In [1]:
```
def num():
    for i in range(1,26):
        print(i)
num()
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```

In [4]:
```
num=int(input("Enter a number:",))
def evenOdd(num):
    if num%2== 0:
        print(num,'is an even number')
    else:
        print(num,'is an odd number')
evenOdd(num)
```

```
Enter a number:12
12 is an even number
```

In [33]:
```
Data=["name","id","company_name"]
Data1=["shikha","1","oakiand_system"]
d={}
for i in range(len(Data)):
    d[Data[i]]=Data1[i]
print(d)
```

```
{'name': 'shikha', 'id': '1', 'company_name': 'oakiand_system'}
```

In [31]:
```
d={"susmitha":1}
print(d)
newDict={value:key for key,value in d.items()}
print(newDict)
```

```
{'susmitha': 1}
{1: 'susmitha'}
```

In [21]:
```
dictionary={"susmitha":01}
reversed_dictionary={}
for key,value in dictionary.items():
    reversed_dictionary[value]=key
print(reversed_dictionary)
```

```
  Cell In[21], line 1
    dictionary={"susmitha":01}
                           ^
SyntaxError: leading zeros in decimal integer literals are not permitted; use an 0o prefix for octal integers
```

In [27]:
```
power=lambda x,y:x**y  #lambda arg:expressions
print(power(2,4))
```

```
16
```

In [29]:
```
power=lambda a,b:a**b
for i in range(1,5):
    print(power)
```

```
<function <lambda> at 0x0000022A8E490A40>
<function <lambda> at 0x0000022A8E490A40>
<function <lambda> at 0x0000022A8E490A40>
<function <lambda> at 0x0000022A8E490A40>
```

In [34]:
```
def num(x,y):
    return x,y
x=num(1,6)
```

```
print(x)
```

```
(1, 6)
```

In [35]:
```
 a= lambda x:x+10
print(a(5))
```

```
15
```

In [40]:
```
price=12
txt="The mangoes are {}"
print(txt.format(price))
```

```
The mangoes are 12
```

In [42]:
```
#[12:47] Pallavi raut

price=12
txt="The mangoes are {}"
print(txt.format(price))
```

```
The mangoes are 12
```

In [43]:
```
for i in range(1,26):
    if(i%2==0):
        print(i,"is even")
    else:
        print(i,"is odd")
```

```
1 is odd
2 is even
3 is odd
4 is even
5 is odd
6 is even
7 is odd
8 is even
9 is odd
10 is even
11 is odd
12 is even
13 is odd
14 is even
15 is odd
16 is even
17 is odd
18 is even
19 is odd
20 is even
21 is odd
22 is even
23 is odd
24 is even
25 is odd
```

In [18]:
```
input_string="ABC1.23DE7.8F43"
s1=s2=s3=''
for i in input_string:
    if i.isalpha():
        s1 +=i
    elif:
         i.isdigit():
            s2+=i
    else:
         i.isdigit():
            s3+=i

print(s2+s1+s3)
```

```
  Cell In[18], line 6
    elif:
        ^
SyntaxError: invalid syntax
```

In [26]:
```
d={1:'a', 2:'b', 3:'c', 4:'d'}
d.update({3:"d"})
print(d)
```

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

In [23]:
```
list=[1,2,3,4,5,6,7,8,9,]
del list[,]
print(list)
```

```
  Cell In[23], line 2
    del list[,]
             ^
SyntaxError: invalid syntax
```

In [61]:
```
number=int(input("Enter any number:"))
```

```
x=lambda num:1 if num <= 1 else num*x(num-1)
print('%d != %d'%(number,x(number)))
```

```
Enter any number:5
5 != 120
```

In [63]:
```
list=[1]+[i for i in range(1,10)]
list[1:10]=[list[i-1]* i for i in range(1,10)]
print(list)
```

```
[1, 1, 2, 6, 12, 20, 30, 42, 56, 72]
```

In [13]:
```
def fact(n):
 list=[fact(x) for x in range(10)]
print(list)
fact(10)
```

```
<class 'list'>
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Cell In[13], line 4
      2  list=[fact(x) for x in range(10)]
      3 print(list)
----> 4 fact(10)

Cell In[13], line 2, in fact(n)
      1 def fact(n):
----> 2  list=[fact(x) for x in range(10)]

Cell In[13], line 2, in <listcomp>(.0)
      1 def fact(n):
----> 2  list=[fact(x) for x in range(10)]

Cell In[13], line 2, in fact(n)
      1 def fact(n):
----> 2  list=[fact(x) for x in range(10)]

Cell In[13], line 2, in <listcomp>(.0)
      1 def fact(n):
----> 2  list=[fact(x) for x in range(10)]

    [... skipping similar frames: <listcomp> at line 2 (1484 times), fact at line 2 (1484 times)]

Cell In[13], line 2, in fact(n)
      1 def fact(n):
----> 2  list=[fact(x) for x in range(10)]

Cell In[13], line 2, in <listcomp>(.0)
      1 def fact(n):
----> 2  list=[fact(x) for x in range(10)]

RecursionError: maximum recursion depth exceeded
```

In [14]:
```
import math
[math.factorial(n) for n in range(10)]
```

Out[14]:
```
[1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880]
```

In [55]:
```
input_list=[1,2,3,4,5,6,7,8,9,]
for i in range(1,10):
    for j in range(1,i+1):
        print(j,end=" ")
print()
print(input_list)
```

