```
# square of 2 num in tuples
list1 = [1, 2, 5, 6]
res = [(val, pow(val, 2)) for val in list1]
print(res)
```

```
[(1, 1), (2, 4), (5, 25), (6, 36)]
```

+ Code + Text

```
#program for ncr
print("Enter the Value of n: ", end="")
n = int(input())
print("Enter the Value of r: ", end="")
r = int(input())
fact = i = 1
while i<=n:
  fact = i*fact
  i += 1

numerator = fact
sub = n-r
fact = i = 1

while i<=sub:
  fact = i*fact
  i += 1

denominator = fact
fact = i = 1
while i<=r:
  fact = i*fact
  i += 1

denominator = fact*denominator
comb = numerator/denominator
print("\nCombination (nCr) =", comb)
```

```
Enter the Value of n: 4
Enter the Value of r: 4

Combination (nCr) = 1.0
```

```
#max and min value in dictionary
my_dict = {'x':500, 'y':5874, 'z': 560}

key_max = max(my_dict.keys(), key=(lambda k: my_dict[k]))
key_min = min(my_dict.keys(), key=(lambda k: my_dict[k]))

print('Maximum Value: ',my_dict[key_max])
print('Minimum Value: ',my_dict[key_min])
```

```
Maximum Value:  5874
Minimum Value:  500
```

```
 # Python prog to find elements in a listof largest and smallest
def find_len(list1):
    length = len(list1)
    list1.sort()
    print("Largest element is:", list1[length-1])
    print("Smallest element is:", list1[0])
    print("Second Largest element is:", list1[length-2])
    print("Second Smallest element is:", list1[1])
list1=[12, 45, 2, 41, 31, 10, 8, 6, 4]
Largest = find_len(list1)
```

```
Largest element is: 45
Smallest element is: 2
Second Largest element is: 41
Second Smallest element is: 4
```

```
# Python program to count uppercase and lowercase characters
string = input('Enter any string: ')
upper, lower = 0, 0
for i in string:
    if(i.islower()):
        lower = lower + 1
    elif(i.isupper()):
        upper = upper + 1
print('Lowercase characters:',lower)
print('Uppercase characters:',upper)
```

```
Enter any string: manaswINI
Lowercase characters: 6
Uppercase characters: 3
```

```
color = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']
color = [x for (i,x) in enumerate(color) if i not in (0,4,5)]
print(color)
```

```
['Green', 'White', 'Black']
```

```
n=int(input("Input a number : "))
d ={}
for x in range(1,n+1):
    d[x]=x*x
print(d)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-17-574bd4a669f0> in <module>()
----> 1 n=int(input("Input a number : "))
      2 d ={}
      3 for x in range(1,n+1):
      4     d[x]=x*x
      5 print(d)

TypeError: 'str' object is not callable
```

SEARCH STACK OVERFLOW

```
#1 and last should be same
def match_words(words):
  ctr = 0

  for word in words:
    if len(word) > 1 and word[0] == word[-1]:
      ctr += 1
  return ctr

print(match_words(['abc', 'xyz', 'aba', '1221']))
```

```
2
```

```
#print more than  2 characters
test_list = [(4, 5), (4, ), (8, 6, 7), (1, ), (3, 4, 6, 7)]
print("The original list : " + str(test_list))
K = 1
res = [ele for ele in test_list if len(ele) != K]
print("Filtered list : " + str(res))
```

```
The original list : [(4, 5), (4,), (8, 6, 7), (1,), (3, 4, 6, 7)]
Filtered list : [(4, 5), (8, 6, 7), (3, 4, 6, 7)]
```

```
# Function to replace all occurrences of AB with C
def replaceABwithC(input, pattern, replaceWith):
    return input.replace(pattern, replaceWith)
if __name__ == "__main__":
    input = 'helloABworld'
    pattern = 'AB'
    replaceWith = 'C'
    print (replaceABwithC(input,pattern,replaceWith))
```

```
helloCworld
```

```
def countOccurrences(str, word):
    wordslist = list(str.split())
    return wordslist.count(word)
str = "have a good day"
word = "good"
print(countOccurrences(str, word))
```

```
1
```

```
def match_words(words):
  ctr = 0
  for word in words:
    if len(word) > 1 and word[0] == word[-1]:
      ctr += 1
  return ctr
print(match_words(['abc', 'xyz', 'aba', '1221']))
```

```
2
```

```
#remove key from dictionary
myDict = {'a':1,'b':2,'c':3,'d':4}
print(myDict)
if 'a' in myDict:
    del myDict['a']
print(myDict)
```

```
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
{'b': 2, 'c': 3, 'd': 4}
```

```
# Python code t get difference of two lists

def Diff(li1, li2):
    return list(set(li1) - set(li2)) + list(set(li2) - set(li1))
li1 = [10, 15, 20, 25, 30, 35, 40]
li2 = [25, 40, 35]
print(Diff(li1, li2))
```

```
[10, 20, 30, 15]
```

```
# product of 2 numbers
def product(num1,num2):
    if(num1<num2):
        return product(num2,num1)
    elif(num2!=0):
         return(num1+product(num1,num2-1))
    else:
         return 0

num1=int(input("Enter first number: "))
num2=int(input("Enter second number: "))
```

```
print("product is: ",product(num1,num2))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-25-2a035ce5340e> in <module>()
      8         return 0
      9 
---> 10 num1=int(input("Enter first number: "))
     11 num2=int(input("Enter second number: "))
     12 

TypeError: 'str' object is not callable
```

SEARCH STACK OVERFLOW

```
# Python program to display the Fibonacci sequence
def recur_fibo(n):
   if n <= 1:
       return n
   else:
       return(recur_fibo(n-1) + recur_fibo(n-2))
nterms =8
if nterms <= 0:
   print("Plese enter a positive integer")
else:
   print("Fibonacci sequence:")
   for i in range(nterms):
       print(recur_fibo(i))
```

```
Fibonacci sequence:
0
1
1
2
3
5
8
13
```

```
#gcd using functions
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
GCD=gcd(a,b)
```

```
print("GCD is: ")
print(GCD)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-27-a3e69b1be24e> in <module>()
      4     else:
      5         return gcd(b,a%b)
----> 6 a=int(input("Enter first number:"))
      7 b=int(input("Enter second number:"))
      8 GCD=gcd(a,b)

TypeError: 'str' object is not callable
```

SEARCH STACK OVERFLOW

```
#gcd using functions
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)
a=int(input("Enter first number:"))
b=int(input("Enter second number:"))
GCD=gcd(a,b)
print("GCD is: ")
print(GCD)
```

```
#replace bb with cc
animal='rabbit'
print(animal.replace('bb','cc'))
print(animal)
```

```
raccit
rabbit
```

```
# product of 2 numbers
def product(num1,num2):
    if(num1<num2):
        return product(num2,num1)
    elif(num2!=0):
         return(num1+product(num1,num2-1))
    else:
         return 0

num1=int(input("Enter first number: "))
num2=int(input("Enter second number: "))
```

```
print("product is: ",product(num1,num2))
```

✓ 8s completed at 17:26 ● ✕