untitled3-2

May 25, 2023

[6]:
```
#1)Display "Hello World" in your output screen.

print("hello world")
```

```
hello world
```

[7]:
```
#2)Get the input from the user and perform addition of two numbers
a=int(input("enter the value of a"))
b=int(input("enter the value of b"))
c=a+b
print(c)
```

```
enter the value of a2
enter the value of b3
5
```

[10]:
```
#3)swap two variables without temp variable

a=int(input("enter the value of a:"))
b=int(input("enter the value of b:"))
a=a+b
a=a-b
b=a+b
print("the value of a is",a)
print("the value of b is",b)
```

```
enter the value of a:2
enter the value of b:4
the value of a is 2
the value of b is 6
```

[17]:
```
#4)convert the entered kilometres ( Convertion  Factor= 0.621371)

kilometer=int(input("enter the kilometer"))
conversionfactor=0.621371
a=kilometer*conversionfactor
print(a)
```

```
enter the kilometer4
2.485484
```

[24]:
```
#5)check whether the given number is positive, negative or 0

a=int(input("enter the value of a"))
if a>0:
    print("positive")
elif a==0:
    print("negative")
else:
    print("zero")
```

```
enter the value of a6
positive
```

[29]:
```
#6)verify that the given year is a leap year

year=int(input("enter the year"))
if (year%4==0)and(year%100!=0)or(year%400==0):
   print("leap year")
else:
    print("not leap year")
```

```
enter the year2000
leap year
```

[1]:
```
#7)display the prime numbers within the given interval

n = int(input("enter the number :"))
count = 0
for i in range(1,n+1):
    if n % i == 0 :
        count+=1
if count == 2 :
    print("its a prime number")
else:
    print("not a prime number")
```

```
enter the number :6
not a prime number
```

[2]:
```
#8) display the Fibonacci sequence up to n-th term

n = int(input("enter the number:")) #0,1,1,2,3,5,8.....
output = []
if n == 1:
    output.append(0)
```

```
    print(output)
elif n == 2:
    output.append(0)
    output.append(1)
    print(output)
else:
    output.append(0)
    output.append(1)
    a = 0
    b = 1
    sum  = 0
    for i in range(n):
        sum = a + b
        output.append(sum)
        a = b
        b = sum
    print(output)
```

```
enter the number:10
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

[3]:
```
#9) check if the number is an Armstrong number or not

n = input("enter the number :")
power = len(n)
output = 0
for i in n:
    a = int(i)**power
    output+=a
if output == int(n):
    print("it a Armstrong number")
else:
    print("its not a Armstrong number")
```

```
enter the number :153
it a Armstrong number
```

[4]:
```
#10) Find the Sum of natural numbers up to n-th term

n = int(input("enter the number : "))
output = 0
for i in range(n+1):
    output +=i
print(output)
```

```
enter the number : 10
55
```

[1]:
```
#11) Write a function called show_stars(rows). If rows are 5, it should print␣
 ↪the following

def show_stars(rows):
    for i in range(1, rows+1):
         print("*"*i)
show_stars(int(input("Enter your number: ")))
```

```
Enter your number: 9
*
**
***
****
*****
******
*******
********
*********
```

[2]:
```
# 12. New string from old string by removing
def remove_chars(str, n):
    return str[n:]
my_string = input("Enter your string:")
i=int(input("Enter the index number where u want to remove: "))
new_string = remove_chars(my_string, i)
print(new_string)
```

```
Enter your string:aaaaaajaa
Enter the index number where u want to remove: 6
jaa
```

[4]:
```
# 13. Numbers divisible by 5
numbers = [47,96,56,22,70,35,53,55,48,75,36]
print("The numbers divisible by 5 from the list are:")
for number in numbers:
     if number % 5 == 0:
            print(number)
```

```
The numbers divisible by 5 from the list are:
70
35
55
75
```

[8]:
```
# 14. HI Count
str=("Hi,This is my python assignment ,Hi, Hi , Hi ")
substr="Hi"
```

