1. WAP to input 2 strings and swap the strings

In [23]:

```
s1=input("Enter fisrt string :")
s2=input("Enter second stirng :")
print("Before swaping :",s1,s2)
s1=s1+s2
s2=s1[0:(len(s1)-len(s2))]
print(s2)
s1=s1[len(s2):]
print(s1)
print("After swaping :",s1,s2)
```

```
Enter fisrt string :akriti
Enter second stirng :gupta
Before swaping : akriti gupta
akriti
gupta
After swaping : gupta akriti
```

2. WAP to generate 4 random numbers in the range 0-26 and print their average

In [22]:

```
import random

def get_rand_list():
    a=list()
    for i in range(0,5):
         a.append(random.randint(100,200))
    return a
r=get_rand_list()
print(r)
sum=0
for val in r:
    sum = sum+val
print("Addition is:",sum)
average=sum/len(r)
print("Average is :",average)
```

```
[131, 187, 194, 191, 189]
Addition is: 892
Average is : 178.4
```

3. WAP to generate and print a random uppercase or lowercase alphabet. Try these:

• Create a string containing all alphabets and then select a random alphabet. • Check the module string

In [23]:

```
import random
import string
#Giving length to genearate string
n=10
result=''.join(random.choice(string.ascii_uppercase)for i in range(n))
#to print string in lower case
#result=''.join(random.choice(string.ascii_lowercase)for i in range(n))
print("generated string is:",str(result))
string1=random.sample(result,5)
#string1=random.choice(result) gives single random character from the string
```

```
print(string1)
```

```
generated string is: ZUWWSRVPYW
['M', 'Z', 'F', 'B', 'Q']
```

WAF get_si() that takes Principle, Rate and Time as arguments and returns the Simple interest

Interest.

In [26]:

```
def get_si(p,r,t):
    return (p*r*t)/100
# simple_interest=get_si(1000,5,6)
# print(simple_interest)
p=input("Enterv Principal :")
r=input("Enter rate :")
t=input("Enter time :")
print()
```

```
300.0
```

In [32]:

```
def get_si(p,r,t):
    return (p*r*t)/100
# simple_interest=get_si(1000,5,6)
# print(simple_interest)
p=int(input("Enter Principal :"))
r=int(input("Enter rate :"))
t=int(input("Enter time :"))
simple_interest= get_si(p,r,t)
print(simple_interest)
```

```
Enter Principal :1800
Enter rate :5
Enter time :6
300.0
```

WAF get_amount() that takes Principle, Rate and Time as arguments and returns the Total amount using the get_si() function from above to calculate the SI. Also provide Rate = 10 and Time = 1 as default arguments.

In [40]:

```
def get_amount(p,r=10,t=1):
    si=p*r*t/100
    amount=si+p
    return amount
p=int(input("Enter Principal :"))
final_amount=get_amount(p)
print(final_amount)
```

```
Enter Principal :1800
1980.0
```

WAP get_ci() that takes Principle, Rate and Time as arguments and returns the Compound Interest.

In [42]:

```
def get_ci(p,r,t):
    return p*(pow((1+r/100),t))
p=int(input("Enter Principal :"))
r=int(input("Enter rate :"))
t=int(input("Enter time :"))
compound_interest= get_ci(p,r,t)
print(compound_interest)
```

```
Enter Principal :250000
Enter rate :36
Enter time :1
339999.99999999994
```

WAP get_q_r() taking 2 numbers as parameters and returns the quotient and remainder in the form of a tuple.

In [43]:

```
def get_q_r(x,y):
    a=x/y
    b=x%y
    return a,b
print(get_q_r(10,3))
```

```
(3.3333333333333335, 1)
```

WAP to find the length of hypotenuse of a right angled triangle, input the height and base from user.

In [1]:

```
h=float(input("Enter height of right angled triangle: "))
print(h)
b=float(input("Enter base of right angled triangle: "))
print(b)
def hypo(h,b):
    hy=0
    hy=(h**2)+(b**2)

    return hy

ht=hypo(h,b)
print(ht)
print("Length of hypotenuse",ht**0.5)
```

```
Enter height of right angled triangle: 4
4.0
Enter base of right angled triangle: 3
3.0
25.0
Length of hypotenuse 5.0
```

WAP to input number of seconds and print in days, hours, minutes and seconds

In [12]:

```
seconds = int(input("Enter Seconds :"))
def time(sec):
    mint = secondss/60
    hour = mint/60
    days = hour/24
    print(int(days),"day",int(hour)," hour",int(mint)," minute",int(seconds),"seconds")
```

```
time(seconds)
```

```
Enter Seconds :10000
0 day 0  hour 166  minute 10000 seconds
```

In [16]:

```
x=2
x*=3
print("**",x)
y=-x
x=x%4
print("::",x)

print(x,y)
```

```
** 6
:: 2
2 -6
```

In [17]:

```
def funct():
    pass
print(funct())
```

```
None
```

Convert a Tuple t = (1,2,3,4,5) to a list

In [20]:

```
t=(1,2,3,4,5)
ls=list(t)
print(ls)
```

```
[1, 2, 3, 4, 5]
```

WAP to join a list and a tuple:

L = [1,3,5,7] T = (8,6,4,2) Store the result in a list LS

In [25]:

```
L=[1,3,5,7]
T=(8,6,4,2)
TL=list(T)
print("Tuple elements in list:",TL)
LS=L+TL
print("All elements :",LS)
```

```
Tuple elements in list: [8, 6, 4, 2]
All elements : [1, 3, 5, 7, 8, 6, 4, 2]
```

Print the list in reverse order

l = ['a', 'd', 'c', 'A', 'C']

In [26]:

```
l=['a','b','c','A','C']
ls=l[::-1]
print(ls)
```

```
['C', 'A', 'c', 'b', 'a']
```

Print Elements at Odd indexes from a list (Do not use loop)

l = [10,11,20, 21,30, 31, 40, 41]

In [28]:

```
l = [10,11,20, 21,30, 31, 40, 41]
ls=l[1::2]
print(ls)
```

```
[11, 21, 31, 41]
```

How many ways you can copy a list.

In [30]:

```
l=[1,21,23,56,45]
ls.append(l)
print("By append ",ls)

new_list=l.copy()
print("By copy function: ",new_list)
new_list2=l[:]
print("By slicing :",new_list2)
```

```
[11, 21, 31, 41, [1, 21, 23, 56, 45], [1, 21, 23, 56, 45]]
** [1, 21, 23, 56, 45]
[1, 21, 23, 56, 45]
```

In [32]:

```
n=["Happy",[2,0,1,5]]
print(n[0][1])
print(n[1][3])
```

```
a
5
```

In [35]:

```
odd=[2,4,6,8]
odd[0]=1
print(odd)
odd[1:4]=[3,7,4]
print(odd)
```

```
[1, 4, 6, 8]
[1, 3, 7, 4]
```

In [36]:

```
odd=[2,4,6,8]
odd.append([11,13])
print(odd)
odd.extend([9,7])
print(odd)
```

```
[2, 4, 6, 8, [11, 13]]
[2, 4, 6, 8, [11, 13], 9, 7]
```

In [39]:

```
t=tuple('string')
print(t)
print(t[::1])
print(t[::2])
print(t[::2][::-3])
```

```
('s', 't', 'r', 'i', 'n', 'g')
('s', 't', 'r', 'i', 'n', 'g')
('s', 'r', 'n')
('n',)
```

In [41]:

```
t=([10,20,30,40,50,60,70])
print(60 in t)
print('60' in t)
print(t.count(10))
print(t.index(60))
print(len(t))
```

```
True
False
1
5
7
```

Write a program to input a string and print if it is palindrome or not.

In [47]:

```
s=input("Enter a string :")
print(len(s))
if s[::1]==s[::-1]:
    print("String is pallindrom")
else:
    print("Not pallindrome",)
print("Because the string is:",s[::-1])
```

```
Enter a string :hii
3
Not pallindrome
Because the string is: iih
```

Use the range method and create a tuple containing the following values:

(20, 15, 10, 5)

In [64]:

```
import random
t=tuple(range(20,0,-5))
print(t)
```

```
(20, 15, 10, 5)
```

WAP to convert string to list of characters.

In [3]:

```
def split(word):
```

```
    return list(word)
word='String'
print(split(word))
```

```
['S', 't', 'r', 'i', 'n', 'g']
```

In [ ]:

```
print(type('1 2'.split()))
print(type([1,3,2].sort()))
print(type('abc2'.upper()))
print(type(i in [1,2,3])
```

WAP to print first n natural numbers (input n from user).

In [12]:

```
ip=int(input("Enter a number:"))
for i in range(1,ip+1):
    print(i,end='\n')
#print(ip)
```

```
Enter a number:10
1
2
3
4
5
6
7
8
9
10
```

WAP to find sum of first n natural numbers

In [15]:

```
ip=int(input("Enter a number:"))
sum=0
for i in range(1,ip+1):
    sum=sum+i
print(sum)
```

```
Enter a number:10
55
```

WAP to print first n natural numbers in reverse order.

In [17]:

```
ip=int(input("Enter a number:"))
for i in range(1,ip+1)[::-1]:
    print(i,end='\n')
```

```
Enter a number:10
10
9
8
7
6
5
4
```

```
3
2
1
```

WAP to input a number and print its factorial.

In [22]:

```
from functools import reduce
ip=int(input("enter a number :"))
def factorial(n):
    if n==0:
        return 1
    else:
        return reduce(lambda x,y:x*y,range(1,n+1))
print(factorial(ip))
```

```
enter a number :5
120
```

WAP to print Fibonacci sequence till n.

In [30]:

```
from functools import reduce
ip=int(input("Enter a number:"))
fib = lambda n: reduce(lambda x,_: x+[x[-1]+x[-2]],range(n-2), [0, 1])
print(fib(ip))
```

```
Enter a number:5
[0, 1, 1, 2, 3]
```

WAP to print all digits of a number input from user.

In [45]:

```
def digit(n):
    for i in n:
        print(int(i))
n=input("Enter number:")
digit(n)
```

```
Enter number:12598
1
2
5
9
8
```

WAP to find sum of following series given n as input from user

1 + 2! + 3! + 4! + 5! + ….n! Where n! denotes the factorial of number n.

def fact_sum(n): sum = 0 fact=1 for i in range(1, n+1): sum += fact(i) return sum ip=int(input()) fact_sum(ip)

WAP to input base and exponent and print result without using inbuilt function pow(use for or while loop)

In [60]:

```
b = int(input("Enter base :"))
e = int(input(" Enter exponent : "))
power = 1

for i in range(1, e+ 1):
    power = power * n

print(power)
```

```
Enter base :5
 Enter exponent : 3
125
```

In [ ]:

```

```