimir el arreglo de números / print the array of numbers
```

```
Escribe un dato o valor
4
Escribe un dato o valor
r
Escribe un dato o valor
8
Escribe un dato o valor
u
Escribe un dato o valor
k
Escribe un dato o valor
l
Escribe un dato o valor
b
Escribe un dato o valor
8
Escribe un dato o valor
n
la lista tiene 6
el arreglo tiene 4
['r', 'u', 'k', 'l', 'b', 'n']
[2, 4, -1, 8, -1, -1, -1, -1, 8, -1]
```



Opción 2 Programa 5

```
6 array = [] # arreglo para almacenar números / array to store numbers
7 lista = [] # lista para almacenar caracteres / list to store characters
8
9 while True:
10     entrada_usuario = input("Escribe un número o un caracter: ") # pedir al usuario un dato / ask the user for a value
11
12     if entrada_usuario.isdigit(): # verificar si la entrada es un número / check if the input is a number
13         array.append(entrada_usuario) # agregar el número al arreglo / add the number to the array
14         print("El número se agrego al array") # mensaje de confirmación / confirmation message
15     else:
16         lista.append(entrada_usuario) # agregar el caracter a la lista / add the character to the list
17         print("El caracter se agrego a la lista") # mensaje de confirmación / confirmation message
18
19     e_array = len(array) # contar elementos en el arreglo / count elements in the array
20     e_lista = len(lista) # contar elementos en la lista / count elements in the list
21     elementos = e_array + e_lista # total de elementos / total number of elements
22
23     if elementos >= 10: # verificar si se alcanzaron 10 elementos / check if 10 elements are reached
24         break # salir del bucle / exit the loop
25
26 cantidad_array = len(array) # cantidad final de números / final count of numbers
27 cantidad_lista = len(lista) # cantidad final de caracteres / final count of characters
28
29 print(f"La cantidad de elementos dentro del array son: {cantidad_array}") # mostrar cantidad de números
30 #/ show number count
31 print(array) # mostrar el arreglo / show the array
32
33 print(f"La cantidad de elementos dentro del lista son: {cantidad_lista}") # mostrar cantidad de caracteres
34 #/ show character count
35 print(lista) # mostrar la lista / show the list
```

```
El numero se agrego al array
Escribe un número o un caracter: h
El caracter se agrego a la lista
Escribe un número o un caracter: u
El caracter se agrego a la lista
Escribe un número o un caracter: i
El caracter se agrego a la lista
Escribe un número o un caracter: 9
El número se agrego al array
Escribe un número o un caracter: k
El caracter se agrego a la lista
Escribe un número o un caracter: 0
El número se agrego al array
Escribe un número o un caracter: g
El caracter se agrego a la lista
Escribe un número o un caracter: f
El caracter se agrego a la lista
La cantidad de elementos dentro del array son: 4
['1', '2', '9', '0']
La cantidad de elementos dentro del lista son: 6
['h', 'u', 'i', 'k', 'g', 'f']
```



PROGRAMA 6

Hacer un programa que lea 10 datos, si el dato es un número se almacenará en un arreglo, si es un carácter o caracteres se meterá a una lista, cuando finalice el programa nos mostrará cuántos elementos numéricos y cuántos caracteres hay en cada estructura. **Utilizando el método principal.**