```
count=str.count(substr)
print("The count of the substring is : ",count)
```

```
The count of the substring is :  4
```

[9]:
```
# 15. Number Pattern
n=int(input("Enter the range: "))
for i in range(1, n+1):
    for j in range(i):
         print(i, end=" ")
    print()
```

```
Enter the range: 7
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7
```

[10]:
```
def palindrome(n):
    temp=n
    rev=0
    while(n>0):
        d=n%10
        rev=rev*10+d
        n=n//10
    if temp==rev:
        print("it is a palindrome number")
    else:
        print("it is not palindrom number")
n=int(input("Enter your number:"))
palindrome(n)
```

```
Enter your number:858
it is a palindrome number
```

[12]:
```
# 17. Swapping first and last element
my_list = [76,86,95,76,73,99,25,34]
print("Initial list: ")
print(my_list)
my_list[0], my_list[-1] = my_list[-1], my_list[0]
print("Updated list after swapping:")
print(my_list)
```

```
Initial list:
[76, 86, 95, 76, 73, 99, 25, 34]
```

```
Updated list after swapping:
[34, 86, 95, 76, 73, 99, 25, 76]
```

[13]:
```python
# 18. Swapping of two numbers in a list
my_list = [58,75,69,37,25,589]
print("The initial list is:")
print(my_list)
i1 =int(input("Enter i1:"))
i2 =int(input("Enter i2:"))
temp = my_list[i1]
my_list[i1] = my_list[i2]
my_list[i2] = temp
print("The Updated list is:")
print(my_list)
```

```
The initial list is:
[58, 75, 69, 37, 25, 589]
Enter i1:2
Enter i2:3
The Updated list is:
[58, 75, 37, 69, 25, 589]
```

[14]:
```python
# 19. Length of the list
my_list = [46,79,53,75,56,498,53]
print("My list elements: ")
print(my_list)
length = len(my_list)
print("The total length of my list is: ")
print(length)
```

```
My list elements:
[46, 79, 53, 75, 56, 498, 53]
The total length of my list is:
7
```

[15]:
```python
# 20. Maximum of two numbers
a=int(input("Enter A: "))
b=int(input("Enter B: "))
if (a>b):
 print("A is greater")
else:
 print("B is greater")
```

```
Enter A: 10
Enter B: 5
A is greater
```

[16]:
```
# 21. Minimum of two numbers
a=int(input("Enter A: "))
b=int(input("Enter B: "))
if (a<b):
    print("A is smaller")
else:
    print("B is smaller")
```

```
Enter A: 87
Enter B: 45
B is smaller
```

[17]:
```
# 22. Palindrome and Symmetricity of a srting
my_string = input("Enter the string:")
symmetrical = my_string == my_string[::-1]
palindrome = my_string == "".join(reversed(my_string))
if symmetrical:
    print("The string is symmetrical")
else:
    print("The string is not symmetrical")
if palindrome:
    print("The string is a palindrome")
else:
    print("The string is not a palindrome")
```

```
Enter the string:racecar
The string is symmetrical
The string is a palindrome
```

[18]:
```
# 23. Reversing of string
my_string = "Python Programming"
print("My initial string is:")
print(my_string)
words = my_string.split()
words.reverse()
new_string = " ".join(words)
print("My reversed string is:")
print(new_string)
```

```
My initial string is:
Python Programming
My reversed string is:
Programming Python
```

[19]:
```
# 24. Removing of index
my_string = "Hello, World!"
index_to_remove =int(input("Enter the index number to be removed:"))
new_string = my_string[:index_to_remove] + my_string[index_to_remove+1:]
```

```
print(new_string)
```

```
Enter the index number to be removed:7
Hello, orld!
```

[20]:
```
# 25. Length of the string
my_string = "This is my program"
string_length = len(my_string)
print("Length of my string is:")
print(string_length)
```

```
Length of my string is:
18
```

[22]:
```
# 26. Python code to print even length words in string
print("Enter your string:")
n=input()
s=n.split(" ")
print("The even indexed strings are:")
for i in s:
 #checking the length of words
 if len(i)%2==0:
        print(i)
```

```
Enter your string:
hi , how are you ??
The even indexed strings are:
hi
??
```

[23]:
```
# 27. Python Tuple Size
import sys
# Define a tuple
my_tuple = (19,9,3,'hi','there')
# Get the size of the tuple in bytes
size = sys.getsizeof(my_tuple)
# Print the size in bytes
print(f"The size of the tuple is {size} bytes")
```

```
The size of the tuple is 80 bytes
```

[24]:
```
# 28. Max and Min elements of a list
import heapq
def find_k_largest_smallest_elements(k, my_tuple):
    # Find the k largest elements using the nlargest function
    largest_elements = heapq.nlargest(k, my_tuple)
    # Find the k smallest elements using the nsmallest function
    smallest_elements = heapq.nsmallest(k, my_tuple)
```

```
    return largest_elements, smallest_elements
my_tuple = (55,595,262,962,858,25,2562,52,6)
k=int(input("Enter no. of elements needed:"))
largest, smallest = find_k_largest_smallest_elements(k, my_tuple)
print(f"The {k} largest elements in the tuple are: {largest}")
print(f"The {k} smallest elements in the tuple are: {smallest}")
```

```
Enter no. of elements needed:6
The 6 largest elements in the tuple are: [2562, 962, 858, 595, 262, 55]
The 6 smallest elements in the tuple are: [6, 25, 52, 55, 262, 595]
```

[25]:
```
# 29. Sum of tuple elements
my_tuple=(16,132,53, 44, 56)
print("Tuple=",my_tuple)
sum_of_tuple = sum(my_tuple)
print("The sum of my tuple elements is:", sum_of_tuple)
```

```
Tuple= (16, 132, 53, 44, 56)
The sum of my tuple elements is: 301
```

[1]:
```
# 30. Addition of row matrix
matrix = ((18,25,32),(47,55,36),(71,58,99))
print("My row matrix:",matrix)
print("The sum of each row matrix is:")
for row in matrix:
    row_sum = sum(row)
    print(row_sum)
```

```
My row matrix: ((18, 25, 32), (47, 55, 36), (71, 58, 99))
The sum of each row matrix is:
75
138
228
```

[2]:
```
# 31)Create a list of tuples from given list having number and its cube in each␣
 ↪tupl

def cubeoflist(li):
    result=[(num, num**3) for num in li]
    return result
li = [3, 4, 1, 2]
print(cubeoflist(li))
```

```
[(3, 27), (4, 64), (1, 1), (2, 8)]
```

[3]:
```
#32)Python | Sort Python Dictionaries by Key or Value
myDict = {'ravi': 10, 'rajnish': 9,
        'sanjeev': 15, 'yash': 2, 'suraj': 32}
```

```
myKeys = list(myDict.keys())
myKeys.sort()
sorted_dict = {i: myDict[i] for i in myKeys}

print(sorted_dict)
```

```
{'rajnish': 9, 'ravi': 10, 'sanjeev': 15, 'suraj': 32, 'yash': 2}
```

[4]:
```
#33)Python dictionary with keys having multiple inputs
dic = {}

a,b,c= 5, 3, 10

p,q,r= 12, 6, 9
dic["x-y+z"] = [a-b+c,p-q+r]
print(dic)
```

```
{'x-y+z': [12, 15]}
```

[7]:
```
#34)Python program to find the sum of all items in a dictionary
dic={ 'x':455, 'y':223, 'z':300, 'p':908 }

print("Dictionary: ", dic)

