**Q1**

"True"

"False"

**Q2**

Logical AND (and): The logical AND operator returns True if both operands are True, and False otherwise. It's denoted by the keyword and in Python.

Logical OR (or): The logical OR operator returns True if at least one of the operands is True, and False otherwise. It's denoted by the keyword or in Python.

Logical NOT (not): The logical NOT operator returns the opposite of the operand's value. If the operand is True, it returns False, and if the operand is False, it returns True. It's denoted by the keyword not in Python

**Q3**

Logical AND (and):

Operand A Operand B Result

False False False

False True False

True False False

True True True

Logical OR (or):

Operand A Operand B Result

False False False

False True True

True False True

True True True

Logical NOT (not):

Operand Result

False True

True False

**Q4**

False

False

True

False

False

True

**Q5**

Equal (==):

Checks if two values are equal to each other.

Not Equal (!=):

Checks if two values are not equal to each other.

Greater Than (>):

Checks if the left operand is greater than the right operand.

Less Than (<):

Checks if the left operand is less than the right operand.

Greater Than or Equal To (>=):

Checks if the left operand is greater than or equal to the right operand.

Less Than or Equal To (<=):

Checks if the left operand is less than or equal to the right operand.

**Q6**

Equal To Operator (==):

The equal to operator is used to compare two values to check if they are equal to each other. It returns a Boolean value (True or False) based on the comparison.

Example:

x = 5

y = 10

print( x == y)

# False

Assignment Operator (=):

The assignment operator is used to assign a value to a variable. It assigns the value on the right-hand side to the variable on the left-hand side.

Example:

x = 5 # Assigns the value 5 to the variable x

**Q7**

Block 1:

spam = 0

if spam == 10:

print('eggs')

Block 2:

if spam > 5:

print('bacon')

else:

print('ham')

Block 3:

print('spam')

print('spam')

**Q8**

spam = int(input("enter the value of spam"))

if spam == 1:

print("hello")

elif spam == 2:

print("Howdy")

else:

print("Greetings")

**Q9**

break

**Q10**

When "break" is used, the loop is immediately terminated, and the program execution continues with the next statement after the loop. "Break" is commonly used when we want to exit a loop early based on some condition.

"Continue" is used when you want to skip certain iterations based on a specific condition but continue with the loop.

**Q11**

There is no difference between range(10), range(0,10) and range(0,10,1) in case of for loop

**Q12**

**#**using for loop

for i in range(1,11):

print(i)

#using while loop

i=1

while i<11:

print(i)

i+=1

**Q13**

import spam

spam.bacon()