```
2 def resultados():
3     c2 = 0 # contador para elementos numéricos en el arreglo
4     #/ counter for numeric elements in the array
5     print(f'la lista tiene {len(carc)}') # mostrar la cantidad de elementos en la lista
6     #/ show the number of elements in the list
7     for i in arr: # recorrer el arreglo / loop through the array
8         if i != -1: # verificar si el valor es válido / check if the value is valid
9             c2 += 1 # incrementar el contador / increment the counter
10    print(f'el arreglo tiene {c2}') # mostrar la cantidad de números en el arreglo
11    #/ show the number of numbers in the array
12    print(carc) # mostrar la lista de caracteres / show the character list
13    print(arr) # mostrar el arreglo de números / show the array of numbers
14
15
16 def hola(): # DEFINICIÓN DE MÉTODO O FUNCIÓN / FUNCTION DEFINITION
17     c = 0 # contador para el arreglo / counter for the array
18     while True: # CICLO INFINITO / INFINITE LOOP
19         a = input('Escribe un dato: ') # solicitar dato al usuario / request input from the user
20         if a.isdigit(): # VALIDAR DIGITOS / CHECK DIGITS
21             arr[c] = int(a) # guardar número en el arreglo / store number in the array
22         elif a.isalpha(): # VALIDAR LETRAS / CHECK LETTERS
23             carc.append(a) # agrega elemento al final de la lista / add element to the end of the list
24             c += 1 # incrementar contador / increment counter
25
26         if c >= 10: # verificar si se alcanzaron 10 elementos / check if 10 elements are reached
27             break # salir del bucle / exit the loop
28         resultados() # llamar a la función resultados / call the resultados function
29
30 carc = [] # lista para caracteres / list for characters
31 arr = [-1, -1, -1, -1, -1, -1, -1, -1, -1, -1] # arreglo inicializado con -1 / array initialized with -1
32
33 if __name__ == '__main__': # MÉTODO PRINCIPAL / MAIN METHOD
34     hola() # llamar a la función hola / call the hola function
```



```
Escribe un dato: hola
la lista tiene 1
el arreglo tiene 0
['hola']
[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1]
Escribe un dato: 64
la lista tiene 1
el arreglo tiene 1
['hola']
[-1, 64, -1, -1, -1, -1, -1, -1, -1, -1]
Escribe un dato: Leslie
la lista tiene 2
el arreglo tiene 1
['hola', 'Leslie']
[-1, 64, -1, -1, -1, -1, -1, -1, -1, -1]
Escribe un dato: 23
la lista tiene 2
el arreglo tiene 2
['hola', 'Leslie']
```



PROGRAMA 7

Hacer un programa que lea el nombre, edad y sexo de 5 personas. Estos elementos tienen que estar dentro de una lista

```
4 def inicio(): # DEFINICIÓN DE LA FUNCIÓN / FUNCTION DEFINITION
5     c = 0 # contador de personas / counter for people
6     while True: # CICLO INFINITO / INFINITE LOOP
7         n = input('Escribe el nombre') # solicitar nombre al usuario / ask the user for the name
8         e = input('Escribe la edad') # solicitar edad al usuario / ask the user for the age
9         s = input('Escribe el sexo') # solicitar sexo al usuario / ask the user for the gender
10        lis.append([n, e, s]) # agregar los datos como lista dentro de la lista principal
11        #/ add the data as a list inside the main list
12        c += 1 # incrementar el contador / increment the counter
13        if c >= 5: # verificar si ya se ingresaron 5 personas / check if 5 people have been entered
14            break # salir del bucle / exit the loop
15        print(lis) # mostrar la lista completa / print the full list
16
17 lis = [] # lista principal para almacenar los datos de las personas / main list to store people's data
18
19 if __name__ == "__main__": # MÉTODO PRINCIPAL / MAIN METHOD
20     inicio() # llamar a la función inicio / call the inicio function
```

Escribe el nombreLeslie
Escribe la edad19
Escribe el sexoF
Escribe el nombreMario
Escribe la edad15
Escribe el sexoM
Escribe el nombreJuan
Escribe la edad35
Escribe el sexom
Escribe el nombresara
Escribe la edad21
Escribe el sexof
Escribe el nombreerick
Escribe la edad17
Escribe el sexom
[['Leslie', '19', 'F'], ['Mario', '15', 'M'], ['Juan', '35', 'm'], ['sara', '21', 'f'], ['erick', '17', 'm']]



PROGRAMA 8

Hacer un programa que lea una cadena y muestre cuantas minúsculas, mayúsculas, números y espacios tiene

```
5 def inicio(): # DEFINICIÓN DE LA FUNCIÓN / FUNCTION DEFINITION
6     mi = 0 # contador de letras minúsculas / counter for lowercase letters
7     may = 0 # contador de letras mayúsculas / counter for uppercase letters
8     c = 0 # contador de números / counter for numbers
9     e = 0 # contador de espacios / counter for spaces
10    numeros = "0123456789" # cadena que contiene los dígitos / string containing digits
11    cadena = input('Escribe una cadena') # solicitar al usuario que ingrese una cadena / ask the user to enter a string
12    for i in cadena: # recorrer cada carácter de la cadena / loop through each character in the string
13        if i in numeros: # verificar si el carácter es un número / check if the character is a number
14            print('es numero') # mostrar mensaje si es número / display message if it is a number
15            c += 1 # incrementar contador de números / increment number counter
16        if i == ' ': # verificar si el carácter es un espacio / check if the character is a space
17            e += 1 # incrementar contador de espacios / increment space counter
18        if ord(i) >= 97 and ord(i) <= 122: # verificar si es minúscula / check if it is lowercase
19            mi += 1 # incrementar contador de minúsculas / increment lowercase counter
20        if ord(i) >= 65 and ord(i) <= 90: # verificar si es mayúscula / check if it is uppercase
21            may += 1 # incrementar contador de mayúsculas / increment uppercase counter
22    print(f'Los numeros son: {c}\n y los espacios: {e}\n y las minusculas: {mi}\n y las mayusculas: {may}\n')
23    # mostrar resultados / display results
24
25 if __name__ == "__main__": # MÉTODO PRINCIPAL / MAIN METHOD
26     inicio() # llamar a la función inicio / call the inicio function
27
```

```
• Escribe una cadenaHola
  Los numeros son: 0
  y los espacios: 0
  y las minusculas: 3
  y las mayusculas: 1
```



Opción 2 programa 8

```
3  entrada_usuario = input("Escribe una cadena: ")
4  # solicitar al usuario que ingrese una cadena de caracteres / ask the user to enter a string
5
6  lista_digitos = [] # lista para almacenar números / list to store digits
7  lista_mayus = [] # lista para almacenar letras mayúsculas / list to store uppercase letters
8  lista_mins = [] # lista para almacenar letras minúsculas / list to store lowercase letters
9  lista_espacios = [] # lista para almacenar espacios u otros caracteres / list to store spaces or other characters
10
11 for e in entrada_usuario: # recorrer cada carácter en la cadena / loop through each character in the string
12     if e.isdigit(): # verificar si es un número / check if it is a number
13         lista_digitos.append(e) # agregar número a la lista / add number to the list
14     elif e.isupper(): # verificar si es mayúscula / check if it is uppercase
15         lista_mayus.append(e) # agregar letra mayúscula a la lista / add uppercase letter to the list
16     elif e.islower(): # verificar si es minúscula / check if it is lowercase
17         lista_mins.append(e) # agregar letra minúscula a la lista / add lowercase letter to the list
18     else: # si no es número ni letra / if it is neither number nor letter
19         lista_espacios.append(e) # agregar a lista de espacios / add to spaces list
20
21 print(f"La palabra es: {entrada_usuario}") # mostrar la cadena ingresada / display the entered string
22 print("La cantidad de numeros es:", len(lista_digitos)) # mostrar cantidad de números / display number count
23 print("La cantidad de mayúsculas es:", len(lista_mayus)) # mostrar cantidad de mayúsculas / display uppercase count
24 print("La cantidad de minúsculas es:", len(lista_mins)) # mostrar cantidad de minúsculas / display lowercase count
25 # La comilla cuenta como espacio / The quotation mark counts as a space
26 print("La cantidad de espacios es:", len(lista_espacios)) # mostrar cantidad de espacios / display space count
27
```

```
Escribe una cadena: Hola Buenos Dias 7
La palabra es: Hola Buenos Dias 7
La cantidad de numeros es: 1
La cantidad de mayúsculas es: 3
La cantidad de minúsculas es: 11
La cantidad de espacios es: 3
```




PROGRAMA 9

Hacer un programa que almacene cadenas de caracteres en una lista, donde las cadenas no tengan espacios, solo la primera letra esté en mayúscula, contengan obligatoriamente todas las vocales, y que el programa no termine hasta que la lista tenga cinco elementos que cumplan estas condiciones.

```
7 def vocales(cad):
8     # Variables booleanas para cada vocal / Boolean variables for each vowel
9     ba = False
10    be = False
11    bi = False
12    bo = False
13    bu = False
14    # Verifica si la cadena contiene cada vocal (mayúscula o minúscula)
15    # Checks if the string contains each vowel (uppercase or lowercase)
16    if 'a' in cad or 'A' in cad:
17        ba = True
18    if 'e' in cad or 'E' in cad:
19        be = True
20    if 'i' in cad or 'I' in cad:
21        bi = True
22    if 'o' in cad or 'O' in cad:
23        bo = True
24    if 'u' in cad or 'U' in cad:
25        bu = True
26    # Si todas las vocales están presentes, se agrega la cadena a la lista
27    # If all vowels are present, add the string to the list
28    if ba == True and be == True and bi == True and bo == True and bu == True:
29        lista.append(cad)
30    # Imprime la lista acumulada de cadenas válidas
31    # Prints the accumulated list of valid strings
32    print(lista)
33
34
35
36 def minusculas(c1):
37     # Contador de caracteres en minúscula / Counter for lowercase characters
38     cm = 0
39     print(c1) # Imprime la cadena recibida / Prints the received string
40     # Recorre la cadena desde el segundo carácter (exceptuando el primero)
41     # Loops through the string starting from the second character
42     for i in c1[1:]:
```



```
42     for i in c1[1:]:
43         # Verifica si el carácter está en minúscula usando su valor ASCII
44         # Checks if the character is lowercase using its ASCII value
45         if ord(i) >=97 and ord(i)<=122:
46             cm += 1
47     # Si todos menos el primero son minúsculas
48     # If all except the first are lowercase
49     if cm == len(c1)-1:
50         print(f'la cadena son minusculas excepto la primera{cm}')
51         # Llama a la función vocales (aunque aquí pasa un número, debería ser la cadena)
52         # Calls the vowels function (though here it passes a number, should be the string)
53         vocales(cm)
54     else:
55         # Si no se cumple la condición
56         # If the condition is not met
57         print('Error la cadena no cumple')
58
59
60 def inicio():
61     # Contador de caracteres distintos de espacio
62     # Counter for characters different from space
63     ce = 0
64     nc = "" # Nueva cadena para depuración / New string for debugging
65     c = input("Escribe la cadena: \n") # Pide al usuario una cadena / Asks user for a string
66     # Recorre cada carácter de la cadena
67     # Loops through each character of the string
68     for i in c:
69         if ord(i) != 32: # Si no es espacio / If it's not a space
70             ce += 1
71
72     # Si no había espacios en la cadena
73     # If there were no spaces in the string
74     if ce == len(c):
75         if c.isalpha(): # Si la cadena solo tiene letras / If string contains only letters
76             # Devuelve la cadena con espacios de la primera
```



```
76         # Revisa si son minúsculas con excepción de la primera
77         # Checks if they are lowercase except the first one
78         minusculas(c)
79     else:
80         # Si no son todas letras, revisa carácter por carácter
81         # If not all are letters, check character by character
82         for i in c:
83             # Si es un número, lo omite / If it's a number, skip it
84             if ord(i) >= 48 and ord(i) <= 57:
85                 pass
86             else:
87                 # Si no es número, lo agrega a la nueva cadena
88                 # If not a number, add it to the new string
89                 nc += i
90         print(nc) # Imprime la nueva cadena parcial / Prints the new partial string
91         minusculas(nc) # Verifica minúsculas en la nueva cadena / Checks lowercase in new string
92     else:
93         # Si contiene espacios
94         # If contains spaces
95         print('Error en la cadena')
96
97     # Lista global para almacenar las cadenas válidas
98     # Global list to store valid strings
99     lista = []
100
101     if __name__ == "__main__":
102         # Ciclo infinito hasta que haya al menos 5 cadenas válidas
103         # Infinite loop until there are at least 5 valid strings
104         while(True):
105             inicio() # Llama al inicio / Calls the main function
106             if len(lista) >= 5: # Si ya hay 5 cadenas en la lista / If there are 5 strings in the list
107                 break # Rompe el ciclo / Breaks the loop
108
```



Opción 2 Programa 9

```
5 lista = []
6 # Lista global para almacenar las palabras válidas
7 # Global list to store valid words
8
9 def ejercicio(cadena):
10     # Borramos los espacios
11     # Remove spaces
12     x = cadena.replace(" ", "")
13
14     # Hacemos que toda la palabra sea en minúsculas y validamos vocales
15     # Convert the string to lowercase and validate vowels
16     y = x.lower()
17     vocales = {"a", "e", "i", "o", "u"}
18     # Conjunto de vocales
19     # Set of vowels
20     string_completa_set = set(y)
21     # Convertimos la cadena en un conjunto de caracteres únicos
22     # Convert the string into a set of unique characters
23     validar_vocales = vocales.issubset(string_completa_set)
24     # Verificamos si todas las vocales están presentes
25     # Check if all vowels are present
26
27     # Si las vocales están, transformamos la primera letra en mayúscula
28     # y agregamos la cadena formateada a la lista
29     # If vowels are present, capitalize the first letter and add the formatted string to the list
30     # En caso contrario, solicitamos otra palabra
31     # Otherwise, ask for another word
32     if validar_vocales == True:
33         string_formateada = y[0].upper() + y[1:]
34         # Primera letra en mayúscula, resto en minúscula
35         # First letter uppercase, rest lowercase
```



```
35     # First letter uppercase, rest lowercase
36     lista.append(string_formateada)
37     # Agregamos la cadena a la lista
38     # Add the string to the list
39     print(f"La palabra: {string_formateada} se agrego a la lista")
40     # Mostramos mensaje de éxito
41     # Show success message
42     print(f"La lista : {lista}")
43     # Mostramos la lista actual
44     # Show current list
45 else:
46     print(" Esta palabra no tiene todas las vocales Vuelve a escribir una que contenga todas las vocales:")
47     # Mensaje de error si faltan vocales
48     # Error message if vowels are missing
49     return None
50     # Retornamos None
51     # Return None
52
53
54 while len(lista) < 6:
55     # Bucle hasta que la lista tenga 6 elementos
56     # Loop until the list has 6 elements
57
58     # Pedimos la palabra al usuario
59     # Ask the user for a word
60     cadena = input("Ingresa Texto: ")
61
62     # Llamamos a la función para validar y agregar la palabra
63     # Call the function to validate and add the word
64     ejercicio(cadena)
65
```

```
Ingresa Texto: Hola
Esta palabra no tiene todas las vocales Vuelve a escribir una que contenga todas las vocales:
Ingresa Texto: aeiou
La palabra: Aeiou se agrego a la lista
La lista : ['Aeiou']
Ingresa Texto: 
```



REPASO 1

Hacer un programa que lea el nombre y el precio de un producto; el programa calculará el costo (que incluye un 12% adicional) y el precio de venta (que incluye IVA del 16%).

```
23 for i in range(1,5): # Repite 4 veces: desde 1 hasta 4 (el 5 no se incluye)
24     # Repeat 4 times: from 1 to 4 (5 is not included)
25     # precio = 12.55 # Inicialización de ejemplo, se sobrescribe después
26     # Example initialization, overwritten later
27     nombre = input('Escribe el nombre del producto\n')
28     # Pide al usuario el nombre del producto
29     # Ask the user for the product name
30     precio = float(input("Escribe el precio del producto\n"))
31     # Pide al usuario el precio del producto y lo convierte a flotante
32     # Ask the user for the product price and convert it to float
33     costo = precio * 1.12
34     # Calcula el costo agregando 12% (sobrepeso o gastos)
35     # Calculate the cost adding 12% (overhead or expenses)
36     precioventa = costo * 1.16
37     # Calcula el precio de venta agregando IVA del 16%
38     # Calculate the selling price adding 16% VAT
39     print(f'el costo es {costo:.2f} y el precio de venta {precioventa:.2f}\n')
40     # Imprime el costo y precio de venta con 2 decimales
41     # Print the cost and selling price with 2 decimals
42     print(f'el costo es {costo} y el precio de venta {precioventa}\n')
43     # Imprime el costo y precio de venta sin limitar decimales
44     # Print the cost and selling price without limiting decimals
45
46     # Código comentado para permitir al usuario continuar o terminar
47     # Commented code to allow the user to continue or stop
48     # res = input('Deseas otro numero (s/n) \n')
49     # if res == 'n' or res == 'N':
50     #     break
51
```

```
Escribe el nombre del producto
tortillas
Escribe el precio del producto
15
el costo es 16.80 y el precio de venta 19.49

el costo es 16.8 y el precio de venta 19.488

Escribe el nombre del producto
```



REPASO 2

Hacer un programa que tenga el funcionamiento de la formula general

```
5  a = 1
6  # Coeficiente a de la ecuación cuadrática
7  # Coefficient a of the quadratic equation
8  b = 2
9  # Coeficiente b de la ecuación cuadrática
10 # Coefficient b of the quadratic equation
11 c = -15
12 # Coeficiente c de la ecuación cuadrática
13 # Coefficient c of the quadratic equation
14 p = 0
15 # Variable para b al cuadrado (b^2)
16 # Variable for b squared (b^2)
17 m = 0
18 # Variable para 4*a*c
19 # Variable for 4*a*c
20 r = 0
21 # Variable para el discriminante
22 # Variable for the discriminant
23 ra = 0.0
24 # Variable para la raíz cuadrada del discriminante
25 # Variable for the square root of the discriminant
26 d = 0.0
27 # Variable para 2*a
28 # Variable for 2*a
29 x1 = 0.0
30 # Variable para la primera solución
31 # Variable for the first solution
32 x2 = 0.0
33 # Variable para la segunda solución
34 # Variable for the second solution
35
36 p = b**2
37 # Calculamos b al cuadrado
38 # Calculate b squared
39 m = a*c*4
```



```
40 # Calculamos 4*a*c
41 # Calculate 4*a*c
42 r = p - m
43 # Calculamos el discriminante b^2 - 4ac
44 # Calculate the discriminant b^2 - 4ac
45
46 if r > 0:
47     # Si el discriminante es positivo, existen soluciones reales
48     # If the discriminant is positive, real solutions exist
49     print('Si se puede')
50     # Indicamos que es posible calcular las soluciones
51     # Indicate that it is possible to calculate the solutions
52     ra = r **(1/2)
53     # Calculamos la raíz cuadrada del discriminante
54     # Calculate the square root of the discriminant
55     d = 2*a
56     # Calculamos 2*a para la fórmula general
57     # Calculate 2*a for the quadratic formula
58     x1 = (-b + ra)/ d
59     # Calculamos la primera solución usando la fórmula general
60     # Calculate the first solution using the quadratic formula
61     x2 = (-b - ra)/ d
62     # Calculamos la segunda solución usando la fórmula general
63     # Calculate the second solution using the quadratic formula
64     print(f'El valor de x1 es {x1:.2f} y de x2 {x2:.2f}')
65     # Mostramos los valores de las soluciones con 2 decimales
66     # Display the values of the solutions with 2 decimals
67 else:
68     # Si el discriminante no es positivo, no hay soluciones reales
69     # If the discriminant is not positive, there are no real solutions
70     print('No se puede')
71     # Indicamos que no es posible calcular soluciones reales
72     # Indicate that it is not possible to calculate real solutions
73
```

Si se puede

El valor de x1 es 3.00 y de x2 -5.00



REPASO 3

Hacer un programa que lea números y defina si es decimal o entero

```
4 def validar(a): # Definición de la función validar que recibe un parámetro 'a'
5     # Definition of the function validar that takes one parameter 'a'
6     c = 0 # Variable entera inicializada en 0
7     # Integer variable initialized to 0
8     d = 0.0 # Variable decimal inicializada en 0.0
9     # Float variable initialized to 0.0
10    try: # Intentar convertir 'a' en un número entero
11        # Try to convert 'a' into an integer
12        c = int(a)
13        print('Es un número entero sin decimales')
14        # It is an integer without decimals
15        print('Es un número entero')
16        # It is an integer
17    except ValueError: # Si ocurre un error, significa que no es entero
18        # If an error occurs, it means it's not an integer
19        try: # Intentar convertir 'a' en un número decimal (float)
20            # Try to convert 'a' into a decimal (float)
21            d = float(a)
22            print('Es un número entero con decimales')
23            # It is a number with decimals
24            print('Es un número decimal\n')
25            # It is a decimal number
26        except ValueError: # Si tampoco se puede, no es un número válido
27            # If it also fails, then it is not a valid number
28            print('No es un número válido')
29            # It is not a valid number
30
31    def leer(): # Función que pide al usuario un dato
32        # Function that asks the user for a value
33        a = input('Escribe un dato o valor\n')
34        # Read input from the user
35        validar(a)
36        # Call the function validar with the input
37    if __name__ == '__main__': # Punto de entrada principal del programa
38        # Main entry point of the program
39        leer()
40        # Call the function leer
```

```
Escribe un dato o valor
2
Es un número entero sin decimales
Es un número entero
```



```
Escribe un dato o valor
2.3
Es un número entero con decimales
Es un número decimal
```



PRACTICA REPASO 3

Hacer un programa que lea números enteros y los almacene en una lista. Si el número ingresado no es un entero, el programa volverá a pedirlo hasta que se ingrese correctamente. Los números se seguirán almacenando hasta que el usuario decida que no quiere agregar más datos.

```
8 def validar(a):
9     ne = 0
10    try:
11        ne = int(a) # Intenta convertir el dato a entero / Try to convert the input into an integer
12        return ne
13    except ValueError:
14        print('No es un entero') # Muestra mensaje si no es un número entero / Shows a message if it's not an integer
15    try:
16        nf = float(a) # Intenta convertir el dato a decimal / Try to convert the input into a float
17        return nf
18    except ValueError:
19        print('No es un numero con decimales') # Muestra mensaje si no es decimal / Shows a message if it's not a decimal
20    return a # Devuelve el dato original si no es número / Returns the original input if it's not a number
21
22 def leer():
23     a = input('Escribe un dato \n') # Pide al usuario un dato / Asks the user for an input
24     dato = validar(a) # Valida el dato ingresado / Validates the entered input
25     lista.append(dato) # Agrega el dato a la lista / Appends the data to the list
26
27 lista = [] # Lista vacía para guardar los datos / Empty list to store the data
28 if __name__ == '__main__':
29     while(True): # Ciclo infinito hasta que el usuario decida salir / Infinite loop until the user decides to exit
30         leer() # Llama a la función para leer y validar datos / Calls the function to read and validate data
31         res = input('Deseas otro s/n') # Pregunta si quiere continuar / Asks if the user wants to continue
32         if res == 'n' or res == 'N': # Si la respuesta es 'n' o 'N' termina / If response is 'n' or 'N', it ends
33             print(lista) # Muestra la lista de datos capturados / Prints the list of captured data
34             break # Rompe el ciclo y finaliza el programa / Breaks the loop and ends the program
35
```

```
Escribe un dato
1
Deseas otro s/nS
Escribe un dato
4
Deseas otro s/nN
[1, 4]
```