1. **What are logical operators? How many are they?**
   Ans: Logical operators are used to combine Conditional statements. They are 3 types: AND, OR, NOT.
   AND: Returns True if both statements are true
   OR: Returns True if one of the statements is true
   NOT: Reverse the result, returns False if the result is true
2. **What is d/f b/w the logical AND & logical OR ?**
   Ans: Logical AND returns True if both statements are true whereas, Logical OR returns True if one of the statements is true.
3. **What are membership operators ? how many are they?**
   Ans: Membership operators are used to test if a sequence is presented in an object or not. They are of 2 types: IN and NOT IN
   IN: Returns True if a sequence with the specified value is present in the object
   NOT IN: Returns True if a sequence with the specified value is not present in the object.
4. **what is d/f b/w in and not in operators?**
   Ans: IN operator Checks if a value exists and True if found, otherwise False.
   NOT IN Checks if a value does NOT exist and True if not found, otherwise False.
   Ex1: #IN

```
fruits = ["orange", "banana", "mango"]
if "banana" in fruits:
   print("Banana is in the list")              # Banana is in the list
```

   Ex2: #NOT IN

```
fruits = ["pineapple", "banana", "Kiwi"]
if "grape" not in fruits:
   print("Grape is not in the list")           # Grape is not in the list
```

5. **what is the d/f b/w == and != operators ?**
   Ans: "=="operator Checks if two values are equal and
   True if values are the same.
   "!= " Checks if two values are NOT equal
   True if values are different.
6. **What are conditional statements in python ? write a syntax ans simple example ?**
   Ans: Conditional statements are used to make decisions in a program. They let the program execute certain blocks of code only if a condition is True.
   if – checks a condition
   if-else – checks and provides an Alternative
   if-elif-else – checks multiple conditions.
   Ex:

```
marks = 79
if marks >= 50:
   print("You passed")
else:
   print("You failed")                  # You passed
```

7. **Write a program to demonstrate the if-else conditions?**
   Ans:

```
a = int(input("enter a: "))
```

```
b = int(input("enter b: "))
c = int(input("enter c: "))
if (a>b and a>c):
print(" a is largest: ")
elif (b>a and b>c):
print("b is largest")
else:
print("c is largest")
Output: enter a: 5
enter b: 9
enter c: 2
b is largest
```

8. write if-else-if-else ladder with a simple example ?

```
buyer = input("Are you a first-time buyer(YES or No): ")
cart_total = int(input("enter cart total:"))
if buyer == "yes":
  if cart_total >= 1000:
      print("30% discount applied")
  else:
       print("10% discount applied")
elif buyer == "no":
   if cart_total >= 2000:
     print("15% discount applied")
   else:
       print("no discount")
else:
print("enter valid choice")
OUTPUT: Are you a first-time buyer(YES or No): yes
enter cart total:1050
30% discount applied
```

9. write a program to demonstrate how nested conditions works in python?

```
Ans: age = int(input("Enter your age: "))
has_voter_id = input("Do you have a voter ID? (yes/no): ")

if age >= 18:
   if has_voter_id == "yes":
      print("You are eligible to vote.")
   else:
      print("You are old enough, but you need a voter ID to vote.")
else:
   print("You are not eligible to vote.")
```

10. What is indentation in python ? what is importance of it in python? explain with an example?

Ans: Indentation in Python means spaces or tabs used at the beginning of a line to indicate a block of code.
It shows which lines of code belong together.
Python will not run without proper indentation.
It makes the code clean, structured, and readable.
Ex: # Incorrect indentation

```
age = 18
if age >= 18:
print("You are an adult")
print("You can vote")
```

11. what is error and how many type of errors do you know?

Ans: An error is a problem in a program that stops it from running correctly.
When Python finds an error, it throws an error message and stops the program.
Types of errors:

1. **Syntax Errors**: These occur when the code does not conform to the rules of the programming language.
2. **Runtime Errors**: These errors occur while the program is running, often due to illegal operations, such as dividing by zero or accessing an out-of-bounds index in an array.
3. **Logical Errors**: These are mistakes in the logic of the program that lead to incorrect results, even though the code runs. ex: using the wrong formula to calculate a value.
4. **Compilation Errors**: These occur when the code cannot be compiled due to syntax issues or other problems that prevent the compiler from generating executable code.
5. **Type Errors**: These occur when an operation is performed on an inappropriate data type, such as trying to add a string to an integer.
6. **Off-by-One Errors**: Common in loops and array indexing, these occur when an iteration or index is incorrectly set to one more or one less than intended.
7. **Semantic Errors**: These occur when the code is syntactically correct but does not produce the intended meaning or result.

12. Write an example each to demonstrate syntax error and name Error and keyError ?

Ans: **Syntax Error:**
Occurs when you break the rules of Python syntax.

```
# Missing colon at the end of if statement
x = 5
if x > 0
   print("Positive number")
```

Name Error:
Occurs when you try to use a variable or function that hasn't been defined.

```
A = 20
Print(b)  #b is not declared
```

Key Error:
Occurs when you try to access a dictionary key that does not exist.

```
person = {"name": "Maya", "age": 25}
```

```
print(person["height"])          # Key 'height' doesn't exist
```

13. what is loop and how many types of loops are there in python?

Ans: A loop is a control structure that allows for the repeated execution of a block of code as long as a specified condition is true.

FOR: Iterates over a sequence

While: Repeats as long as a specified condition is true.

Ex:
```
for i in range(1, 6):
    print(i)
```
Ex:
```
count = 1
while count <= 5:
    print(count)
    count += 1
```

14. Write an example for for loop using list?

```
list1 = [2,3,4,5]
prod = 1
for i in list1:
    prod *= i
print(prod)
```
output: 120

15. write an example for for loop using str and dict and tuple ?

#STR:
```
lis = ["apple", "banana", "kiwi", "orange"]
for i in lis:
    print(i)
```
output: APPLE BANANA KIWI ORANGE

#DICT:
```
person = {"name": "Maya", "age": 29, "city": "Delhi"}
for key in person:
    print(key, ":", person[“name”])   # name:Maya
```
#TUPLE:
```
a = (1,4,5,6,7)
sum = 0
for i in a:
sum += i
print(sum)
```
OUTPUT: 1 5 10 16 23