In [1]:
```
n=eval(input())
```

```
python

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call las
t)
Cell In[1], line 1
----> 1 n=eval(input())

File <string>:1

NameError: name 'python' is not defined
```

In [ ]:
```
if <condition>
elif <condtion>
else # No condition
```

In [2]:
```
if a==10
print('hello')
```

```
  Cell In[2], line 2
    print('hello')
    ^
IndentationError: expected an indented block after 'if' statement on line
1
```

In [4]:
```
if a==10:
    print('hello')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call las
t)
Cell In[4], line 1
----> 1 if a==10:
      2     print('hello')

NameError: name 'a' is not defined
```

In [ ]:
```
summ=10
max=25
```

Functions

- Reuse any part of code

With out arguments

In [6]:
```
n1=10
n2=20
add=n1+n2
print("the addition of {} and {} is {}".format(n1,n2,add))
```

```
the addition of 10 and 20 is 30
```

In [ ]:
```
def <function_name>():
    # write your code here
```

In [20]:
```
def addition():
    n1=10
    n2=20
    add=n1+n2
    print("the addition of {} and {} is {}".format(n1,n2,add))
```

In [21]:
```
# The output will not display untill unless you call the function
```

In [22]:
```
addition()
```

```
the addition of 10 and 20 is 30
```

In [23]:
```
def hello():
    n1=eval(input('enter number'))
    print('hello')
    print('im writing function')
```

In [24]:
```
hello()
```

```
enter number50
hello
im writing function
```

In [ ]:
```

```

In [ ]:
```
but how did python execute 2nd and not 1st function.
```

In [17]:
```
a=100
a=200

# why a is not equal to 100
```

In [18]:
```
print(a)
```

```
200
```

In [19]:
```
a=1000
```

In [ ]:
```
# python will latest value one
```

In [ ]:
```
# In this particular cell : 500lines
```

In [ ]:
```
# lin1
```

In [ ]:
```
# lin2
```

In [ ]:
```
# lin3
```

In [25]:
```
n1=100
```

In [26]:
```
n2=200
```

In [27]:
```
print(n1+n2)
```

```
300
```

In [29]:
```
def hello_world():
    print('hello good morning')

hello_world()
```

```
hello good morning
```

In [ ]:
```
# Take three numbers find the average
# WAP ask the user enter 3 numbers
# n1=eval(input())
# n2=eval(input())
# n3=eval(input())
# avg=(n1+n2+n3)/3
# First write normal code
# Then implemnt the function
```

In [30]:
```
num1=eval(input("enter number1:"))
num2=eval(input("enter number2:"))
num3=eval(input("enter number3:"))
avg=(num1+num2+num3)/3
print("the average of {},{} and {} is {}".format(num1,num2,num3,avg))
```

```
enter number1:20
enter number2:30
enter number3:40
the average of 20,30 and 40 is 30.0
```

In [37]:
```
def avg():
    num1=eval(input("enter number1:"))
    num2=eval(input("enter number2:"))
    num3=eval(input("enter number3:"))
    avg=(num1+num2+num33333)/3
    print("the average of {},{} and {} is {}".format(num1,num2,num3,avg))
```

In [38]:
```
avg()
```

```
enter number1:100
enter number2:100
enter number3:100

---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Cell In[38], line 1
----> 1 avg()

Cell In[37], line 5, in avg()
      3 num2=eval(input("enter number2:"))
      4 num3=eval(input("enter number3:"))
----> 5 avg=(num1+num2+num33333)/3
      6 print("the average of {},{} and {} is {}".format(num1,num2,num3,av
g))

NameError: name 'num33333' is not defined
```

Note:

we are not sure , the function is defined correct or wrong

untill unless we call the function

In [39]:
```
try:
    num1=eval(input("enter number1:"))
    num2=eval(input("enter number2:"))
    num3=eval(input("enter number3:"))
    avg=(num1+num2+num3)/3
    print("the average of {},{} and {} is {}".format(num1,num2,num3,avg))

except Exception as e:
    print(e)
```

```
enter number1:20
enter number2:30
enter number3:40
the average of 20,30 and 40 is 30.0
```

In [41]:
```
def avg1():
    try:
        num1=eval(input("enter number1:"))
        num2=eval(input("enter number2:"))
        num3=eval(input("enter number3:"))
        avg=(num1+num2+num3333)/3
        print("the average of {},{} and {} is {}".format(num1,num2,num3,avg

    except Exception as e:
        print(e)

avg1()
```

```
enter number1:20
enter number2:30
enter number3:40
name 'num3333' is not defined
```

In [ ]:
```
# WAP ask the user enter bill amount
#     ask the user enter tip amount
#     calculate total bill

# First write normal code
# Then implement the function
```

In [42]:
```
def Bill():
    try:
        Bill_Amt= eval(input("Enter Bill amount"))
        Tip_Amt= eval(input("Enter Tip amount"))
        Total_Bill = Bill_Amt+Tip_Amt
        print("Total Bill is {}".format(Total_Bill))
    except exception as e:
        print(e)

Bill()
```

```
Enter Bill amount20
Enter Tip amount30
Total Bill is 50
```

In [ ]:
```
#WAP ask the user get a random number between 1 to 100
# print it is even or odd number
# Implement using function
# import <package>
# num=<package>.<method_name>(1,100)
# if <condition>:
#      print("{} is an even number".format(num))
#else:
#    print("{} is an odd number".format(num))
```

In [44]:
```
import random
num=random.randint(1,100)
if num%2==0:
    print("{} is an even number".format(num))
else:
    print("{} is an odd number".format(num))
```

```
54 is an even number
```

In [47]:
```
import random
def even_odd1():
    print('we are implementing even odd function')
    print("take one number")
    num=random.randint(1,100)
    if num%2==0:
        print("the remainder is zero")
        print("{} is an even number".format(num))
    else:
        print("the remainder is not equal to zero")
        print("{} is an odd number".format(num))

even_odd1()
```

```
wev are implementing even odd function
take one number
the remainder is zero
76 is an even number
```

In [48]:
```
import random
def even_odd2():
    try:
        print('we are implementing even odd function')
        print("take one number")
        num=random.randint(1,100)
        if num%2==0:
            print("the remainder is zero")
            print("{} is an even number".format(num))
        else:
            print("the remainder is not equal to zero")
            print("{} is an odd number".format(num))

    except Exception as e:
        print(e)

even_odd2()
```

```
we are implementing even odd function
take one number
the remainder is zero
8 is an even number
```

In [49]:

```
import random
print('hello')
print("python")
def even_odd2():
    try:
        print('we are implementing even odd function')
        print("take one number")
        num=random.randint(1,100)
        if num%2==0:
            print("the remainder is zero")
            print("{} is an even number".format(num))
        else:
            print("the remainder is not equal to zero")
            print("{} is an odd number".format(num))

    except Exception as e:
        print(e)

print('calling function')
even_odd2()
print("done!")
```

```
hello
python
calling function
we are implementing even odd function
take one number
the remainder is zero
50 is an even number
done!
```

In [ ]:
```
def addition():
    n1=20
    n2=30
    print('addition:',n1+n2)

print("mul function starts")
def mul():
    n1=20
    n2=30
    print('mul:',n1*n2)
    print("mul is done")

print("subtraction function starts")
def sub():
    n1=20
    n2=30
    print('sub:',n1-n2)
    print("subtraction is done")

print("anything remains")
print("no")
print("then call the function")

sub()
mul()
addition()
```

In [ ]:
```

```

In [ ]:
```
def addition():
    n1=10
    n2=20
    add=n1+n2
    print("the addition of {} and {} is {}".format(n1,n2,add))

addition()
```

In [ ]:
```
def Bill():
    try:
        Bill_Amt= eval(input("Enter Bill amount"))
        Tip_Amt= eval(input("Enter Tip amount"))
        Total_Bill = Bill_Amt+Tip_Amt
        print("Total Bill is {}".format(Total_Bill))
    except exception as e:
        print(e)

Bill()
```

In [ ]:
```
addition()
Bill()
```

- if you are not mentioning any values inside the function bracket
- with out arguments or with out parmeters
- arguments or parameters

With arguments

In [ ]:
```
def addition():
    n1=10
    n2=20
    add=n1+n2
    print("the addition of {} and {} is {}".format(n1,n2,add))

addition()

# 1Q) inside function how many variables are there
#      3 variables are there
# 2Q) how many user provided variables
#       2 variables
```

In [53]:
```
def addition(n1,n2): # arguments
    add=n1+n2
    print("the addition of {} and {} is {}".format(n1,n2,add))

addition(100,200)  # n1=100  n2=200
```

```
the addition of 100 and 200 is 300
```

In [55]:
```
def addition(n1,n2):
    add=n1+n2
    print("the addition of {} and {} is {}".format(n1,n2,add))

addition(200)
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[55], line 6
      3     add=n1+n2
      4     print("the addition of {} and {} is {}".format(n1,n2,add))
----> 6 addition(200)

TypeError: addition() missing 1 required positional argument: 'n2'
```

In [57]:
```
def addition11():
    add=n111+n222
    print("the addition of {} and {} is {}".format(n1,n2,add))

addition11(200,300)   # what is the error
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[57], line 5
      2     add=n111+n222
      3     print("the addition of {} and {} is {}".format(n1,n2,add))
----> 5 addition11(200,300)

TypeError: addition11() takes 0 positional arguments but 2 were given
```

In [58]:
```
# Implement below code using with arguments
def avg(num1,num2,num3):
    print("num1:",num1) # 20
    print("num2:",num2) #30
    print("num3:",num3) # 40
    avg=(num1+num2+num3)/3
    print("the average of {},{} and {} is {}".format(num1,num2,num3,avg))

avg(20,30,40)
```

```
num1: 20
num2: 30
num3: 40
the average of 20,30 and 40 is 30.0
```

In [67]:
```
num=12345
v1=num%10     #5
v11=num//100
v11  # 1234

v2=v11%10    # 4
v2
v22=v11//10 # 123
v22

# 12345========= 1234=====4
# 54321
```

Out[67]:
```
12
```

In [69]:
```
num=12345
v1=num%100
v1
```

Out[69]:
```
45
```

In [66]:
```
v11=num//10
v11
```

Out[66]:
```
1234
```

In [64]:
```
if a=10:
    print('h')
```

```
  Cell In[64], line 1
    if a=10:
       ^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

In [ ]:
```
7260927090
```

In [ ]:
```
12345 ======== 54
```

In [70]:
```
12345%10
```

Out[70]:
```
5
```

In [ ]:
```
12345 ======== 4?
```

In [ ]:
```
1234?
```

In [71]:
```
print(5,end='')
print(4)
```

```
54
```

In [ ]:
```
12345%10=====> 5
```

In [73]:
```
12345//10
```

Out[73]:
```
1234
```

In [78]:
```

```

Out[78]:
```
2
```

In [79]:
```
180
100
```

Out[79]:
```
100
```

In [ ]:
```
1100
150
310
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: