1.What are the two values of the Boolean data type? How do you write them?

Ans : The two values of the Boolean data type are typically represented as **"True"** and **"False".**

condition = True

while condition:

print("This loop is running.")

2. What are the three different types of Boolean operators?

Ans : The three different types of Boolean operators are:

1. **AND**: The AND operator returns true if and only if both operands are true. Otherwise, it returns false.
2. **OR**: The OR operator returns true if at least one of the operands is true. It returns false only if both operands are false.
3. **NOT**: The NOT operator negates the value of its operand. If the operand is true, the NOT operator returns false, and vice versa.

3. Make a list of each Boolean operator's truth tables (i.e. every possible combination of Boolean values for the operator and what it evaluate).

AND Operator:

A	B	A AND B
False	False	False
False	True	False
True	False	False
True	True	True

OR Operator :

A	B	A OR B
False	False	False
False	True	True
True	False	True
True	True	True

NOT Operator:

A	NOT A
False	True
True	False

4. What are the values of the following expressions?

Ans : (5 > 4) and (3 == 5) - > False

not (5 > 4) -> False

(5 > 4) or (3 == 5) -> True

not ((5 > 4) or (3 == 5)) -> False

(True and True) and (True == False) -> False

(not False) or (not True) -> True

5. What are the six comparison operators?

Ans : The six comparison operators are:

i) **Equal to (==)**

ii) **Not equal to (!=)**

iii) **Greater than (>)**

iv) **Less than (<)**

v) **Greater than or equal to (>=)**

vi) **Less than or equal to (<=)**

6. How do you tell the difference between the equal to and assignment operators? Describe a condition and when you would use one.

Ans : In pyhton language, the equal to operator (**==**) is used for comparison, while the assignment operator (**=**) is used to assign a value to a variable.

Example : x == 5, compares the values of ‘x’ with 5.

x = 5, assigns the value 5 to ‘x’

7. Identify the three blocks in this code:

```
spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')
```

else:

print('ham')

print('spam')

print('spam')

Ans :

Main Block: Contains the variable assignment spam = 0.

First if Block: Contains the statement **print('eggs')** and is executed if **spam == 10**.

Second if Block: Contains the statement **print('bacon')** and is executed if **spam > 5**. If it is false, **'else'** statement is printed.

8. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam.

Ans :

```
spam = 1
if spam == 1:
   print('Hello')
elif spam == 2:
   print('Howdy')
else:
   print('Greetings!')
```

9.If your programme is stuck in an endless loop, what keys you'll press?

Ans : Pressing **Ctrl + C** sends a keyboard interrupt signal to the program, causing it to terminate. This key combination is often used to stop programs that are running indefinitely or to break out of an unintended infinite loop.

10. How can you tell the difference between break and continue?

Ans : The **break** and **continue** statements are both used in loops (such as **for** loops and **while** loops) in Python to control the flow of execution.

'Break' Statement :

- The break statement is used to exit or terminate the loop prematurely.

- When encountered within a loop, the **break** statement immediately exits the loop, regardless of whether the loop's condition has been met.

'Continue' statement:

- The **continue** statement is used to skip the rest of the code inside the loop for the current iteration and move to the next iteration.
- When encountered within a loop, the **continue** statement skips the remaining code in the loop for the current iteration and proceeds with the next iteration.

11. In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?

Ans : **Range(10)** -> This creates a sequence of numbers starting from **0** up to (but not including) **10**.

Range(0,10) -> This explicitly specifies the start and end points of the sequence.

Range(0, 10, 1) -> This explicitly specifies the start, end, and step size of the sequence.

12. Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.

Ans :

```
# Using a for loop

for i in range(1, 11):
    print(i)

# Using a while loop

i = 1

while i <= 10:
    print(i)
    i += 1
```

13. If you had a function named bacon() inside a module named spam, how would you call it after importing spam?

Ans :

```
import spam

spam.bacon()
```