#using sum() and values()
print("sum: ",sum(dic.values()))
```

```
Dictionary:  {'x': 455, 'y': 223, 'z': 300, 'p': 908}
sum:  1886
```

[8]:
```
#35)Python program to find the size of a Dictionary
import sys
dic1 = {"A": 1, "B": 2, "C": 3}
dic2 = {"Geek1": "Raju", "Geek2": "Nikhil", "Geek3": "Deepanshu"}
dic3 = {1: "Lion", 2: "Tiger", 3: "Fox", 4: "Wolf"}
print("Size of dic1: " + str(sys.getsizeof(dic1)) + "bytes")
print("Size of dic2: " + str(sys.getsizeof(dic2)) + "bytes")
print("Size of dic3: " + str(sys.getsizeof(dic3)) + "bytes")
```

```
Size of dic1: 232bytes
Size of dic2: 232bytes
Size of dic3: 232bytes
```

[9]:
```
#36)Find the size of a Set in Python
import sys
Set1 = {"A", 1, "B", 2, "C", 3}
Set2 = {"Geek1", "Raju", "Geek2", "Nikhil", "Geek3", "Deepanshu"}
```

```
Set3 = {(1, "Lion"), ( 2, "Tiger"), (3, "Fox")}
print("Size of Set1: " + str(sys.getsizeof(Set1)) + "bytes")
print("Size of Set2: " + str(sys.getsizeof(Set2)) + "bytes")
print("Size of Set3: " + str(sys.getsizeof(Set3)) + "bytes")
```

```
Size of Set1: 472bytes
Size of Set2: 472bytes
Size of Set3: 216bytes
```

[10]:
```
#37)Iterate over a set in Python
test_set = set("geEks")
for val in test_set:
    print(val)
```

```
E
k
s
g
e
```

[13]:
```
#38)Python - Maximum and Minimum in a Set
def MAX(sets):
    return (max(sets))


sets = set([8, 16, 24, 1, 25, 3, 10, 65, 55])
print(MAX(sets))

def MIN(sets):
    return (min(sets))


sets = set([8, 16, 24, 1, 25, 3, 10, 65, 55])
print(min(sets))
```

```
65
1
```

[15]:
```
#39)Python - Remove items from Set
languages = {'Python', 'Java', 'English'}


languages.remove('English')


print(languages)
```

```
{'Python', 'Java'}
```

[16]:
```
#40)Python - Check if two lists have atleast one element common

def common_data(list1, list2):
    result = False


    for x in list1:

        # traverse in the 2nd list
        for y in list2:

            # if one common
            if x == y:
                result = True
                return result

    return result

# driver code
a = [1, 2, 3, 4, 5]
b = [5, 6, 7, 8, 9]
print(common_data(a, b))

a = [1, 2, 3, 4, 5]
b = [6, 7, 8, 9]
print(common_data(a, b))
```

```
True
False
```

[17]:
```
#41)Python - Assigning Subsequent Rows to Matrix first row elements


test_list = [[5, 8, 9], [2, 0, 9], [5, 4, 2], [2, 3, 9]]


print("The original list : " + str(test_list))


res = {test_list[0][ele] :  test_list[ele + 1] for ele in range(len(test_list)␣
 ↪- 1)}

print("The Assigned Matrix : " + str(res))
```

```
The original list : [[5, 8, 9], [2, 0, 9], [5, 4, 2], [2, 3, 9]]
The Assigned Matrix : {5: [2, 0, 9], 8: [5, 4, 2], 9: [2, 3, 9]}
```

[22]:
```
#42)Adding and Subtracting Matrices in Python
import numpy as np
A = np.array([[1, 2], [3, 4]])
B = np.array([[4, 5], [6, 7]])

print("Printing elements of first matrix")
print(A)
print("Printing elements of second matrix")
print(B)


print("Addition of two matrix")
print(np.add(A, B))
print("Subtraction of two matrix")
print(np.subtract(A, B))
```

