Solutions:

1]

marks = int(input("Enter Percentage:"))

if marks>90:

print("Grade: A")

elif marks>80:

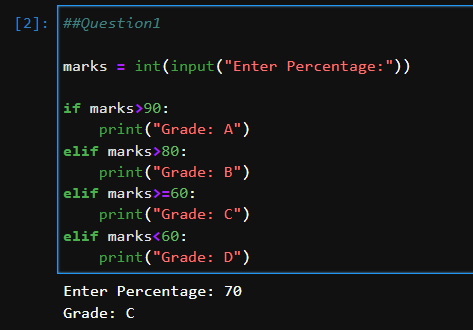
print("Grade: B")

elif marks>=60:

print("Grade: C")

elif marks<60:

print("Grade: D")



2]

cost\_price = int(input("Enter Cost Price of a Bike:"))

if cost\_price>100000:

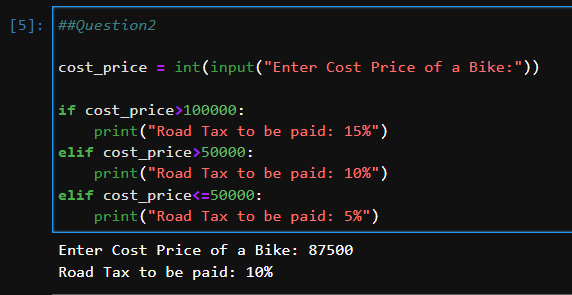
print("Road Tax to be paid: 15%")

elif cost\_price>50000:

print("Road Tax to be paid: 10%")

elif cost\_price<=50000:

print("Road Tax to be paid: 5%")



3]

city = input("Enter City[Agra, Delhi, Jaipur]:")

if city == 'Delhi':

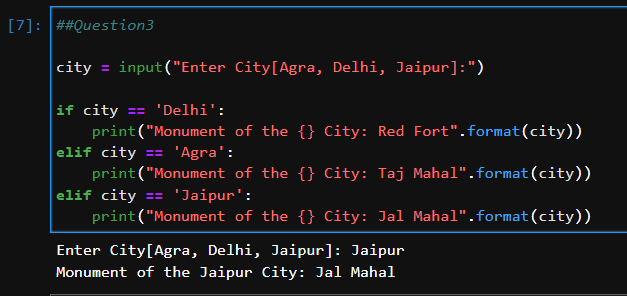
print("Monument of the {} City: Red Fort".format(city))

elif city == 'Agra':

print("Monument of the {} City: Taj Mahal".format(city))

elif city == 'Jaipur':

print("Monument of the {} City: Jal Mahal".format(city))



4]

num = int(input("Enter a Number:"))

count = 0

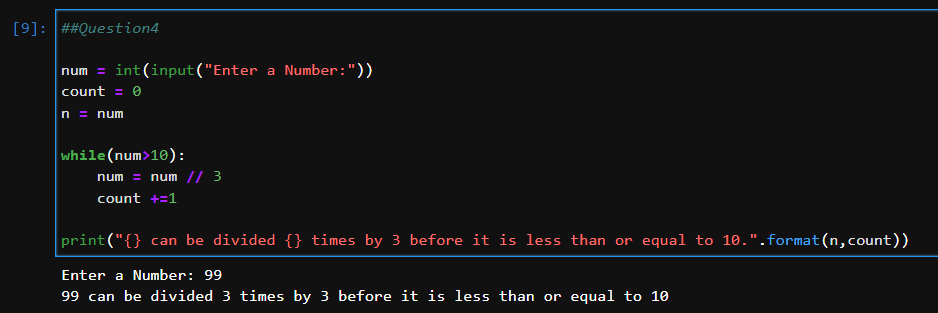
n = num

while(num>10):

num = num // 3

count +=1

print("{} can be divided {} times by 3 before it is less than or equal to 10.".format(n,count))



5]

A while loop in Python is used to execute a block of code repeatedly until a certain condition is met. The condition is checked before each iteration of the loop, and as long as it is True, the loop continues to execute. Once the condition becomes False, the loop terminates, and the program continues to execute the code after the loop.

The syntax of a while loop in Python is as follows:

while condition:

# Code to be executed while the condition is True

The condition is a Boolean expression that determines whether the loop should continue to execute or not. The code inside the loop can include any Python statements, including control flow statements like if, elif, and else.

Here's an example of a while loop that counts from 1 to 5:

count = 1

while count <= 10:

print(count)

count += 1

In the above example, the loop starts with a count variable set to 1. The while loop checks whether the count is less than or equal to 10. Since 1 is less than or equal to 10, the loop executes the first iteration, which prints 1 and increments count by 1. The loop checks the condition again and finds that 2 is less than or equal to 10, so it executes the second iteration, which prints 2 and increments count by 1. This process continues until count becomes 11, at which point the condition becomes False, and the loop terminates.

You can use a while loop in Python in situations where you need to repeat a block of code until a certain condition is met. For example, you can use a while loop to read user input until the user enters a valid value:

while True:

age = input("Enter your Age: ")

if age.isdigit():

age = int(age)

break

else:

print("Invalid Input, Please Enter a valid Age.")

In this example, the loop reads user input using the input() function and checks whether the input is a valid integer using the isdigit() method of the string object. If the input is valid, the loop converts the input to an integer using the int() function and breaks out of the loop. If the input is not valid, the loop prints an error message and continues to execute until the user enters a valid input.

6]

n = int(input('Enter Number of Rows : '))

#Pattern1

i = 1

while i <= n :

j = 1

while j <= i:

print("\*", end = " ")

j += 1

print("")

i += 1

print("\n")

#Pattern2

i = 1

while i <= n:

j = n

while j >= i:

print("\*", end=" ")

j-=1

print("")

i+=1

print("\n")

#Pattern3

i=1

while i <= n:

j = i

while j <= n-1:

print(" ", end=' ')

j+=1

k = 1

while k <= i:

print("\*", end=' ')

k+=1

l = 1

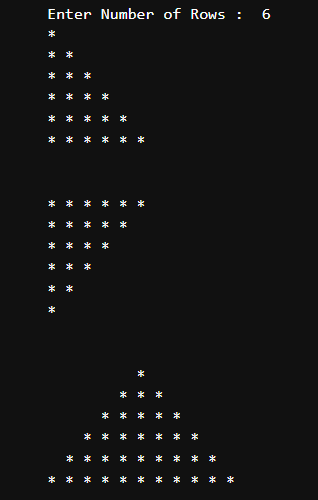
while l <= i-1:

print("\*", end=' ')

l+=1

print("")

i+=1



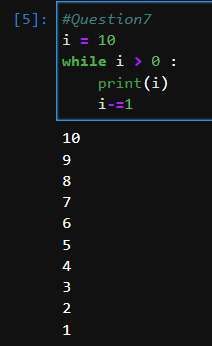
7]

i = 10

while i > 0 :

print(i)

i-=1



8]

i = 10

while i > 0 :

print(i)

i-=1

