1. Write a program to accept percentage from the user and display the grade according to the following criteria:

Marks	Grade
>90	A
>80 and <=90	B
>=60 and <=80	C
Below 60	D

Code:

```
marks = float(input("Enter the marks:")) if marks
> 90:     grade = "A" elif marks > 80 and marks
<= 90:
   grade = "B" elif marks >= 60 and
marks <= 80:     grade = "C" else:
grade = "D" print("Grade: " + grade)
```

OUTPUT:

```
Enter the marks: 65
Grade: C
```

2. Write a program to accept the cost price of a bike and display the road tax to be paid according to the following criteria

Tax	Cost Price (in Rs)
15%	>100000
10%	>50000 and <=100000
5%	<=50000

CODE:

```
cost_price = float(input("Enter cost price of bike (in Rs): ")) if cost_price >
100000:
   road_tax = cost_price * 0.15 elif cost_price > 50000 and
cost_price <= 100000:
   road_tax = cost_price * 0.10 else:
   road_tax = cost_price * 0.05 print("Road tax to be paid (in Rs): "
+ str(road_tax))
```

OUTPUT:

```
Enter cost price of bike (in Rs):  50001
Road tax to be paid (in Rs): 5000.1
```

3. Accept any city from the user and display monuments of that city.

City	Monument
Delhi	Red Fort
Agra	Taj Mahal
Jaipur	Jal Mahal

CODE:

```
city = input("Enter a city: ") if
city.lower() == "delhi":
   print("Monuments in Delhi:")     print("Red Fort") elif city.lower() == "agra":
print("Monuments in Agra:")     print("Taj Mahal") elif city.lower() == "jaipur":
print("Monuments in Jaipur:")     print("Jal Mahal") else:     print("Sorry, we do not have
information about the monuments in that city.")
```

OUTPUT:

```
Enter a city:  KOLKATA
Sorry, we do not have information about the monuments in that city.
```

4. Check how many times a given number can be divided by 3 before it is less than or equal to 10.

CODE:

```
def count_divisions(number, divisor):
count = 0                    while number % divisor
== 0:                        number //= divisor
        count += 1                   return count
  n=int(input("enter a number"))     result
= count_divisions(n, 3)      print(result)
```

OUTPUT: enter a number 30
1

5. Why and When to Use while Loop in Python give a detailed description with example A while loop in Python is used to execute a block of code repeatedly as long as a certain condition is true. It is a type of loop that continues to run as long as the condition is true and stops running when the condition becomes false. The syntax for a while loop is:

while condition: # code to execute

Here are some situations when you might use a while loop in Python:

1. When you want to repeat a block of code until a certain condition is met. For example, you might use a while loop to keep prompting the user for input until they provide a valid response.

   CODE:

```
valid_input = False while not
valid_input:
user_input = input("Enter a number: ")          if
user_input.isdigit():                  valid_input = True
num = int(user_input)                  else:
                        print("Invalid input. Please enter a number.")
```

   OUTPUT:
   Enter a number: 10

2. When you want to perform a task a specific number of times. For example, you might use a while loop to iterate through a list until a certain condition is met.

   CODE:

```
my_list = [1, 2, 3, 4, 5] index = 0
while index < len(my_list):
print(my_list[index])                  index +=
1
```

   OUTPUT:
   1
   2
   3
   4
   5

3. When you want to continuously perform a task until the program is interrupted or stopped. For example, you might use a while loop to run a game loop that updates the game state and redraws the screen.

6. Use nested while loop to print 3 different patterns.

Pattern 1: A right triangle made of asterisks

CODE:

```
i = 1
while i <= 5:
    j = 1
    while j <= i:
        print("*", end="")
        j += 1
    print()
    i += 1
```

OUTPUT:

```
*
**
***
****
*****
```

Pattern 2: A square made of asterisks

CODE:

```
i = 1
while i <= 5:
    j = 1
    while j <= 5:
        print("*", end="")
        j += 1
    print()
    i += 1
```

OUTPUT:

```
*****
*****
*****
*****
*****
```

Pattern 3: A reverse right triangle made of asterisks

CODE:

```
i = 5
while i >= 1:
    j = 1
    while j <= i:
        print("*", end="")
        j += 1
    print()
    i -= 1
```

OUTPUT:

```
*****
****
***
** *
```

7. Reverse a while loop to display numbers from 10 to 1.

CODE:

```
i = 10
while i >= 1:
    print(i)
    i -= 1
```

OUTPUT:

```
10
9
8
7
6
5
4
3
2
1
```