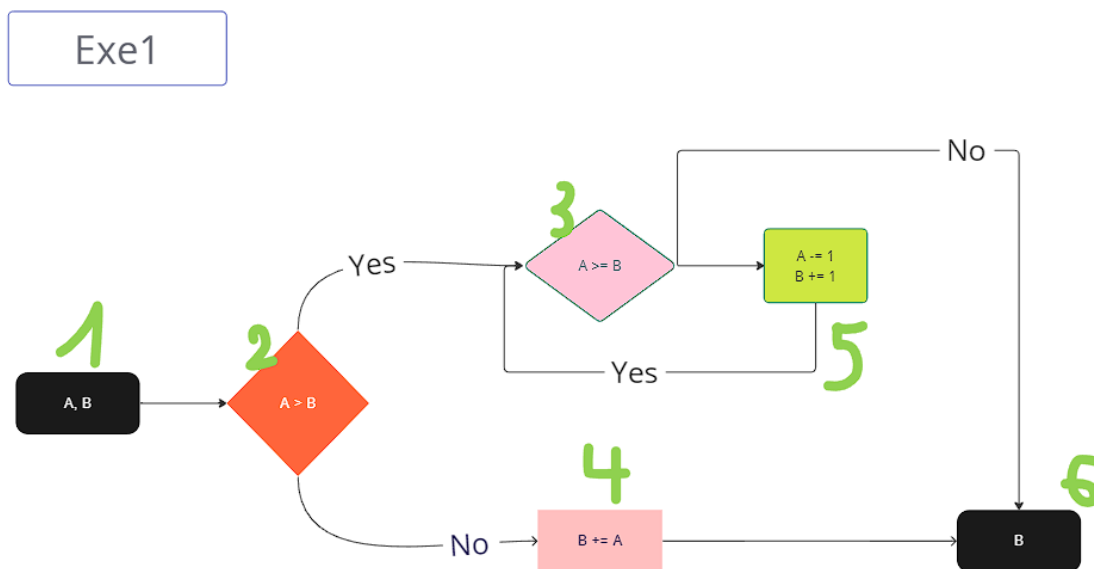


EXE1:

Convert code to Python

```
def func1(a, b):  
    if(a > b):  
        while(a >= b):  
            a -= 1  
            b += 1  
    else:  
        b += a  
    return b
```

Control Flow:



- TC1: (3,2)

```
a = 3  
b = 2  
3|
```

- Statement coverage: line 1,2,3,4,5,8 -> 6/8 = 75%
- Decision coverage: 1/2 = 50%
- TC2: (3,4)

```
a = 3
b = 4
7
```

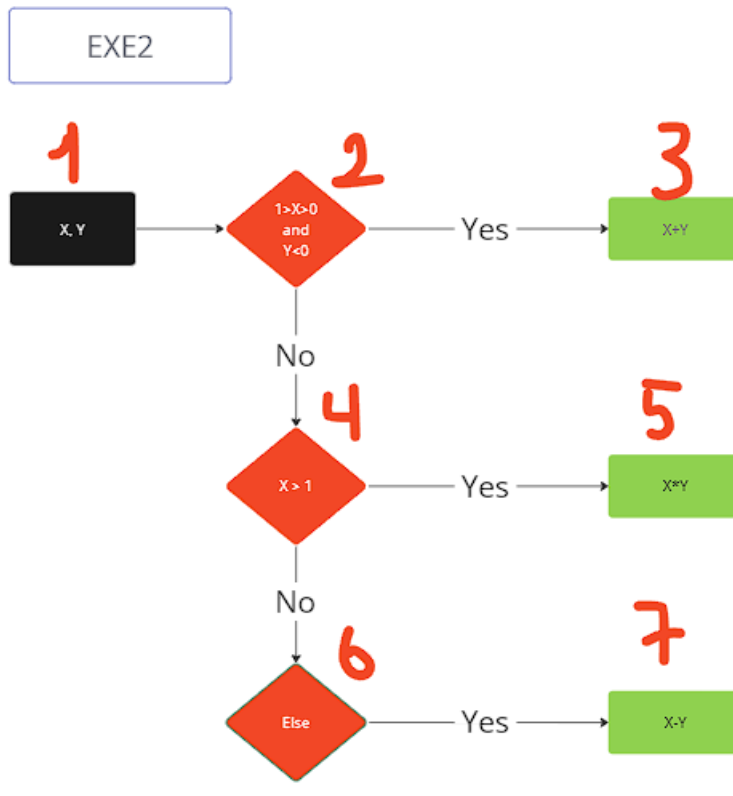
- Statement coverage: line 1,2,6,7,8 ->  $5/8 = 62,5\%$
- Decision coverage:  $1/2 = 50\%$

To get 100% decision coverage, I choose TC1(3,2) and TC2(3,4)

EXE 2

```
def func2(x,y):
    if((1 > x > 0) and (y < 0)):
        return x + y
    elif(x > 1):
        return x*y
    else:
        return x-y
```

Control Flow:



Test case:

- TC1: (0.5,-4)

```

Enter x: 0.5
Enter y: -4
-3.5|
  
```

- Statement coverage: line 1,2,3 ->  $3/7 = 42,9\%$
- Decision coverage:  $1/3 = 33.3\%$

- TC2: (5,10)

```

Enter x: 5
Enter y: 10
50.0
  
```

- Statement coverage: line 1,2,4,5 ->  $4/7 = 57,1\%$
- Decision coverage:  $1/3 = 33.3\%$

To get 100% decision coverage, I choose TC1(0.5,-4); TC2(5,10); TC3(-4,1)

EXE 3

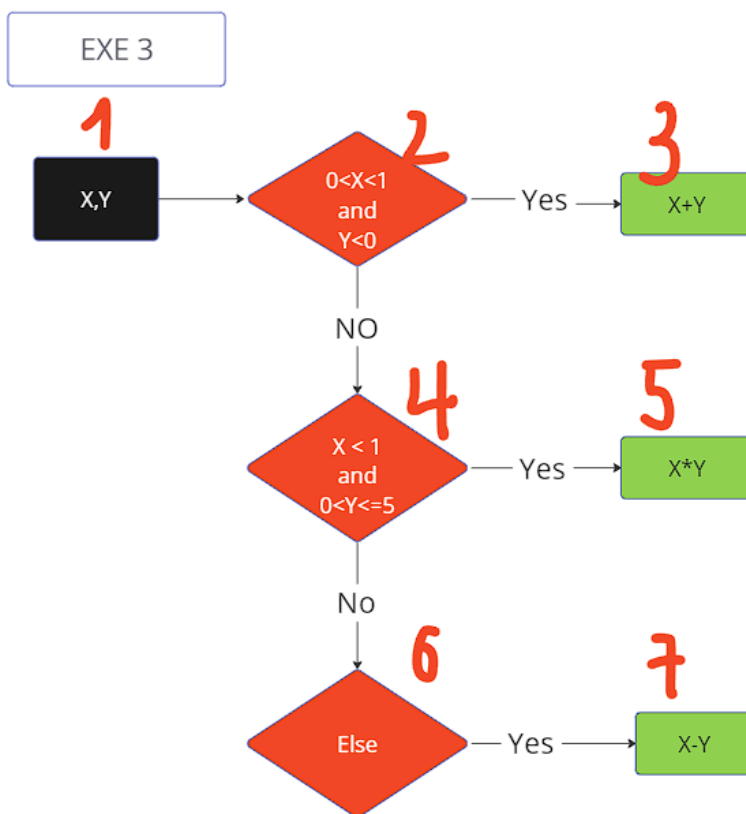
```
def funct(x,y):
```

```

if((0 < x <= 1) and (y < 0)):
    return x + y
elif((x > 1) and (0 < y <= 5)):
    return x * y
else:
    return x - y

```

Control flow:



Test case:

- TC1(0.5, -1):

```

Enter x: 0.5
Enter y: -1
-0.5

```

- Statement coverage: line 1,2,3 -> 3/7 = 42,9%
- Decision coverage: 1/3 = 33.3%

- TC2(6,3):

```
Enter x: 6
Enter y: 3
18.0
```

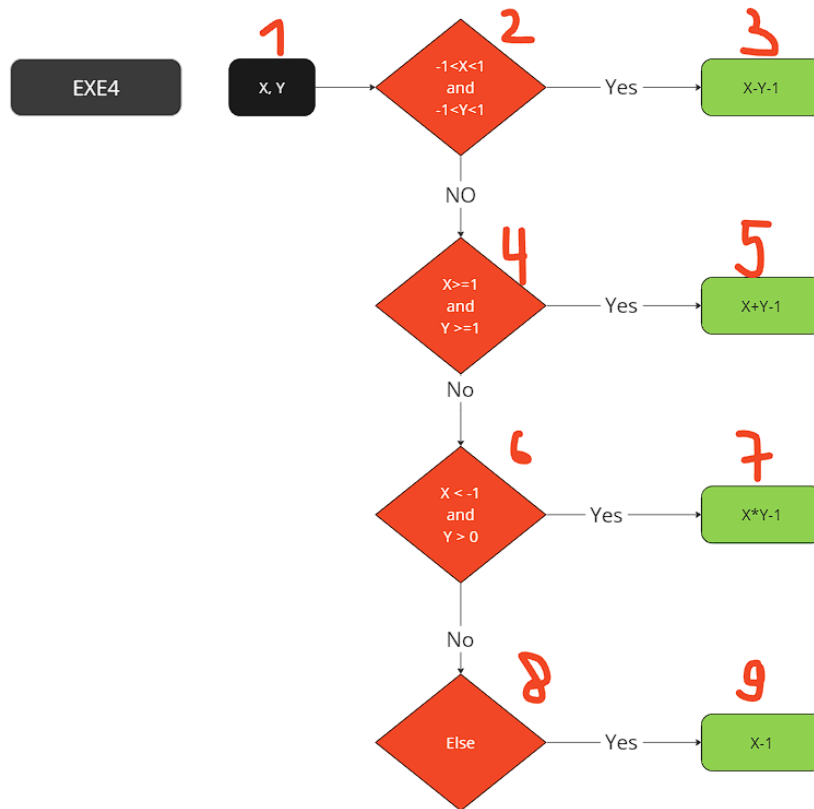
- Statement coverage: line 1,2,4,5 ->  $4/7 = 57,1\%$
- Decision coverage:  $1/3 = 33.3\%$

To get 100% decision coverage, I choose TC1(0.5,-1); TC2(6,3); TC3(-4,1)

EXE 4:

```
def funct(x,y):
    if((-1<x<1) and (-1<y<1)):
        return x-y-1
    elif((x>=1) and (y>=1)):
        return x+y-1
    elif((x < -1) and (y>0)):
        return x*y-1
    else:
        return x-1
```

Control flow



Test case:

- TC1(0,0):

```

Enter x: 0
Enter y: 0
-1.0
  
```

- Statement coverage: line 1,2,3 ->  $3/9 = 33.3\%$
- Decision coverage:  $1/4 = 25\%$

- TC2(5,6):

```

Enter x: 5
Enter y: 6
10.0
  
```

- Statement coverage: line 1,2,4,5 ->  $4/9 = 44.4\%$
- Decision coverage:  $1/4 = 25\%$

To get 100% decision coverage, I choose TC1(0,0); TC2(5,6); TC3(-4,4); TC4(-5,-1)

EXE 5

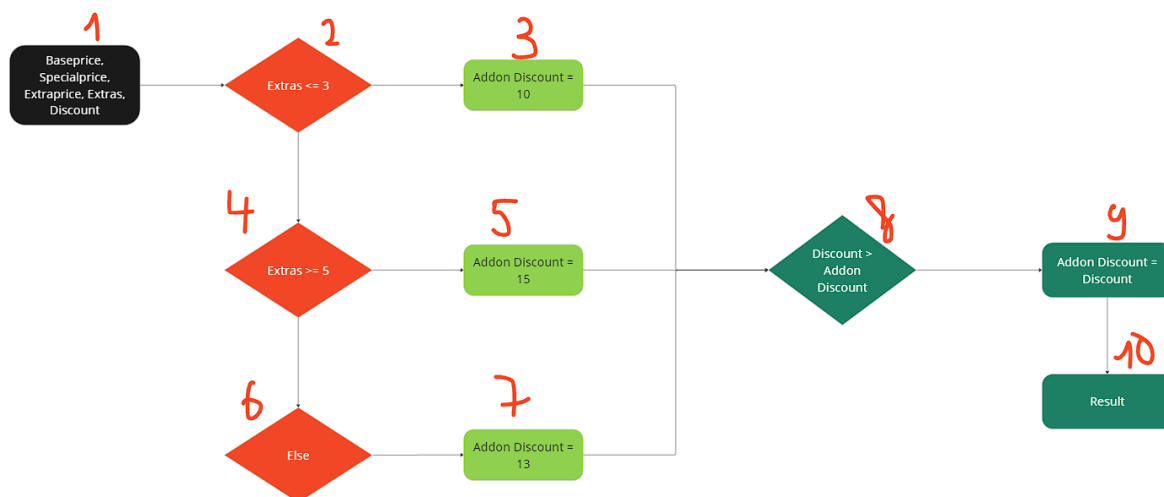
```

def calculate_price_Tdque(baseprice, specialprice,
extraprice, extras, discount):
    addon_discount = 0
    result = 0
    if extras <= 3:
        addon_discount = 10
    elif extras >= 5:
        addon_discount = 15
    else:
        addon_discount = 13

    if discount > addon_discount:
        addon_discount = discount
    result = baseprice/100.0*(100-discount) +
specialprice + extraprice/100.0*(100-addon_discount)
    return result

```

Control flow:



Test case:

- TC1(50,50,100,4,15):

```
Enter baseprice: 50
Enter specialprice: 50
Enter extraprice: 100
Enter extras: 4
Enter discount: 15
177.5
```

- Statement coverage: line 1,2,3,4,5,8,9,10,11 -> 9/11 = 81.82%
- Decision coverage: 1/6 = 16.67%
- TC2(100,90,80,10,10):

```
Enter baseprice: 100
Enter specialprice: 90
Enter extraprice: 80
Enter extras: 10
Enter discount: 10
248.0
```

- Statement coverage: line 1,2,3,4,6,7,8,10,11 -> 9/11 = 81.82%
- Decision coverage: 1/6 = 16.67%

To get 100% decision coverage, I choose TC1(50,50,100,4,15); TC2(100,90,80,10,10); TC3(50,50,100,4,8), TC4(100,90,80,10,20); TC5(100,90,80,1,20); TC6(100,90,80,2,0)

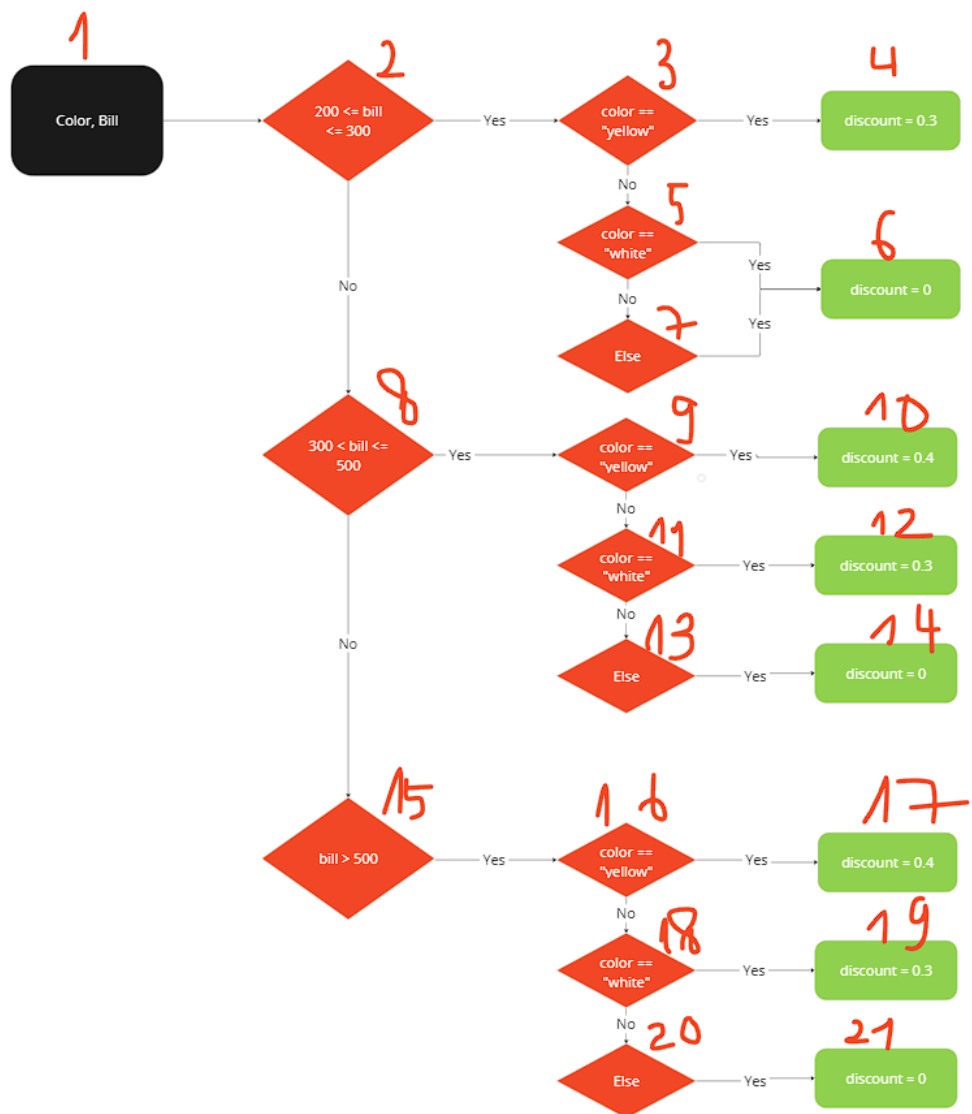
EXE 6:

```
def calculate_price(color, bill):
    discount = 0
    if(200 <= bill <= 300):
        if(color == "yellow"):
            discount = 0.3
        elif(color == "white"):
            discount = 0
        else:
            discount = 0
    elif(300 < bill <= 500):
        if(color == "yellow"):
            discount = 0.4
        elif(color == "white"):
```



```
        discount = 0.3
    else:
        discount = 0
elif(bill > 500):
    if(color == "yellow"):
        discount = 0.6
    elif(color == "white"):
        discount = 0.5
    else:
        discount = 0
else:
    discount = 0
return discount
```

Control Flow



Test:

- TC1(250,yellow):

```

Enter bill: 250
Enter color: yellow
0.3
  
```

- Statement coverage: line 1,2,3,4,5,26 -> 6/26 = 23.07%
- Decision coverage: 1/10 = 10%

- TC2(400,white):

```

Enter bill: 400
Enter color: white
0.3
  
```

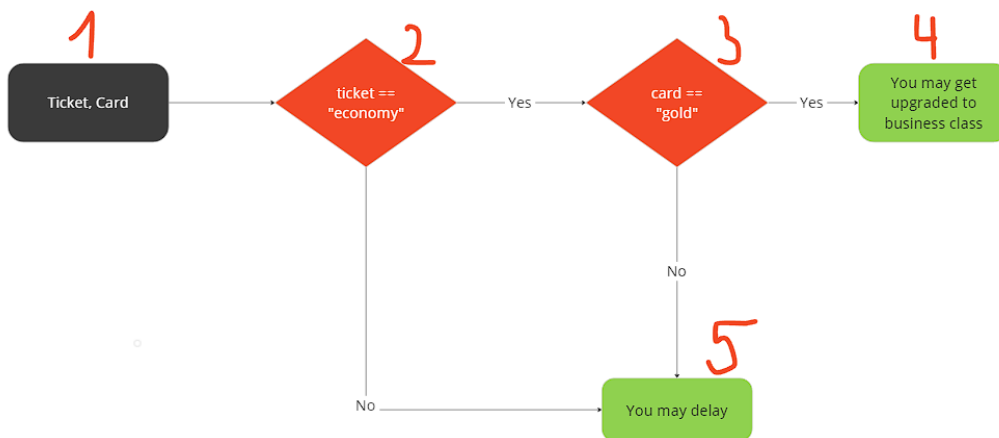
- Statement coverage: line 1,2,3,10,11,13,14,26 -> 8/26 = 30.76%
- Decision coverage: 1/10 = 10%

To get 100% decision coverage, I choose TC1(250,yellow); TC2(400,white); TC3(250,white); TC4(250,black); TC5(400,yellow); TC6(400,black); TC7(600,yellow); TC8(600,white); TC9(600,red); TC10(100,red).

EXE 7:

```
def identify_class(ticket, card):
    notify = ""
    if(ticket == "economy"):
        if(card == "gold"):
            notify = "You may get upgraded to business class"
        else:
            notify = "You may delay"
    else:
        notify = "You may delay"
    return notify
```

Control Flow



Test:

- TC1(economy,gold):

```
Enter your ticket: economy
Enter your card: gold
You may get upgraded to business class
```

- Statement coverage: line 1,2,3,4,5,10 ->  $6/10 = 60\%$
- Decision coverage:  $1/3 = 33.3\%$
- TC2(economy,sliver):

```
Enter your ticket: economy
Enter your card: sliver
You may delay
```

- Statement coverage: line 1,2,3,4,6,7,10 ->  $7/10 = 70\%$
- Decision coverage:  $1/3 = 33.3\%$

To get 100% decision coverage, I choose TC1(economy,gold); TC2(economy,sliver); TC3(cheap economy, sliver)