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

*****Ans:-*** The two values of Boolean data type are True and False.

It can be written as a=True
b=False

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

*****Ans :-*** The 3 different types of Boolean operators are – **and , or , not**

- "and" Operator: The "and" operator returns True if both operands are True, and False otherwise.
- "or" Operator: The "or" operator returns True if at least one of the operands is True, and False if both operands are False.
- "not" Operator: The "not" operator returns the negation of the operand. It returns True if the operand is False, and False if the operand is True.
- **Example:-**

```
a = True
b = False
print(a and b) # and operator
print(a or b)    # or operator
print(not a)   # not operator
```

o/p :-

```
  False
  True
  False
```

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).

*****Ans :-***

and operator :-

x	y	X and y
True	True	True
False	False	False
True	False	False

False	True	False

or operator :-

X	Y	X or y
False	False	False
True	False	True
False	True	True
True	True	True

Not operator:-

x	Not x
false	true
true	false

4. What are the values of the following expressions?

(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 :-

1. Equal to , 2. Not equal to , 3. Greater than, 4. Less than , 5. Greater than or equal to ,   6. Less than or equal to
   a=10

```
b=20
print(a == b)  # Equal to, Output: False
print(a != b)  # Not equal to, Output: True
print(a > b)   # Greater than, Output: False
print(a < b)   # Less than, Output: True
print(a >= b)  # Greater than or equal to, Output: False
print(a <= b)  # Less than or equal to, Output: True
```

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

*****Ans :-*** The equal to (==) operator is used to compare two values to check if they are equal or not and it returns Boolean values i.e true or false . Whereas the assignment operator (=) is used to assign values to the variable.

Eg:- Equal to operator (==)

```
x = 5
y = 10
if x == y:
    print("x and y are equal")
else:
   print("x and y are not equal")
```

In this case, the program will output "x and y are not equal" since the values of x and y are different.

Eg:- Assignment operator(=)

```
x=5
y=x
print(y)
#output=5
```

In this case, the value of x (which is 5) is assigned to the variable y, so when y is printed, it will display the value 5.

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 :-** The following code is divided into 3 parts .

1. If spam is exactly equal to 10 then it will print eggs and later it will print bacon too as the 2nd condition states that the no. should be greater than 5.
2. If spam is any no. which is greater than 5 then it will print bacon.
3. If spam is any number which is smaller than 5 , it can be negative too then it will print ‘ham’ , ‘spam’ , ‘spam’.

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

o/p- Howdy

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

Ans :- You can press **ctrl + c** if your programme is stuck in endless loop.

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

Ans :-

- Break statement terminates the whole process of loop once specified condition is met.

  Eg:-

  ```
  for i in range(1, 6):
   if i == 3:
       break
   print(i)
  ```

  o/p – 1

  ```
  2
  ```

- The continue statement skips the remaining code inside a loop for the current iteration only.

  Eg :- for i in range(1, 6):

  ```
     if i == 3:
         continue
     print(i)
  ```

  o/p –
  ```
  1
  2
  4
  5
  ```

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

Ans :-

- ***range(10)*** - **range(start)**

  The 10 is no. till where the range should stop. The range starts from 0 so it will end at 9.

  ```
  for i in range(10):   # range(stop)
     print(i)
  ```

  o/p –
  ```
  0
  1
  2
  3
  4
  5
  6
  7
  8
  9
  ```

- ***range(0, 10)*** - **range(start ,stop)**
  The 1st number i.e 0 indicates the start point of the range , and the 2nd number i.e 10 indicates the stop point of the range.

```
for i in range(4,10): # range(start, stop)
   print(i)
o/p -   4
         5
         6
         7
         8
         9
```

- ***range(0, 10, 1)*** **– range(start, stop, step)**
  The 1st no. i.e 0 indicates the start point of the range ,whereas the 2nd no. i.e 10 indicates the stop point of the range, and the last no. that is 1 indicates that how much gap should be given between numbers in range.

```
for i in range(0, 10, 2):  #range(start, stop, step)
   print(i)

o/p –   0
         2
         4
         6
         8
```

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 for loop –**

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

Using while loop –

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

o/p for both the programs :-

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

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

Ans :- This function can be called with spam. bacon()