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.

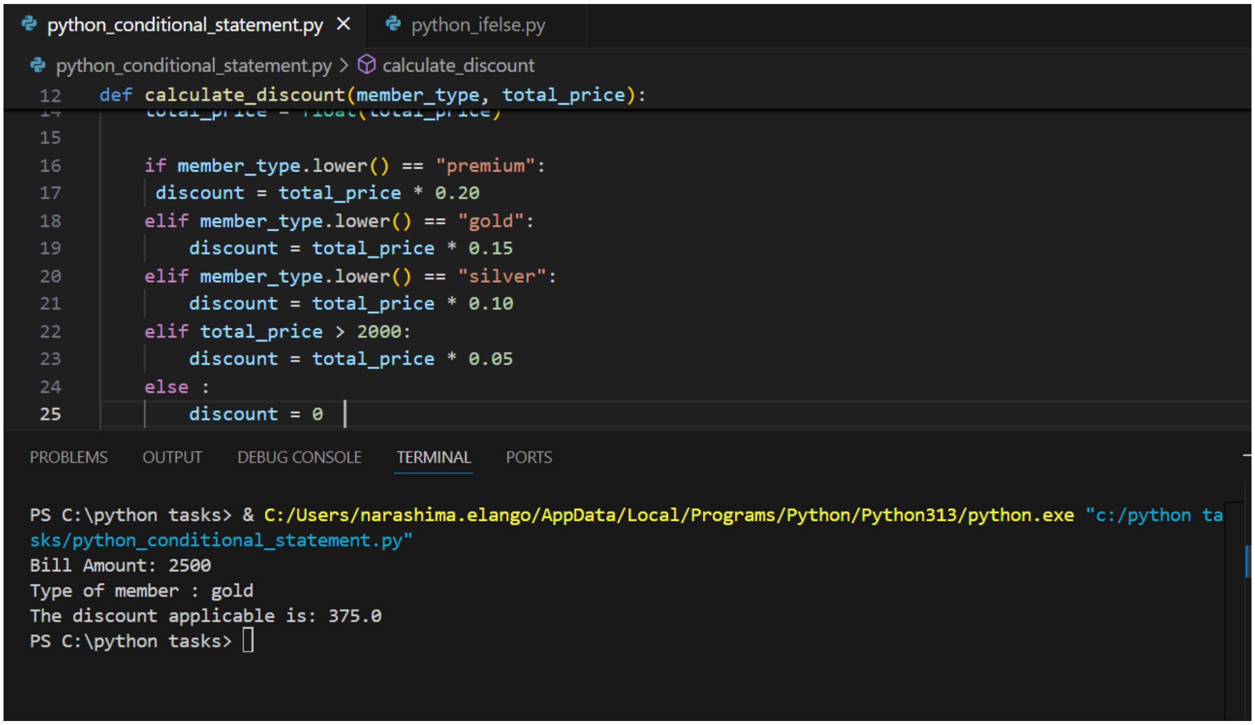
c) If the customer is a Silver member then 10% discount is applicable on total bill value.

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:


python_conditional_statement.py
python_ifelse.py
python_conditional_statement.py > calculate_discount
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```python
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
```

```
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.