```
1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 3 4 5 6 1 2 3 4 5 6 7 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 9
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [2]:
```
f1=open("hello.txt","w")
f1.write("Welcome to python programming")
f1.close()
```

In [3]:
```
import os
os.getcwd()
```

Out[3]:
```
'C:\\Users\\Sindu\\Untitled Folder 2\\Untitled Folder'
```

In [4]:
```
f1=open("hello.txt","w")
f1.write("Welcome to python programming")
f1.close()
f1=open("hello.txt","r")
print(f1.read())
```

```
Welcome to python programming
```

In [5]:
```
f1=open("hello.txt","w")
f1.write("Welcome to python programming")
f1.close()
```

```
f1=open("hello.txt","r")
print(f1.read(5))
```

```
Welco
```

In [6]:
```
f1=open("hello.txt","w")
f1.write("Welcome to python programming")
f1.close()
f1=open("hello.txt","r")
print(f1.read(5))
print(f1.read(4))
```

```
Welco
me t
```

In [7]:
```
f1=open("hello.txt","w")
f1.write("Welcome to python programming")
f1.close()
f1=open("hello.txt","r")
print(f1.read())
f1.close()
```

```
Welcome to python programming
```

In [9]:
```
f1=open("mlines.txt","w")
lines=["welcome\n","python\n","programming"]
f1.writelines(lines)
f1.close()
```

In [12]:
```
f1=open("mlines.txt","w")
lines=["welcome\n","python\n","programming"]
f1.writelines(lines)
f1.close()
f1=open("mlines.txt","r")
print(f1.readline())
f1.close()
```

```
welcome
```

In [13]:
```
f1=open("mlines.txt","w")
lines=["welcome\n","python\n","programming"]
f1.writelines(lines)
f1.close()
f1=open("mlines.txt","r")
print(f1.readline())
print(f1.readline())
f1.close()
```

```
welcome

python
```

In [14]:
```
f1=open("mlines.txt","w")
lines=["welcome\n","python\n","programming"]
f1.writelines(lines)
f1.close()
f1=open("mlines.txt","r")
print(f1.readlines())
f1.close()
```

```
['welcome\n', 'python\n', 'programming']
```

In [15]:
```
f1=open("mlines.txt","w")
lines=["welcome\n","python\n","programming"]
f1.writelines(lines)
f1.close()
f1=open("mlines.txt","a")
f1.write("\nhello")
f1.close()
```

In [6]:
```
f1=open("mlines.txt","r")
print(f1.read())
```

```
welcome
python
programming
hello
```

In [17]:
```
#swap two variables
x=5
y=10
x,y=y,x
print("x=",x)
print("y=",y)
```

```
x= 10
y= 5
```

In [18]:
```
import calendar
```

```
import calendar
yy=2023
mm=8
print(calendar.month(yy, mm))
```

```
    August 2023
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

In [35]:
```
n=int(input("enter a number:"))
if n>0:
    print("postive number")
elif n==0:
    print("zero")
else:
    print("negitive number")
```

```
enter a number:7
postive number
```

In [36]:
```
data={1:"pallavi",2:"shiksha",3:"vaishal"}
print(data)
```

```
{1: 'pallavi', 2: 'shiksha', 3: 'vaishal'}
```

In [ ]:
```
```

In [29]:
```
data={1:"pallavi",2:"shiksha",3:"vaishal"}#reverse dictionary
print(data)
newdict={v:k for k,v in data.items()}
print(newdict)
```

```
{1: 'pallavi', 2: 'shiksha', 3: 'vaishal'}
{'pallavi': 1, 'shiksha': 2, 'vaishal': 3}
```

In [31]:
```
values=[1,1,1,2,3,4,5,6,6,7,8,9]#remove duplicates from list
print(values)
values_set=set(values)
unique_values=list(values_set)
print(unique_values)
```

```
[1, 1, 1, 2, 3, 4, 5, 6, 6, 7, 8, 9]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [39]:
```
celsius=37#convert celsius to fahrenheit
fahrenheit=(celsius*1.8)+32
print("%.2f celsius=%.2f fahrenheit"%(celsius,fahrenheit))
```

```
37.00 celsius=98.60 fahrenheit
```

In [42]:
```
year=int(input("enter a year:"))#check leap year
if (year%400==0) and (year%100==0):
    print(year,"is leap year")
elif (year%4==0) and (year%100!=0):
    print(year,"is leap year")
else:
    print(year,"is not leap year")
```

```
enter a year:2000
2000 is leap year
```

In [2]:
```
num=int(input("enter the number:"))#check prime number
if num==1:
    print(num,"is not prime number")
if num>1:
    for n in range(2,num):
        if num%2==0:
            print(num,"is not prime number")
            break
else:
    print(num,"is prime number")
```

```
enter the number:8
8 is not prime number
```

In [3]:
```
values=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25]
print(list(enumerate(values)))
```

```
[(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9), (9, 10), (10, 11), (11, 12), (12, 13),
(13, 14), (14, 15), (15, 16), (16, 17), (17, 18), (18, 19), (19, 20), (20, 21), (21, 22), (22, 23), (23, 24), (
24, 25)]
```

In [9]:
```
f=open("D:\\FileHandling.txt","a")
f.write("\nNow the file has more content!")
f.close()
f=open("D:\\FileHandling.txt","r")
```

```
print(f.read())f=open("D:\\FileHandling.txt","r")
f.close()
```

```
  Cell In[9], line 5
    print(f.read())f=open("D:\\FileHandling.txt","r")
                   ^
SyntaxError: invalid syntax
```

In [ ]: