Question 1: Why do we call Python as a general purpose and high-level programming language?

Python is a high-level multi-paradigm programming language with a simple and easy to read syntax. By using Python, you can work quickly and integrate systems more effectively to create highly readable, efficient programs and web applications. Python combines exceptional power with clear, user-friendly syntax; the language incorporates such features as classes, modules, exceptions, dynamic typing, and high level dynamic data types.

1. **Powerful:** The language incorporates such features as exceptions, dynamic typing, classes, modules, and high level dynamic data types.
2. **Robust:** Python's standard library includes almost any area of functionality you need.
3. **Portable**: Python runs on modern Windows, Mac, and many Unix variants.
4. **Explicit:** Python can do more with less code thanks to the intentionally explicit syntax that it uses.

Question 2: Why is Python called a dynamically typed language?

Because the things we call "variables" in Python are names which are bound to objects after they've been created (it's a late binding dynamic language). The objects have type; but any variable name can be bound and rebound to any object regardless of types.

Question 3: List some pros and cons of Python programming language?

Pros:
Easy to Use Programming Language
Lowest Learning Curve
Great for Visualising Data
Easy to Read Language
Unmatched Flexibility
Asynchronous Coding
Powerful Programming

Cons:
Speed Limitations
Threading Issues
Simplicity

Question 4: In what all domains can we use Python?

1. Web development
2. Data science
3. OS development
4. Scientific programming
5. Gaming

Question 7: What are variable and how can we declare them?

Python is a dynamic-typed language, which means we don't need to mention the variable type or declare before using it. Before declaring a variable, we must follow the given rules.

- The first character of the variable can be an alphabet or (_) underscore.
- Special characters (@, #, %, ^, &, *) should not be used in variable name.
- Variable names are case sensitive. For example - age and AGE are two different variables.
- Reserve words cannot be declared as variables**.**

Example: variable = “This is a variable”
Num_var = 10
a=b=c = “This is also variable declaration”

Question 6: How can we take an input from the user in Python?

To receive information through the keyboard, Python uses the **input()** function. This function has an optional parameter, commonly known as prompt, which is a string that will be printed on the screen whenever the function is called.

Question 7: What is the default datatype of the value that has been taken as an input using input() function?

Python input() function is used to take user input. By default, it returns the user input in form of a **string.** But we can make a condition check to get the desired output from the persons.

Question 8: What is type casting?

The conversion of one data type into the other data type is known as type casting in python or type conversion in python. Python supports a wide variety of functions or methods like: **int(), float(), str(), ord(), hex(), oct(), tuple(), set(), list(), dict(), etc.** for the type casting in python.

There are two varieties of typecasting in python namely **- Explicit Conversion(Explicit type casting in python), and Implicit Conversion(Implicit type casting in python).**

Question 9: Can we take more than one input from the user using single input() function? If yes, how? If no, why?

Yes, we can do it but that's quite complex.
We can do that by two ways

1. Using the .split() method
2. Using the List comprehension method.

Question 10: What are KeyWords in Python?

Keywords are reserved words in Python that are used to trigger specific tasks. We cannot use a keyword as a variable name, function name, or any other identifier since these keywords each hold a special meaning. All of the keywords in Python are lowercase.
Example: print, while, for, continue, break, pass etc

Question 11: Can we use keywords as a variable? Support your answer with reason.

No.

There are two main reasons:

- It makes the developer's life hard; and
- It makes the compiler's life hard.

If keywords were allowed as variable names, it would be very hard to tell (for the developers and the compilers) whether something was a variable or a keyword. For example, what does the following mean?

if(x == 10)

Is it an if-statement, or calling a function called if? Neither the developer nor the compiler would be able to tell.

Question 12: What is indentation? What's the use of indentaion in Python?

Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

Question 13: How can we throw some output in Python?

We can run a program and once it is done we can use the result of the program to show an output, this can be done in two ways either by using **print()** or **return** commands.

Question 14: What are operators in python?

The operator can be defined as a symbol which is responsible for a particular operation between two operands. Operators are the pillars of a program on which the logic is built in a specific programming language. Python provides a variety of operators, few are listed.

