Let's try writing some simple programs:

Write a program to calculate the area of a triangle:

In [ ]:
```
a = 5
b = 7
c = 8
area = a+b+c
print("Area of the triangle is:", area)
```

Write a program to convert Kilometers to Miles:

In [ ]:
```
print("Enter distance in km: ")
km = float(input())
conv_fac = 0.621371
miles = km * conv_fac
print("The distance is:", miles, "miles.")
```

Write a program to calculate the sum of numbers from 1 to n:

In [ ]:
```
n = int(input())
number_sum = 0
for i in range(1,n+1):
    number_sum = number_sum + i
print("Sum of numbers from 1 to n is:", number_sum)
```

Can you change the code above to calculate the sum of numbers from m to n?

In [ ]:
```
m = int(input())
n = int(input())
number_sum = 0
for i in range(m,n+1):
    number_sum = number_sum + i
print("Sum of numbers from m to n is:", number_sum)
```

Write a program to count the number of times a sub_string has been repeated in a string:

In [ ]:
```
sub_str = "a"
main_str = "London bridge is falling down, falling down, my fair lady."
counter = 0
for char in main_str:
    if char == sub_str:
        counter += 1
print("sub str", sub_str, "has been repeated ", counter, "times.")
```

In [ ]:
```
sub_str = "falling"
main_str = "London bridge is falling down, falling down, my fair lady."
words = main_str.split()
counter = 0
for word in words:
    if word == sub_str:
        counter += 1
print("sub str '", sub_str, "' has been repeated ", counter, "times.")
```

Write a program to caculate the average of a list of numbers:

In [ ]:
```
nums = [5,7,14,17,8,9,19,25]
numbers_sum = 0
for num in nums:
    numbers_sum = numbers_sum +  num
average = numbers_sum / len(nums)
print("Number average is:", average)
```

In [ ]:
```
#better way to do the above:
nums = [5,7,14,17,8,9,19,25]
average = sum(nums)/len(nums)
print("Number average is:", average)
```

Write a program to count the numbers in a list that are above the list's average:

In [ ]:
```
nums = [5,7,14,17,8,9,19,25]
average = sum(nums)/len(nums)
print(average)
counter = 0
for num in nums:
    if num>average:
        counter += 1
print(counter)
```

In [ ]:
```

```

Python Program to Check Prime Number
Python Program to Print the Fibonacci sequence
Python Program to Check Whether a String is Palindrome or Not
Python Program to Count The number of occurances of each character in a string.

In [ ]:
```
n = 10000000000000111111
flag = True

for i in range(2, int((n**(1/2)))+1):
    if n%i == 0:
        flag = False
        break

if flag == True:
    print(n, "is prime.")
else:
    print(n, "is not prime.")
```

In [ ]:
```

```

In [ ]:
```

```

55 79 103

In [ ]:
```

```

In [ ]:
```
input_str = input().split()
print(input_str)
input_numbers = []
for num in input_str:
    input_numbers.append(int(num))
print(input_numbers)
```

In [ ]:
```
input_numbers = list(map(int, input().split()))
```

In [ ]:
```
print(input_numbers)
```

In [ ]:
```
a,b = (2,3)
```

In [ ]:
```
print(a)
```

In [ ]:
```
print(b)
```

45 67

In [ ]:
```
n,k = map(int, input().split())
```

In [ ]:
```
print(n)
print(k)
```

In [ ]:
```
def f(n):
    if n<=1:
        return 1
    return n * f(n-1)
```

1,1,2,3,5,8,13,21,...

In [ ]:
```
def fib(n):
    if n==1 or n==2:
        return 1
    else:
        return fib(n-1) + fib(n-2)
```

In [ ]:
```

```

In [ ]:
```
import time
```

In [ ]:
```
tic = time.time()
fib(50)
tac = time.time()
run_time = tac - tic
print(run_time)
```

In [ ]:
```
fibs = [-1] * 1000
```

In [3]:
```
def fib(n):
    first_number = 1
    second_number = 1
    third_number = -1
    for i in range(n-2):
        third_number = first_number + second_number
        first_number = second_number
        second_number = third_number
    return third_number
```

In [8]:
```
print(fib(1000))
```

```
43466557686937456435688527675040625802564660517371780402481729089536555417949051890403879840079255169
29592259308032263477520968962323987332247116164299644090653318793829896964992851600370447613779516684
9228875
```

In [9]:
```
x = 5
def f():
    global x
    x = x+1
    return x
print(x)
print(f())
```

```
5
6
```

In [ ]:
```

```