Q. Finding all the digits of a entered number

In [2]:

```
#sol 1 : when number of digits are known
%reset -f
num=int(input("enter a 3 digit integer \n"))
for i in list(range(0,3,1)):
    digit=num%10
    num=int(num/10)
    print(digit)
```

```
enter a 3 digit integer
968
8
6
9
```

In [25]:

```
#sol 2 : when number of digits are un known
%reset -f
num=int(input("enter an integer \n"))

#finding all the digits and number of digits of a number
def find_digits(int_num):
    num_of_digits=0
    all_digits=[]
    while int_num>0:
        digit=int_num%10
        int_num=int(int_num/10)

        num_of_digits += 1
        all_digits.append(digit)

    print(num_of_digits)
    print(all_digits)



find_digits(num)
```

```
enter an integer
153
3
None
```

Q. return the list as an output of function and input of function

In [ ]:

```
%reset -f
def inp_from_customer():

    input_nums=[]
    for i in list(range(0,5,1)):
        a=int(input("type your number :  \t "))
        input_nums.append(a)
    #print(input_nums)
    return input_nums

def sum_my_list(new_list):
    print("i am here")
    print(new_list[2])

my_returned_list=inp_from_customer()
print(my_returned_list)
sum_my_list(my_returned_list)
```

In [ ]:

```
```

18. Write a program that will check whether the number is armstrong number or not.

In [8]:

```
%reset -f

#finding all the digits and number of digits of a number
def find_digits(int_num):
    num_of_digits=0
    all_digits=[]
    while int_num>0:
        digit=int_num%10
        int_num=int(int_num/10)

        num_of_digits += 1
        all_digits.append(digit)

    #print(num_of_digits)
    #print(all_digits)
    return all_digits,num_of_digits   #important step

#to check if it is an armstrong number or not ?
def check_armstrong(x,y):
    internal_sum=0
    for i in list(range(0,y,1)):
        internal_sum=internal_sum+x[i]**y
    if internal_sum==num:
        print("Armstrong number")
    else:
        print("NOT a Armstrong number")

#function calls
num=int(input("enter an integer \n"))
a,b= find_digits(num)    #important step
check_armstrong(a,b)
```

```
enter an integer
153
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[8], line 31
     29 num=int(input("enter an integer \n"))
     30 #a,b= find_digits(num)    #important step
---> 31 check_armstrong(find_digits(num))

TypeError: check_armstrong() missing 1 required positional argument: 'y'
```

In [ ]:

```

```

20. Write a program that will give you the in hand salary after deduction of HRA(10%),DA(5%),PF(3%), and tax(if salary is between5-10 lakh–10%),(11-20lakh–20%),(20< _ – 30%)

In [9]:

```python
#function calculate tax
def cal_tax(sal):
    if sal >=5.0 and sal <10.0:
        return 0.1*sal
    elif sal >=10.0 and sal <20.0:
         return 0.2*sal
    elif sal >=20.0:
        return 0.3*sal
    else:
        return 0

net_sal=float(input("Enter your net salary \n"))
print("Your inhand salary is equals to " + str(0.82*net_sal - cal_tax(net_sal)))
```

```
Enter your net salary
16
Your inhand salary is equals to 9.919999999999998
```

21. Write a menu driven program - 1.cm to ft 2.kl to miles 3.usd to inr 4.exit

In [12]:

```
def what_to_convert():
    user_choice=int(input("Choose what you want to convert : \n1.cm to ft\n2.km to miles\n3.usd to inr\n4.exit\n"))
    if user_choice==1:
        cm_to_feet()
    else:
        print("NOT Possible today")

def cm_to_feet():
    in_cm=float(input("enter in cm"))
    print(0.0328084*in_cm)

what_to_convert()
```

```
Choose what you want to convert :
1.cm to ft
2.km to miles
3.usd to inr
4.exit
1
enter in cm65.789
2.1584318276000003
```

25.Write a program that can multiply 2 numbers provided by the user without using the * operator

In [17]:

```
a,b = int(input()),int(input())
sum=0
for i in list(range(0,b,1)):
    sum=sum+a

print(sum)
```

```
4
6
24
```

26.Write a program that can find the factorial of a given number provided by the user.

hint: factorial = n* n-1* n-2.....till n=1

In [18]:

```
%reset -f
def factorial(n):
    total=1
    while n>1:
        total=total * n
        n -= 1
    return(total)

num=int(input("enter the numner \n "))
factorial(num)
```

```
enter the numner
 5
```

Out[18]:

```
120
```

35.Print the first 20 numbers of a Fibonacci series.

0. 1. 1. 2. 3. 5. 8. 13. 21. 34. 55. 89. 144 . . . (Pendina)

In [ ]:

```
%reset -f
n=int(input("How many numbers do you need ? \n"))
present_num=1
previous_num=0
print(previous_num)
print("\t")
print(present_num)
print("\t")

for i in list(range(0,n-2,1)):
    present_num=
    print(present_num + previous_num)
    print("\t")
    previous_num = present_num
```

39.Print all factors of a given number provided by the user.

In [4]:

```
%reset -f
n=int(input("write your number ? \n"))
for i in list(range(1,n+1,1)):
    if n%i == 0:
        print(i)
```

```
write your number ?
14
1
2
7
14
```

40. reverse number entered by user

In [9]:

```
%reset -f
#finding all the digits and number of digits of a number
def find_digits(int_num):
    num_of_digits=0
    all_digits=[]
    while int_num>0:
        digit=int_num%10
        int_num=int(int_num/10)

        num_of_digits += 1
        all_digits.append(digit)

    #print(num_of_digits)
    #print(all_digits)
    return all_digits,num_of_digits   #important step

def reverse_digits(all_digits,num_of_digits):
    new_r_num=0
    for i in list(range(0,num_of_digits,1)):
        new_r_num=new_r_num + all_digits[i]*(10**(num_of_digits-i-1))
    print(new_r_num)

#function calls
num=int(input("enter an integer \n"))
a,b= find_digits(num)    #important step
reverse_digits(a,b)
```

```
enter an integer
69854
45896
```

41 star pattern (important to understand the range)

In [1]:

```python
num=int(input("number of stars at the end \n"))
for i in list(range(0,num,1)):
    #print("--")
    #print(i)
    print("\n")
    for j in list(range(0,i+1,1)):
        #print(j)
        print("*",end="")
```

```
number of stars at the end
7


*

**

***

****

*****

******

*******
```

42.star pattern (important to understand the range)

In [8]:

```python
num=int(input("number of stars at the middle \n"))

for i in list(range(0,num,1)):
    #print("--")
    #print(i)
    print("\n")
    for j in list(range(0,i+1,1)):
        #print(j)
        print("*",end="")
for i in list(range(num,0,-1)):
    #print("--")
    #print(i)
    print("\n")
    for j in list(range(i-1,0,-1)):
        #print(j)
        print("*",end="")
```

```
number of stars at the middle
5


*

**

***

****

*****

****

***

**

*
```

44. Write a program to print the following pattern (unfinished)

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1 2 3 4 5 4 3 2 1

In [ ]:

```
In [8]: %reset -f
        num=int(input("maximum number in the pattern \n"))

        for i in list(range(1,num+1,1)):
            #print("--")
            #print(i)
            print("\n")
            for j in list(range(1,i+1,1)):
                print(j,end=" ")
            for k in list(range(i-1,0,-1)):
                print(k,end=" ")
```

```
maximum number in the pattern
5


1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1 2 3 4 5 4 3 2 1
```

45. Write a program to print the following pattern

1

2 3

4 5 6

In [12]:

```
%reset -f
num=int(input("maximum number in the pattern \n"))
sum=0
for i in list(range(1,num+1,1)):
    print("\n")
    for j in list(range(1,i+1,1)):
        sum += 1
        print(sum,end=" ")
```

```
maximum number in the pattern
4


1 

2 3 

4 5 6 

7 8 9 10 
```

46. Write a program to calculate the sum of the following series till the nth term

1/1! + 2/2! + 3/3! + 4/4! +…….+ n/n!

In [ ]:

```

```

In [12]:

```
%reset -f
def factorial(n):
    total=1
    while n>1:
        total=total * n
        n -= 1
    return(total)

n=int(input("enter n \n "))
sum=0
for i in range(1,n+1,1):
    sum= sum + (i/factorial(i))

print(sum)
```

```
enter n
 10
2.7182815255731922
```

In [ ]:

46. Write a program to calculate the sum of the following series till the nth term

1 + x^2/2 + x^3/3 + … x^n/n

In [13]:

```
%reset -f
x=int(input("enter x \n "))
n=int(input("enter n \n "))
sum=1
for i in range(2,n+1,1):
    sum= sum + (x**i)/i

print(sum)
```

```
enter x
 2
enter n
 3
5.666666666666666
```

In [ ]:

49. Write a program that keeps on accepting a number from the user until the user enters Zero. Display the sum and average of all the numbers.

In [26]:

```
%reset -f
num=1
input_list=[]
while num!=0:
    num=int(input("enter a number \n "))
    if num==0:
        break
    input_list.append(num)

print(input_list)
print(sum(input_list))
type(input_list[1])
print(sum(input_list)/len(input_list))
```

```
enter a number
 1
enter a number
 2
enter a number
 3
enter a number
 4
enter a number
 0
[1, 2, 3, 4]
10
2.5
```

In [ ]:

```

```

50. Write a program that accepts 2 numbers from the user a numerator and a denominator and then simplifies it

Eg if the num = 5, den = 15 the answer should be ⅓

Eg if the num = 6, den = 9 the answer should be ⅔

In [8]:

```
%reset -f
def find_HCF(a,b):
    factors=[]
    larger_num=max(a,b)
    for i in range(larger_num,0,-1):
        if a%i==0 and b%i==0:
            factors.append(i)
    return max(factors)
num=int(input("Enter the numerator \n"))
den=int(input("Enter the denominator \n"))
a=find_HCF(num,den)
print("The simplified version is " + str(num/a) + "/" + str(den/a))
```

```
Enter the numerator
5
Enter the denominator
15
The simplified version is 1.0/3.0
```

51. Find the length of a given string without using the len() function

In [12]:

```
%reset -f
my_name="Whats up bro"
length=0
for i in my_name:
    length += 1

print(length)
```

```
12
```

52.Extract username from a given email. Eg if the email is nitish24singh@gmail.com (mailto:nitish24singh@gmail.com) then the username should be nitish24singh

In [13]:

```
%reset -f
my_id="chinmaya@gmail.com"
a=my_id.split("@")
print(a[0])
```

```
chinmaya
```

54.Find the index position of a particular character in another string

In [19]:

```
%reset -f
my_string="whats up brother, where were you all these days"
search_char=input("Which character you want to search ?")
for i in range(0,len(my_string),1):
    if my_string[i]==search_char:
        print(i)
```

```
Which character you want to search ?s
4
40
46
```

56.Write a program which can remove a particular character from a string

In [24]:

```
%reset -f
my_string="whats up brother, where were you all these days"
search_char=input("Which character you want to remove ?")
new_string=my_string.replace(search_char,"")
print(new_string)
#for i in range(0,len(my_string),1):
 #   if my_string[i]==search_char:
  #      print(i)
```

```
Which character you want to remove ?w
hats up brother, here ere you all these days
```

57.Write a program that can check whether a given string is palindrome or not. e.g. madam

In [29]:

```
%reset -f
count=0
inp_str=input("input your string \n")
total_length= len(inp_str)
for i in range(0,total_length):
    if inp_str[i]==inp_str[total_length-i-1]:
        count += 1

if count==total_length:
    print("Pallindrome")
else:
    print("NON Pallindrome")
```

```
input your string
reviver
Pallindrome
```

58.Write a python program to remove all the duplicates from a list

In [21]:

```
#use del list[i] to delete an item
%reset -f
my_list=[20,30,5,0,20,40,30,10]
length= len(my_list)
for i in range(0,length,1):
    for j in range (0,length,1):
        if my_list[i]==my_list[j] and i!=j:
            del my_list[j]

       # if i!=j and my_list[i]==my_list[j]:
        #    del my_list[i]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[21], line 7
      5 for i in range(0,length,1):
      6     for j in range (0,length,1):
----> 7         if my_list[i]==my_list[j] and i!=j:
      8             del my_list[j]

IndexError: list index out of range
```

In [ ]:

```
```