```
Printing elements of first matrix
[[1 2]
 [3 4]]
Printing elements of second matrix
[[4 5]
 [6 7]]
Addition of two matrix
[[ 5  7]
 [ 9 11]]
Subtraction of two matrix
[[-3 -3]
 [-3 -3]]
```

[20]:
```
#43)Python - Group similar elements into Matrix
from itertools import groupby
test_list = [1, 3, 5, 1, 3, 2, 5, 4, 2]
print("The original list : " + str(test_list))
res = [list(val) for key, val in groupby(sorted(test_list))]
print("Matrix after grouping : " + str(res))
```

```
The original list : [1, 3, 5, 1, 3, 2, 5, 4, 2]
Matrix after grouping : [[1, 1], [2, 2], [3, 3], [4], [5, 5]]
```

[23]:
```
#44)Python - Row-wise element Addition in Tuple Matrix

# initializing list
test_list = [[('Gfg', 3), ('is', 3)], [('best', 1)], [('for', 5), ('geeks', 1)]]

# printing original list
print("The original list is : " + str(test_list))
```

```
# initializing Custom eles
cus_eles = [6, 7, 8]

# Row-wise element Addition in Tuple Matrix
# Using enumerate() + list comprehension
res = [[sub + (cus_eles[idx], ) for sub in val] for idx, val in␣
 ↪enumerate(test_list)]

# printing result
print("The matrix after row elements addition : " + str(res))
```

```
The original list is : [[('Gfg', 3), ('is', 3)], [('best', 1)], [('for', 5),
('geeks', 1)]]
The matrix after row elements addition : [[('Gfg', 3, 6), ('is', 3, 6)],
[('best', 1, 7)], [('for', 5, 8), ('geeks', 1, 8)]]
```

[24]:
```
#45)Create an n x n square matrix, where all the sub-matrix has the sum of␣
 ↪opposite corner elements as even
import itertools


def sub_mat_even(n):

    temp = itertools.count(1)


    l = [[next(temp)for i in range(n)]for i in range(n)]


    if n%2 == 0:
        for i in range(0,len(l)):
            if i%2 == 1:
                l[i][:] = l[i][::-1]


    for i in range(n):
        for j in range(n):
            print(l[i][j],end=" ")
        print()

n = 4
sub_mat_even(n)
```

```
1 2 3 4
8 7 6 5
9 10 11 12
16 15 14 13
```

[1]:
```
#46)How to get list of parameters name from a function in Python?

def fun(a, b):
    return a**b

# import required modules
import inspect

# use signature()
print(inspect.signature(fun))
```

```
Object `Python` not found.
(a, b)
```

[2]:
```
#47)How to Print Multiple Arguments in Python?
def GFG(name, num="25"):
    print("Hello from", name + ', ' + num)


GFG("gfg")
GFG("gfg", "26")
```

```
Hello from gfg, 25
Hello from gfg, 26
```

[10]:
```
#48)Python program to find the power of a number using recursion
def exp(x,y):
    if(y==0):
        return 1
    else:
        return(x*exp(x,y-1))
n=int(input("Enter the first number:"))
m=int(input("Enter the second number:"))
print("Result=",exp(n,m))
```

```
Enter the first number:12
Enter the second number:12
Result= 8916100448256
```

[16]:
```
#49)Sorting objects of user defined class in Python
class GFG:
        def __init__(self, a, b):
                self.a = a
                self.b = b

        def __repr__(self):
                return str((self.a, self.b))
```

```
# list of objects
gfg = [GFG("geeks", 1),
        GFG("computer", 3),
        GFG("for", 2),
        GFG("geeks", 4),
        GFG("science", 3)]

# sorting objects on the basis of value
# stored at variable b
print(sorted(gfg, key=lambda x: x.b))
```

```
[('geeks', 1), ('for', 2), ('computer', 3), ('science', 3), ('geeks', 4)]
```

[17]:
```
#50)Functions that accept variable length key value pair as arguments
# using kwargs
# in functions


def printKwargs(**kwargs):
        print(kwargs)


# driver code
if __name__ == "__main__":
        printKwargs(Argument_1='gfg', Argument_2='GFG')
```

```
{'Argument_1': 'gfg', 'Argument_2': 'GFG'}
```

[ ]: