

```
In [1]: #Main Function
def cal(a,b):
    return (a+b)**2
#lambda parameter: expression
print((lambda a,b: (a+b)**2)(2,3))

print("Main Function", cal(2,3))
```

25  
Main Function 25

```
In [2]: def c(x):
        return x**3
a=c(5)
print(a)

#same code as above
b=(lambda x: x**3)(5)
print(b)
```

125  
125

```
In [3]: #MAP and Filter
def sqr(x):
    return x*x

num=[1,3,5,7,10]
r=list(map(sqr,num))
print(r)
```

[1, 9, 25, 49, 100]

```
In [4]: #Filter remove list
def sqr(x):
    return x%2==1

num=[1,3,5,7,10]

r=list(filter(sqr,num))
print(r)

r=list(map(sqr,num))
print(r)
```

[1, 3, 5, 7]  
[True, True, True, True, False]

```
In [5]: #Comprehension List
num=[1,3,5,7,10]
r=[x*x for x in num]
print(r)
r=[x for x in num if x%2==1]
print(r)
```

[1, 9, 25, 49, 100]  
[1, 3, 5, 7]

```
In [6]: #Zip Function,
roll=[1,2,3,4,5,6,7,8,9,11,12,13,14,15]
name=["A", "B","C","D","E","F","G","H","I","J","K","L","M","N"]

print(list(zip(roll,name,"abcdefghijklmnopg")))

[(1, 'A', 'a'), (2, 'B', 'b'), (3, 'C', 'c'), (4, 'D', 'd'), (5, 'E', 'e'),
(6, 'F', 'f'), (7, 'G', 'g'), (8, 'H', 'a'), (9, 'I', 'b'), (11, 'J', 'c'),
(12, 'K', 'd'), (13, 'L', 'e'), (14, 'M', 'f'), (15, 'N', 'g')]
```

```
In [7]: #Recursion Function
def fact(x):
    if x==1:
        return 1
    else:
        return x*fact(x-1)
print(fact(6))
```

720

```
In [8]: #Read File
file1=open("test.txt","r") #r for read, r+ for read&write
#print(file1.readable())
t=file1.read()
print(t)

fSize=len(t)
print(fSize)
```

```
file1.close()
```

Minhazul Kabir-SUST  
SAGAR-CU  
Minhazul Kabir-SUST  
Minhazul Kabir-SUST  
68

```
In [9]: f=open("test.txt","r+")

for l in f:
    print(l)

f.close()
```

Minhazul Kabir-SUST  
  
SAGAR-CU  
  
Minhazul Kabir-SUST  
  
Minhazul Kabir-SUST

```
In [10]: file=open("new.txt","a")

file.write("\n Adding New Line using 'a' ")

file.close()
```

```
In [11]: fi=open("newF.html","w")
fi.write("HI")
fi.close()
```

In [12]: *#Exception*

```
try:
    l=[10,0,27]
    r=l[0]/l[1]
    print(r)
    print("done")
except ZeroDivisionError:
    print("Zero")
except IndexError:
    print("index")
finally:
    print("Finally Must Kaj korbe. Error HoK ar na hoK")
print("Successful")
```

Zero

Finally Must Kaj korbe. Error HoK ar na hoK

Successful

In [13]: *#Exception*

```
try:
    a=int(input())
    b=int(input())
    r=a/b
    print(r)
    print("done")
except (ZeroDivisionError, IndexError, ValueError) :
    print("Incorrect input")
finally :
    print("Thanks")
```

10

0

Incorrect input

Thanks