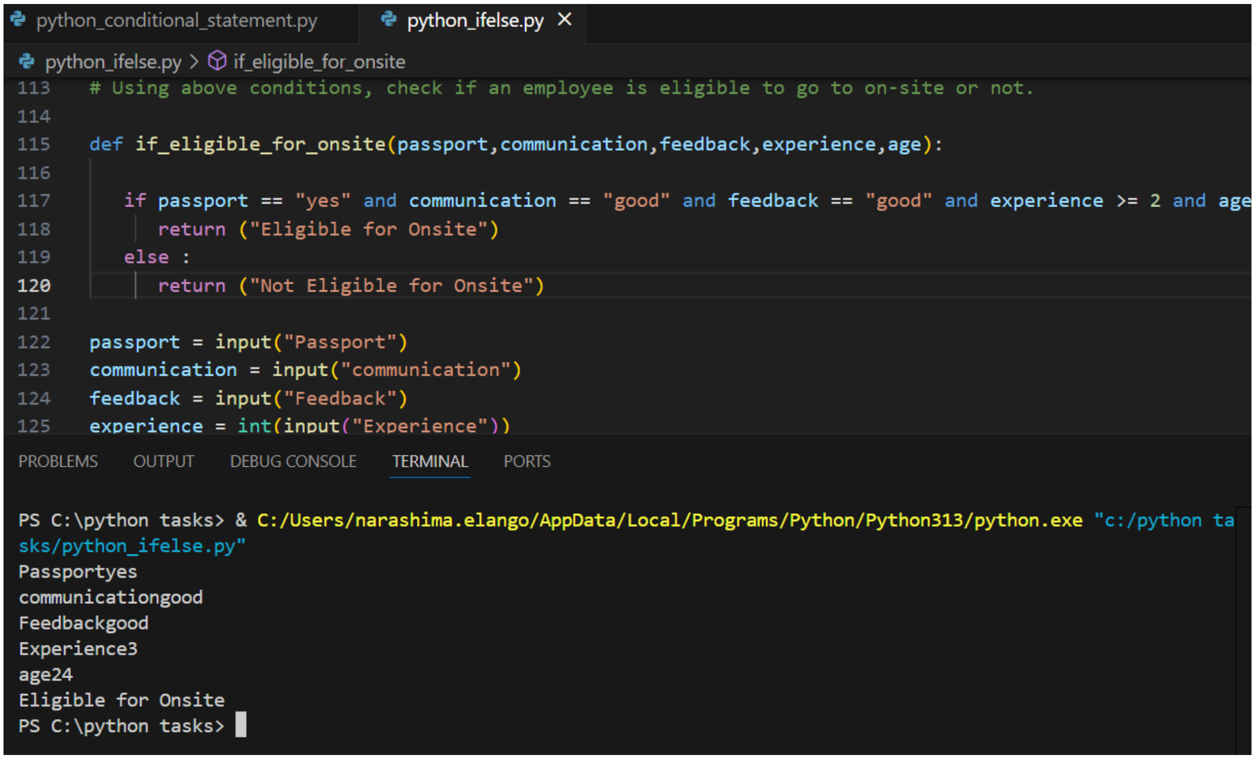
(v) Age should be greater than or equal to 23.

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

Code

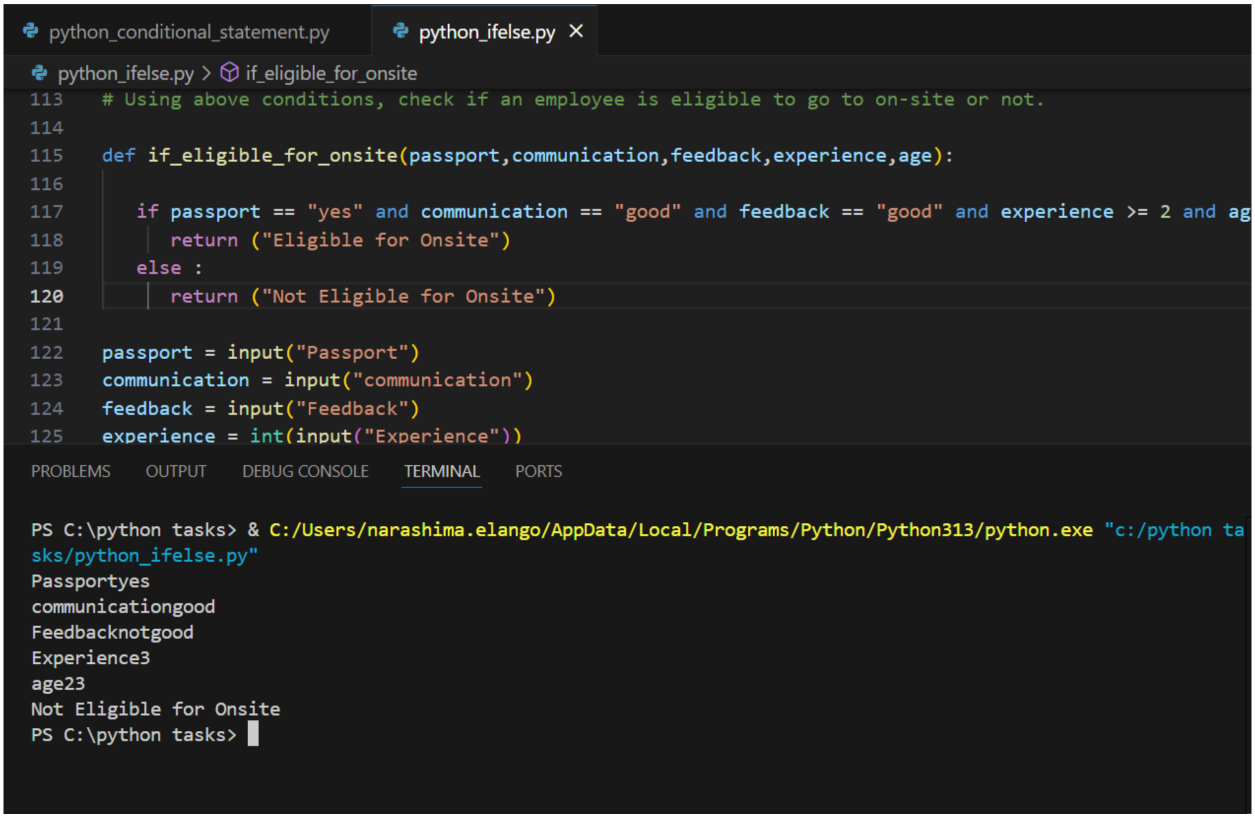
```python
def if_eligible_for_onsite(passport,communication,feedback,experience,age):

  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:


python_conditional_statement.py
python_ifelse.py
python_ifelse.py > if_eligible_for_onsite
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

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

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

    if passport == "yes" and communication == "good" and feedback == "good" and experience >= 2 and age
        return ("Eligible for Onsite")
    else :
        return ("Not Eligible for Onsite")

passport = input("Passport")
communication = input("communication")
feedback = input("Feedback")
experience = int(input("Experience"))
```

```
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
python_ifelse.py > if_eligible_for_onsite
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

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

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

    if passport == "yes" and communication == "good" and feedback == "good" and experience >= 2 and ag
        return ("Eligible for Onsite")
    else :
        return ("Not Eligible for Onsite")

passport = input("Passport")
communication = input("communication")
feedback = input("Feedback")
experience = int(input("Experience"))
```

```
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.

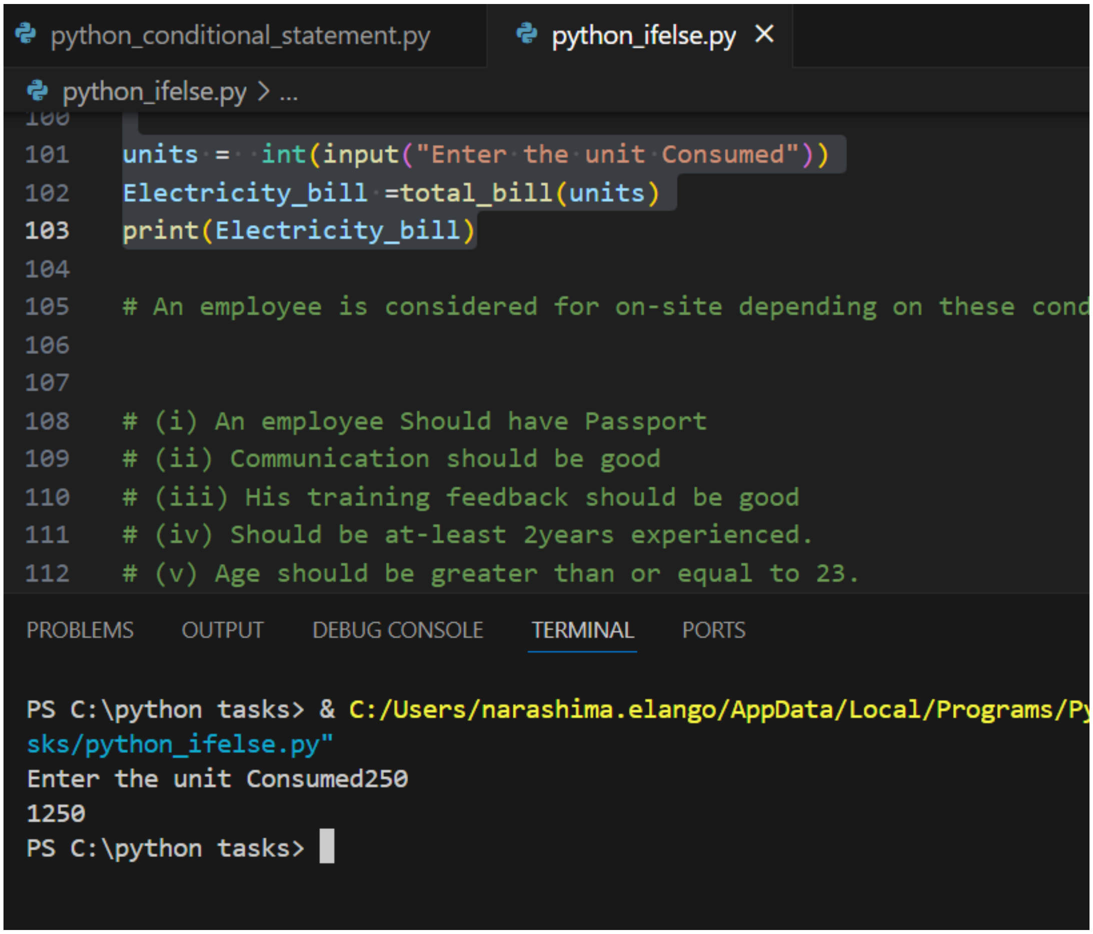
(iii) If units exceeds 200, then charge Rs.2/- per unit.

(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:



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

# An employee is considered for on-site depending on these cond


# (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.
```

```
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> 
```

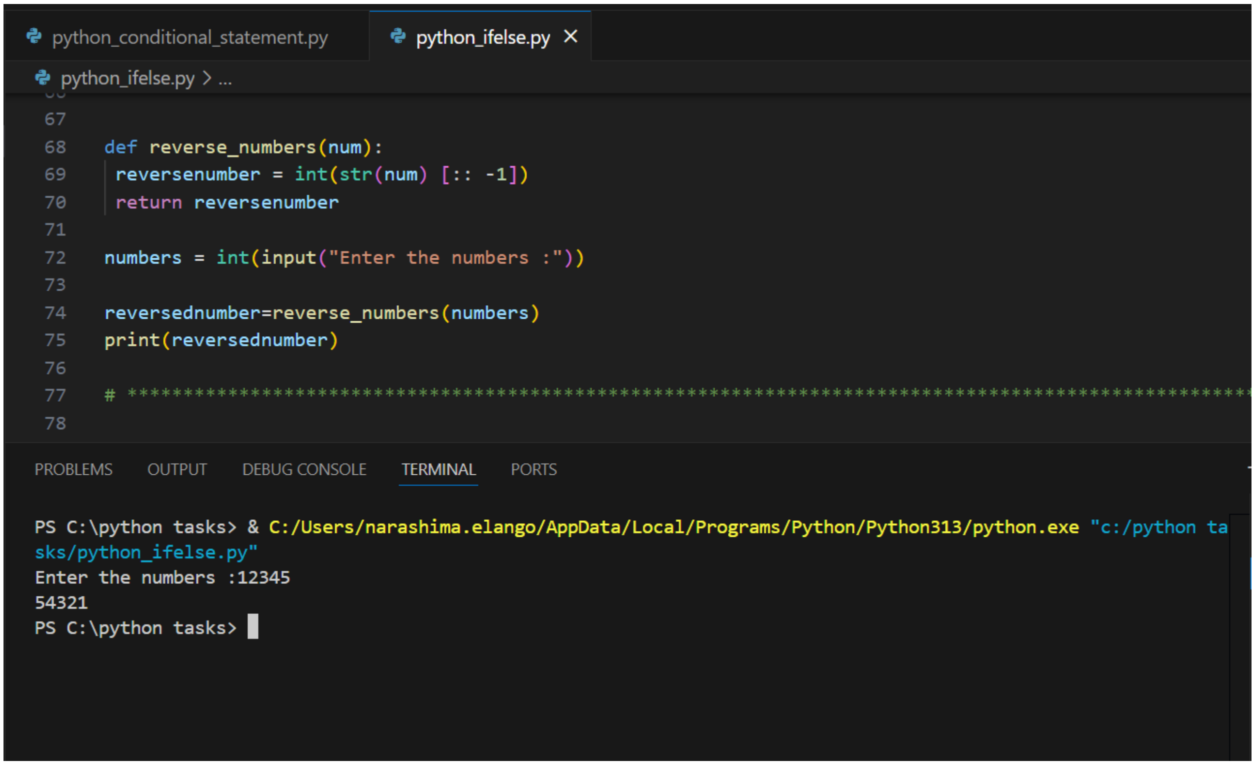
4.Accept a 5 digit decimal number as input and display the number in reverse order.

(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:



```

def reverse_numbers(num):
 reversenumber = int(str(num) [:: -1])
 return reversenumber

numbers = int(input("Enter the numbers :"))

reversednumber=reverse_numbers(numbers)
print(reversednumber)

# ************************************************************************************************************

```

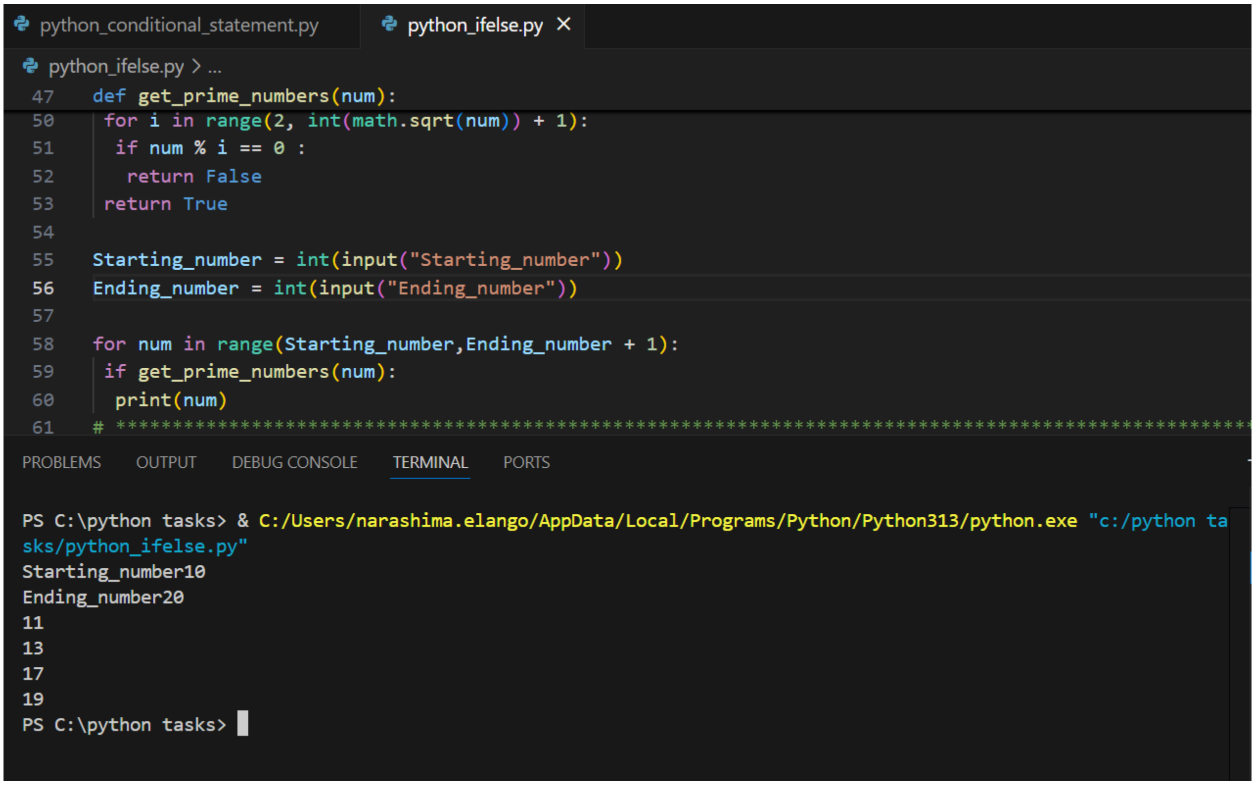
```
PS C:\python tasks> & C:/Users/narashima.elango/AppData/Local/Programs/Python/Python313/python.exe "c:/python ta
sks/python_ifelse.py"
Enter the numbers :12345
54321
PS C:\python tasks>
```

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:



```
def get_prime_numbers(num):
 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)
# ****************************************************************************************************
```

```
PS C:\python tasks> & C:/Users/narashima.elango/AppData/Local/Programs/Python/Python313/python.exe "c:/python ta
sks/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,

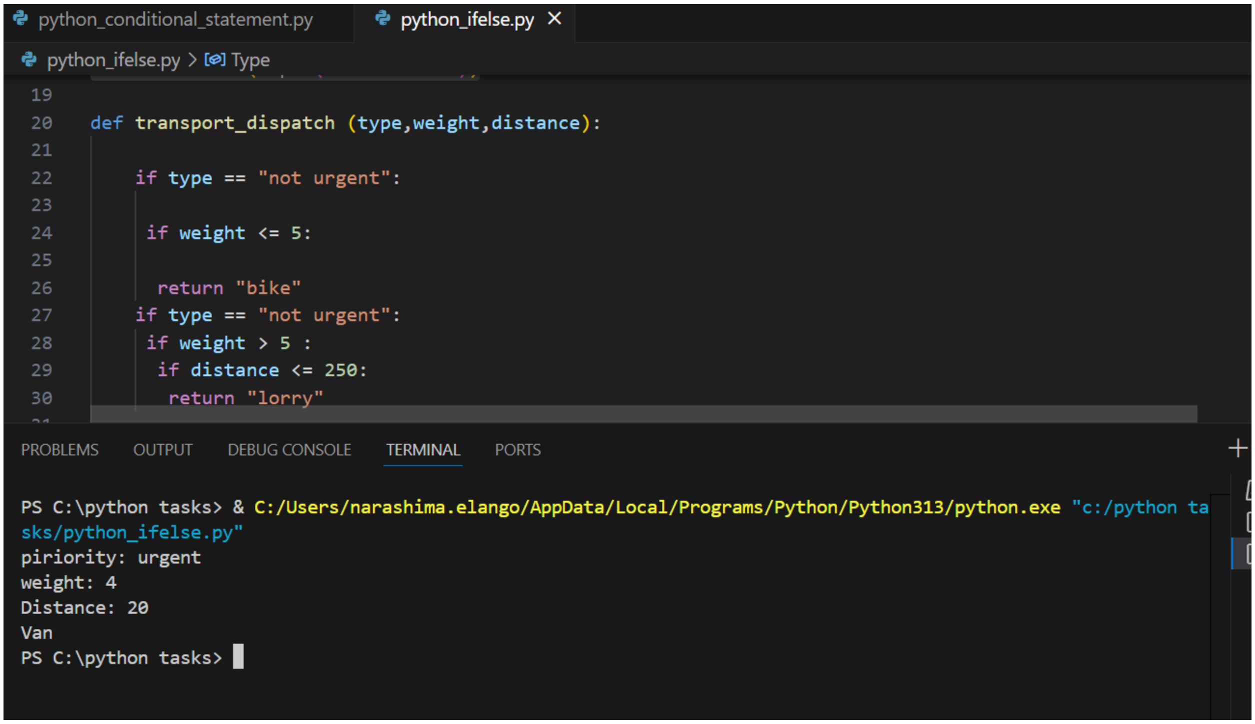
Select a van

d) In all other cases, use a train

Code

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

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:



```python

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"
```

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