

PYTHON CONDITIONAL STATEMENTS AND FUNCTION

1."Calculate the discount applicable for "Pet shop" customers

a) If the customer is a Premium member then 20% discount is applicable on total bill value.

b) If the customer is a Gold member then 15% discount is applicable on total bill value.

c) If the customer is a Silver member then 10% discount is applicable on total bill value.

d) For all other customers the discount will be 5% of their total bill valued over 2000.

Code

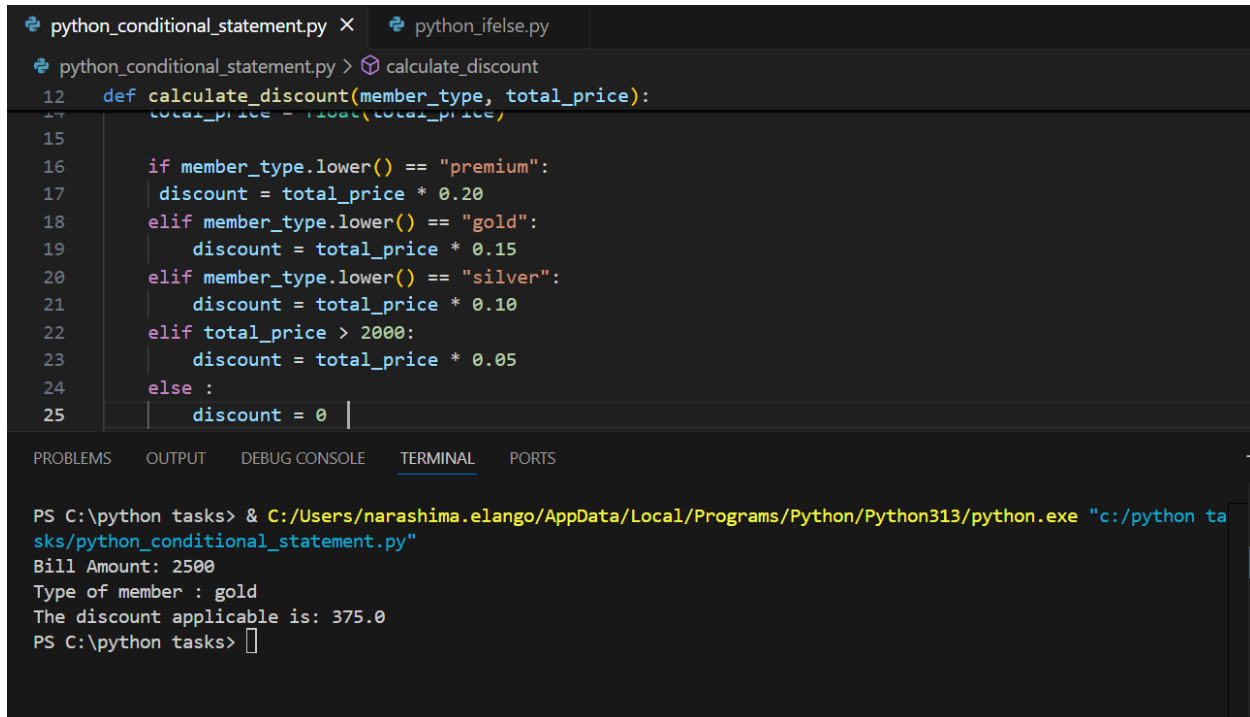
```
price = input("Bill Amount: ")
member = input("Type of member : ")

def calculate_discount(member_type, total_price):
    total_price = float(total_price)
    if member_type.lower() == "premium":
        discount = total_price * 0.20
    elif member_type.lower() == "gold":
        discount = total_price * 0.15
    elif member_type.lower() == "silver":
        discount = total_price * 0.10
    elif total_price > 2000:
        discount = total_price * 0.05
    else :
        discount = 0
    return discount
```

```
discount = calculate_discount(member, price)
```

```
print(f"The discount applicable is: {discount}")
```

Output:



The screenshot shows a code editor with two tabs: 'python_conditional_statement.py' (active) and 'python_ifelse.py'. The active tab contains a Python function 'calculate_discount' that takes 'member_type' and 'total_price' as arguments. The function uses an if-elif-else structure to calculate a discount based on the member type and total price. The terminal output shows the execution of the script, displaying the bill amount, member type, and the calculated discount.

```
python_conditional_statement.py X python_ifelse.py
python_conditional_statement.py > calculate_discount
12 def calculate_discount(member_type, total_price):
13     total_price = float(total_price)
14
15
16     if member_type.lower() == "premium":
17         discount = total_price * 0.20
18     elif member_type.lower() == "gold":
19         discount = total_price * 0.15
20     elif member_type.lower() == "silver":
21         discount = total_price * 0.10
22     elif total_price > 2000:
23         discount = total_price * 0.05
24     else :
25         discount = 0

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\python tasks> & C:/Users/narashima.elango/AppData/Local/Programs/Python/Python313/python.exe "c:/python ta
sks/python_conditional_statement.py"
Bill Amount: 2500
Type of member : gold
The discount applicable is: 375.0
PS C:\python tasks> 
```

2. An employee is considered for on-site depending on these conditions

- (i) An employee Should have Passport
- (ii) Communication should be good
- (iii) His training feedback should be good
- (iv) Should be at-least 2years experienced.
- (v) Age should be greater than or equal to 23.

Using above conditions, check if an employee is eligible to go to on-site or not.

Code

```
def if_eligible_for_onsite(passport, communication, feedback, experience, age):

    if passport == "yes" and communication == "good" and feedback == "good" and
    experience >= 2 and age >= 23 :

        return ("Eligible for Onsite")

    else :
```

```
return ("Not Eligible for Onsite")
```

```
passport = input("Passport")
```

```
communication = input("communication")
```

```
feedback = input("Feedback")
```

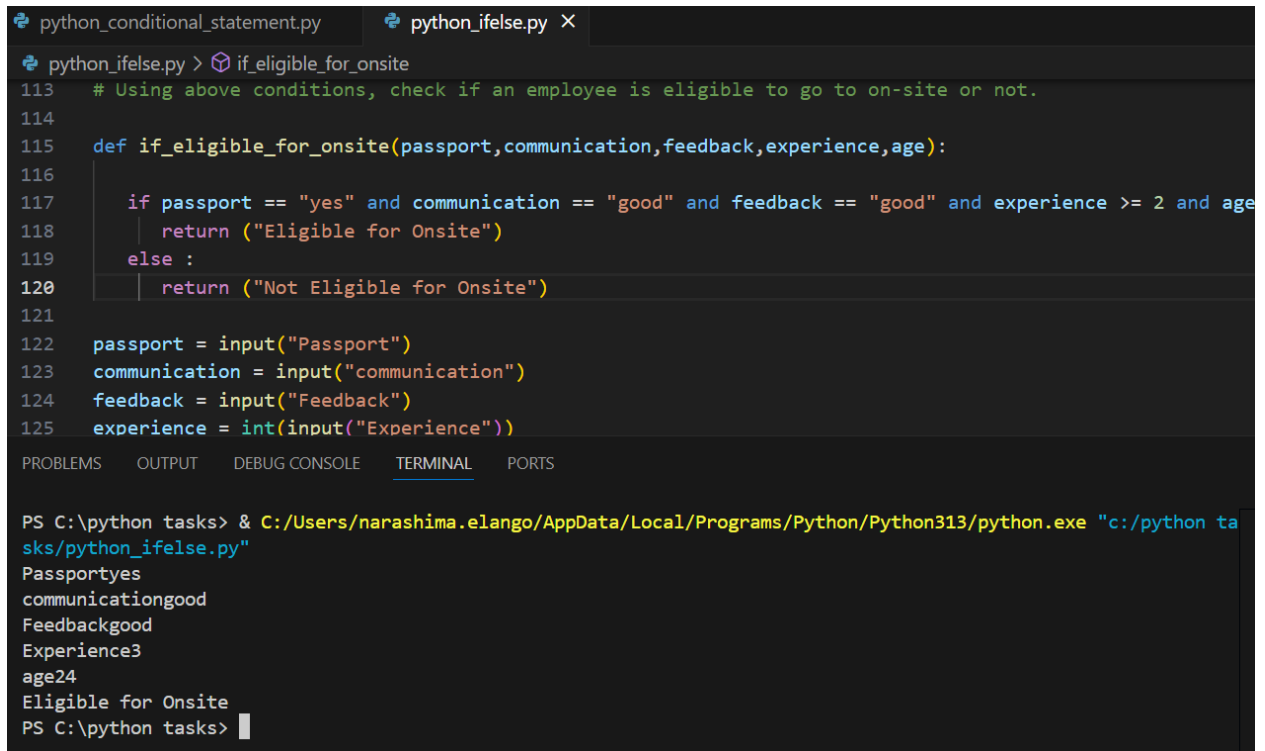
```
experience = int(input("Experience"))
```

```
age = int(input("age"))
```

```
Eligible_check = if_eligible_for_onsite(passport, communication, feedback, experience, age)
```

```
print(Eligible_check)
```

Output:



The screenshot shows a code editor with two tabs: 'python_conditional_statement.py' and 'python_ifelse.py'. The 'python_ifelse.py' tab is active, displaying the following code:

```
python_ifelse.py > if_eligible_for_onsite
113 # Using above conditions, check if an employee is eligible to go to on-site or not.
114
115 def if_eligible_for_onsite(passport, communication, feedback, experience, age):
116
117     if passport == "yes" and communication == "good" and feedback == "good" and experience >= 2 and age
118         return ("Eligible for Onsite")
119     else :
120         return ("Not Eligible for Onsite")
121
122 passport = input("Passport")
123 communication = input("communication")
124 feedback = input("Feedback")
125 experience = int(input("Experience"))
```

Below the code editor, the 'TERMINAL' tab is active, showing the execution of the script:

```
PS C:\python tasks> & C:/Users/narashima.elango/AppData/Local/Programs/Python/Python313/python.exe "c:/python ta
sks/python_ifelse.py"
Passportyes
communicationgood
Feedbackgood
Experience3
age24
Eligible for Onsite
PS C:\python tasks>
```

```
python_conditional_statement.py python_ifelse.py X
python_ifelse.py > if_eligible_for_onsite
113 # Using above conditions, check if an employee is eligible to go to on-site or not.
114
115 def if_eligible_for_onsite(passport,communication,feedback,experience,age):
116
117     if passport == "yes" and communication == "good" and feedback == "good" and experience >= 2 and ag
118         return ("Eligible for Onsite")
119     else :
120         return ("Not Eligible for Onsite")
121
122 passport = input("Passport")
123 communication = input("communication")
124 feedback = input("Feedback")
125 experience = int(input("Experience"))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\python tasks> & C:/Users/narashima.elango/AppData/Local/Programs/Python/Python313/python.exe "c:/python ta
sks/python_ifelse.py"
Passportyes
communicationgood
Feedbacknotgood
Experience3
age23
Not Eligible for Onsite
PS C:\python tasks> 
```

3. Calculate electricity bill for following constraints.

- (i) If units exceeds 1000, then charge Rs.10/- per unit.
- (ii) If units exceeds 500, then charge Rs.5/- per unit.
- (iii) If units exceeds 200, then charge Rs.2/- per unit.
- (iv) In other cases charge Rs.1/- per unit.

Code

```
def total_bill(unit):
    if unit > 500 and unit >= 1000 :
        calculated_bill = 10 * unit
    elif unit <= 500 :
        calculated_bill = 5 * unit
    elif unit > 200 and unit < 500:
```

```
    calculated_bill = 2 * unit
```

```
else:
```

```
    calculated_bill=unit
```

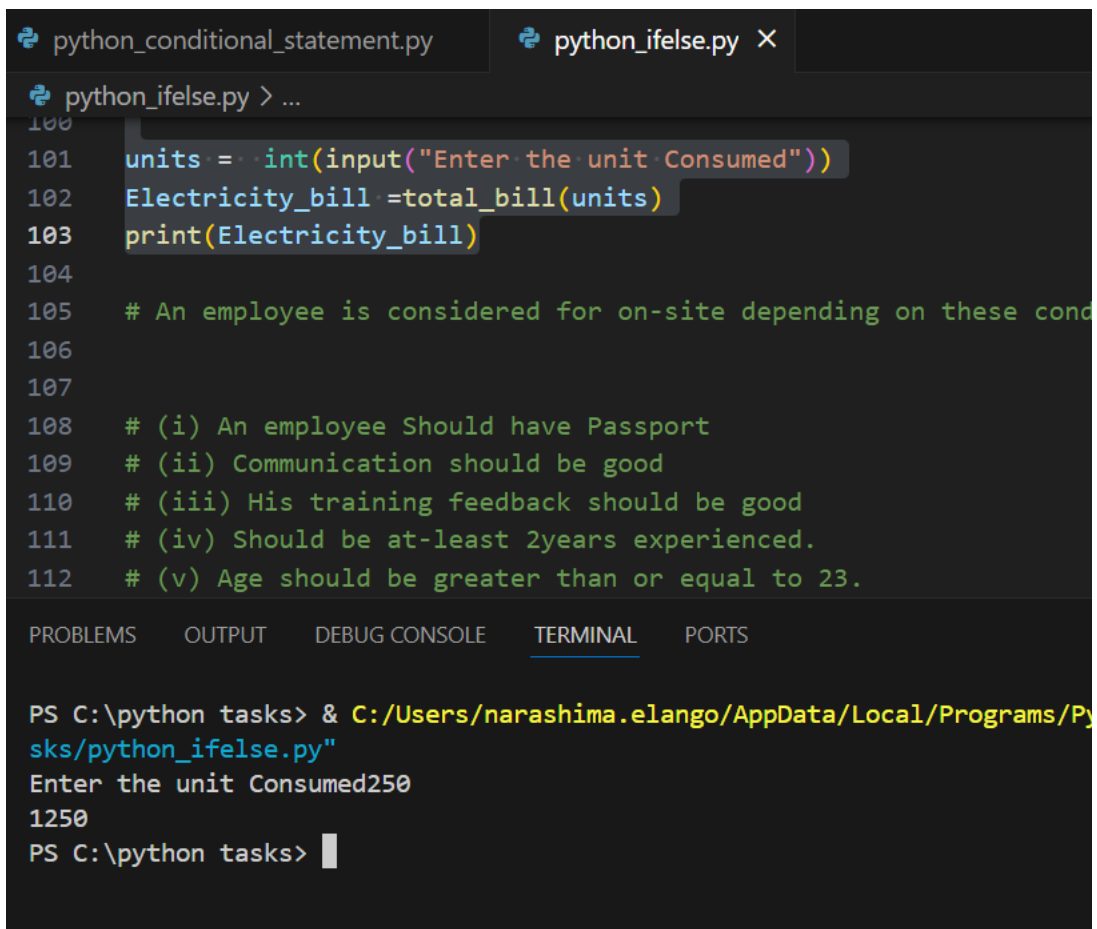
```
return calculated_bill
```

```
units = int(input("Enter the unit Consumed"))
```

```
Electricity_bill =total_bill(units)
```

```
print(Electricity_bill)
```

Output:



The screenshot shows a code editor with two tabs: 'python_conditional_statement.py' and 'python_ifelse.py'. The 'python_ifelse.py' tab is active, showing a Python script with line numbers 100 to 112. The script contains the following code:

```
100
101 units = int(input("Enter the unit Consumed"))
102 Electricity_bill =total_bill(units)
103 print(Electricity_bill)
104
105 # An employee is considered for on-site depending on these cond
106
107
108 # (i) An employee Should have Passport
109 # (ii) Communication should be good
110 # (iii) His training feedback should be good
111 # (iv) Should be at-least 2years experienced.
112 # (v) Age should be greater than or equal to 23.
```

Below the code editor, there is a terminal window with tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the command prompt output:

```
PS C:\python tasks> & C:/Users/narashima.elango/AppData/Local/Programs/Py
sks/python_ifelse.py"
Enter the unit Consumed250
1250
PS C:\python tasks> |
```

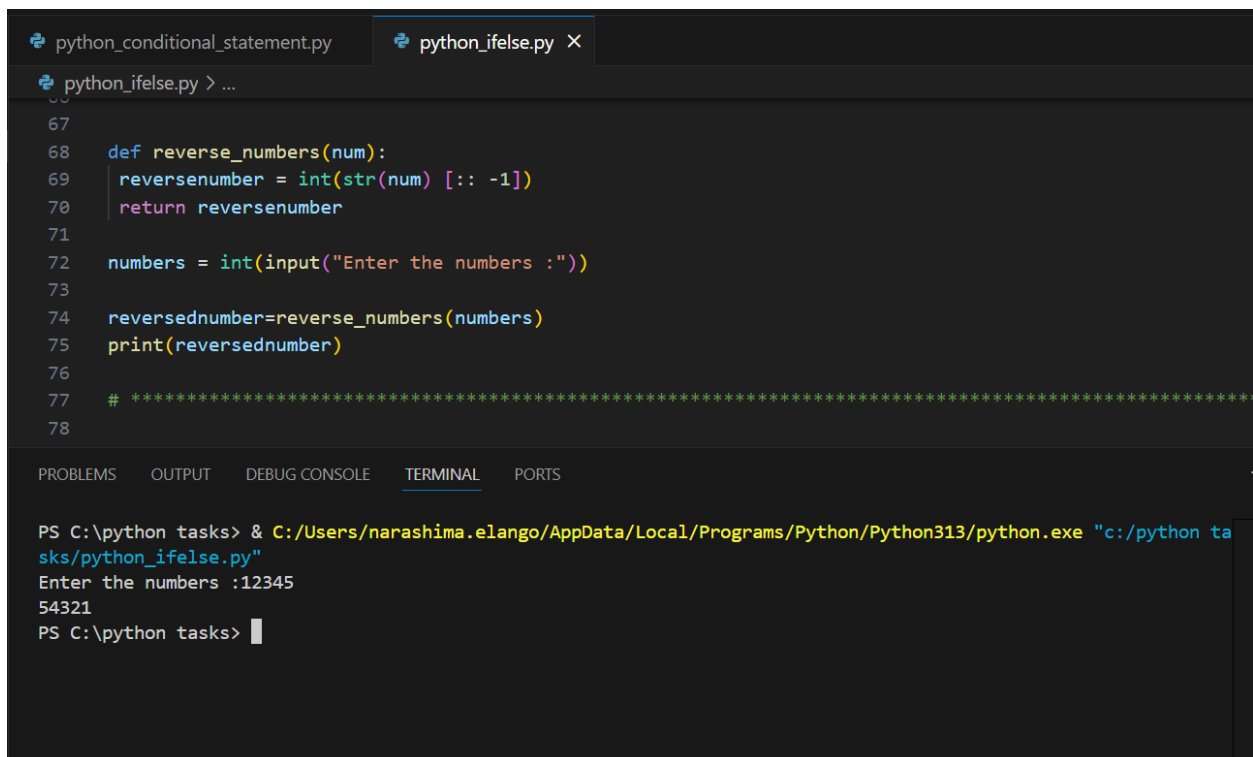
4. Accept a 5 digit decimal number as input and display the number in reverse order.

(Example: if input is 12345, then output must be 54321)

Code

```
def reverse_numbers(num):  
  
    reversenumber = int(str(num) [::-1])  
  
    return reversenumber  
  
numbers = int(input("Enter the numbers :"))  
  
reversednumber=reverse_numbers(numbers)  
  
print(reversednumber)
```

Output:



The screenshot shows a code editor with two tabs: 'python_conditional_statement.py' and 'python_ifelse.py'. The 'python_ifelse.py' tab is active, displaying the following code:

```
67  
68 def reverse_numbers(num):  
69     reversenumber = int(str(num) [::-1])  
70     return reversenumber  
71  
72 numbers = int(input("Enter the numbers :"))  
73  
74 reversednumber=reverse_numbers(numbers)  
75 print(reversednumber)  
76  
77 # *****  
78
```

Below the code editor is a terminal window with the following output:

```
PS C:\python tasks> & C:/Users/narashima.elango/AppData/Local/Programs/Python/Python313/python.exe "c:/python ta  
sks/python_ifelse.py"  
Enter the numbers :12345  
54321  
PS C:\python tasks>
```

5.Display all the prime numbers between x and y

Code

```
import math

def get_prime_numbers(num):

    if num <= 1 :

        return False

    for i in range(2, int(math.sqrt(num)) + 1):

        if num % i == 0 :

            return False

    return True

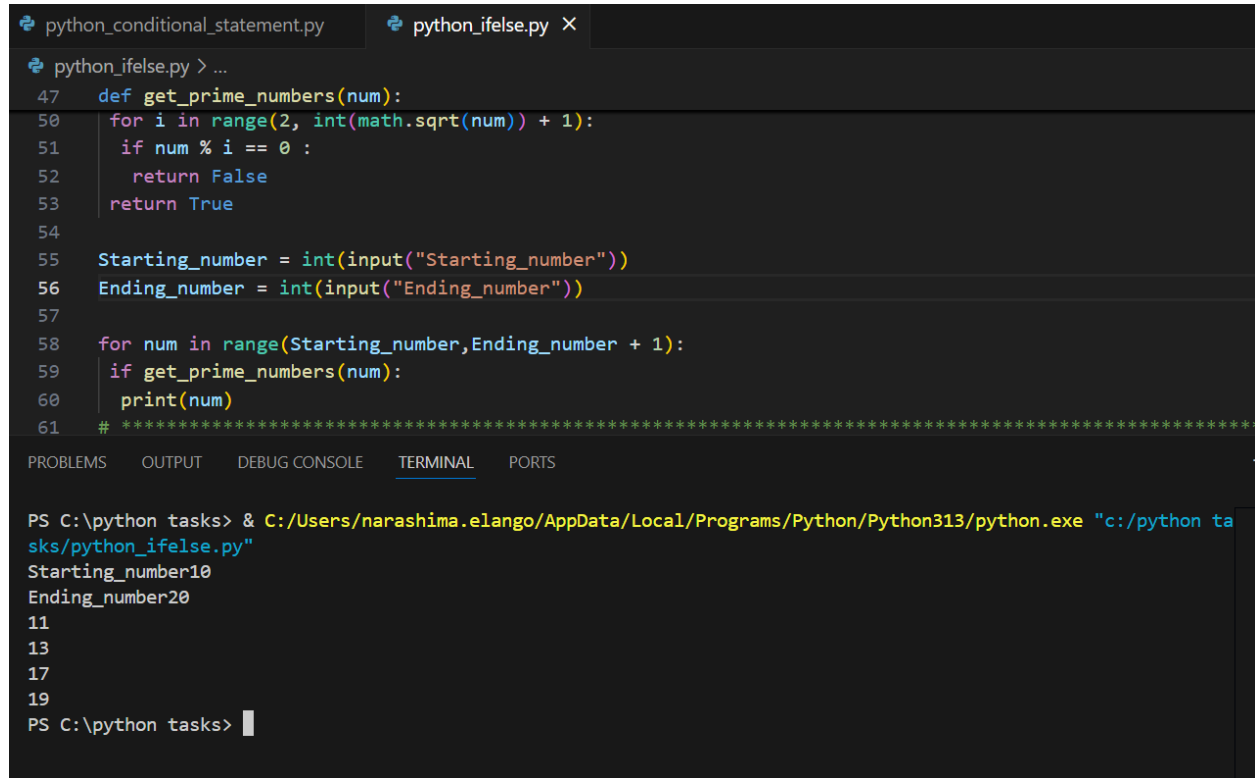

Starting_number = int(input("Starting_number"))
Ending_number = int(input("Ending_number"))


for num in range(Starting_number,Ending_number + 1):

    if get_prime_numbers(num):

        print(num)
```

Output:



```
python_conditional_statement.py  python_ifelse.py X
python_ifelse.py > ...
47 def get_prime_numbers(num):
48     for i in range(2, int(math.sqrt(num)) + 1):
49         if num % i == 0 :
50             return False
51         return True
52
53 Starting_number = int(input("Starting_number"))
54 Ending_number = int(input("Ending_number"))
55
56 for num in range(Starting_number,Ending_number + 1):
57     if get_prime_numbers(num):
58         print(num)
59
60 # *****
61
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\python tasks> & C:/Users/narashima.elango/AppData/Local/Programs/Python/Python313/python.exe "c:/python tasks/python_ifelse.py"
Starting_number10
Ending_number20
11
13
17
19
PS C:\python tasks> |
```

6.Implement the following logic to select the mode of transport for dispatching Pet animals from our

"Pet shop" (DO NOT use logical operators)

- a) If priority is not urgent and the weight is less than or equal to 5 Kg, dispatch by Bike.
- b) If priority is not urgent and the weight is more than 5 Kg, select a lorry if the distance is less Than or equal to 250 Km.
- c) If the priority is urgent and distance is less than 50 Km and weight is less than 100 Kg, Select a van
- d) In all other cases, use a train

Code

```
Type = input("piority: ")
```

```
Weight = int(input("weight: "))
```

```
Distance = int(input("Distance: "))
```

```
def transport_dispatch (type,weight,distance):
```

```
    if type == "not urgent":
```

```
        if weight <= 5:
```

```
            return "bike"
```

```
    if type == "not urgent":
```

```
        if weight > 5 :
```

```
            if distance <= 250:
```

```
                return "lorry"
```

```
    if type == "urgent":
```

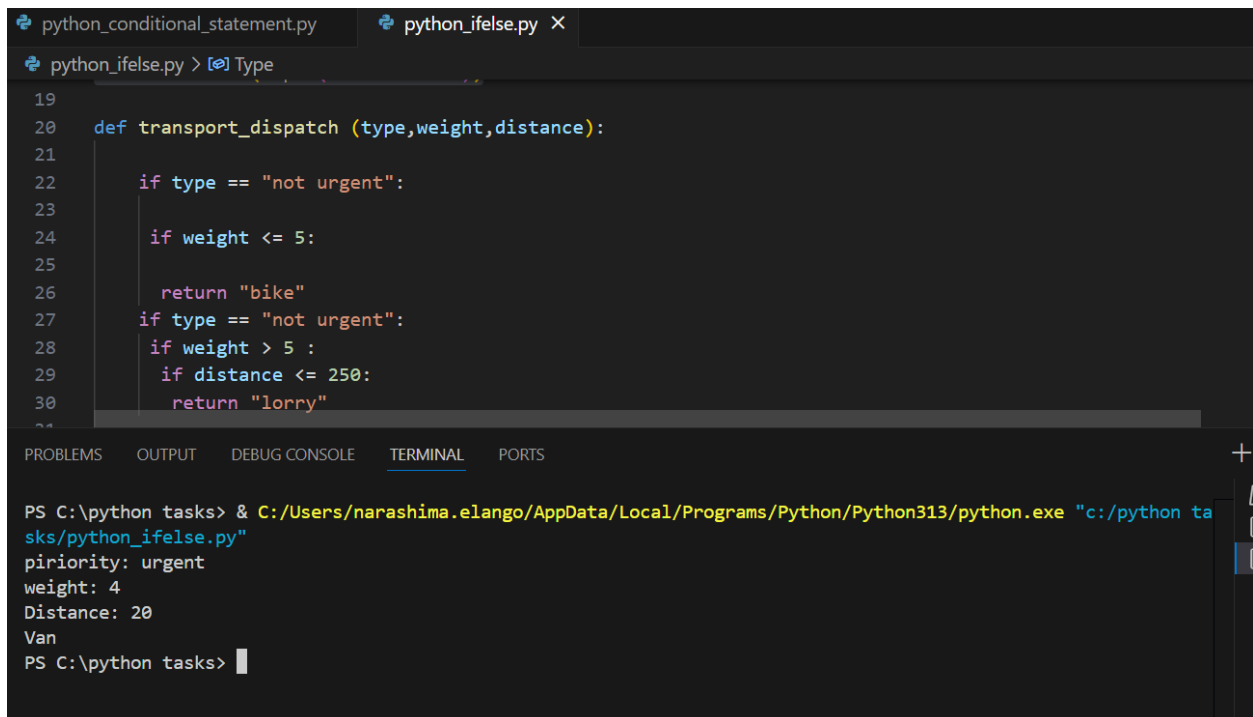
```
        if weight <= 100 :
```

```
            if distance <= 50:
```

```
                return "Van"
```

```
    return "Train"
```

Output:



The image shows a code editor with two tabs: `python_conditional_statement.py` and `python_ifelse.py`. The `python_ifelse.py` tab is active, displaying a Python function `transport_dispatch` with nested if-else statements. Below the code editor is a terminal window with tabs for `PROBLEMS`, `OUTPUT`, `DEBUG CONSOLE`, `TERMINAL`, and `PORTS`. The `TERMINAL` tab is selected, showing the command to run the script and its output.

```
python_conditional_statement.py python_ifelse.py X
python_ifelse.py > [?] Type

19
20 def transport_dispatch (type,weight,distance):
21
22     if type == "not urgent":
23
24         if weight <= 5:
25
26             return "bike"
27     if type == "not urgent":
28         if weight > 5 :
29             if distance <= 250:
30                 return "lorry"
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\python tasks> & C:/Users/narashima.elango/AppData/Local/Programs/Python/Python313/python.exe "c:/python ta
sks/python_ifelse.py"
priority: urgent
weight: 4
Distance: 20
Van
PS C:\python tasks> 
```