- Arithmetic operators
- Comparison operators
- Assignment Operators etc

Question 15: What is difference between / and // operators in python?

In Python programming, you can perform division in two ways. The first one is Float Division("/") and the second is Integer Division("//") or Floor Division.

- **Float Division("/"):** Divides left hand operand by right hand operand.
- **Floor Division("//"):** The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored , i.e., rounded away from zero (towards negative infinity).

Question 16: Write a code that gives following as an output.

```
iNeuroniNeuroniNeuroniNeuron
```

Code:

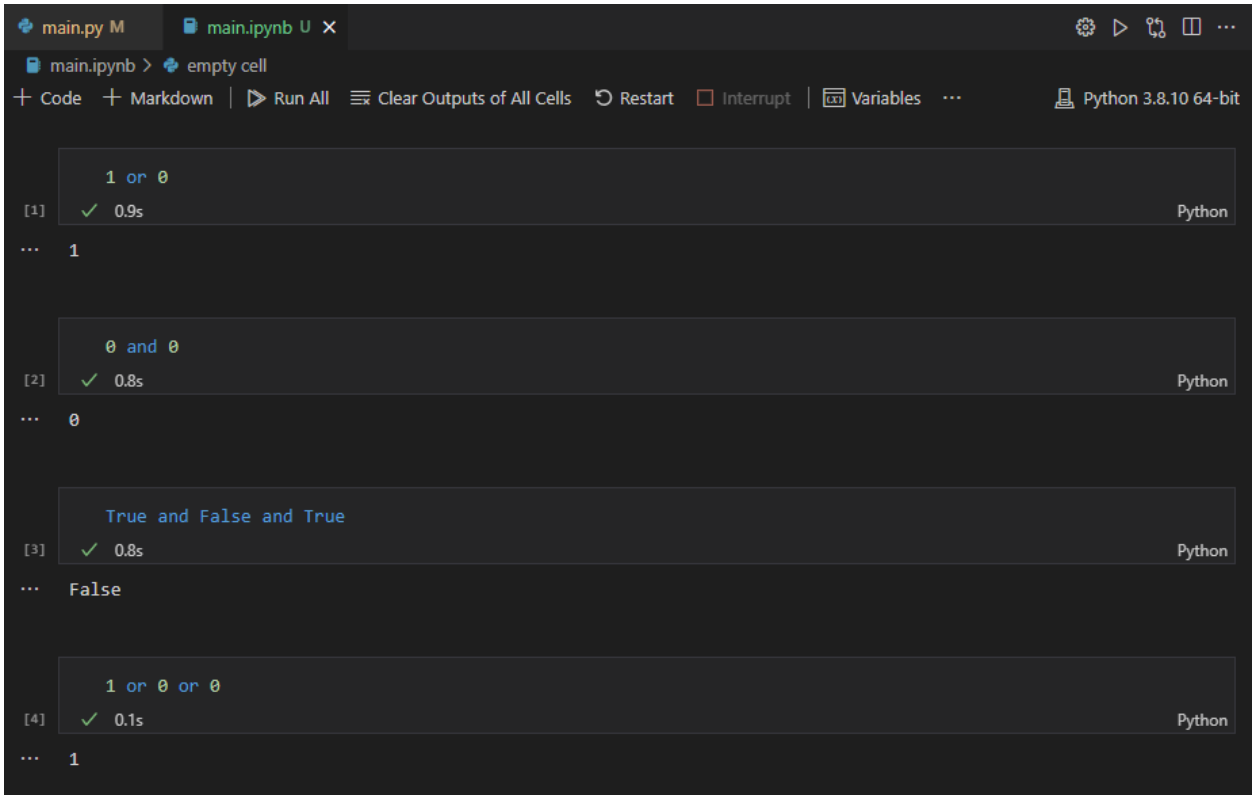
print(“iNeuroniNeuroniNeuroniNeuron”)

Question 17: Write a code to take a number as an input from the user and check if the number is odd or even.

Question 18: What are boolean operator?

Boolean is a set of commands that can be used in almost every search engine, database, or online catalogue. The most popular Boolean commands are AND, OR, and NOT. Other commands include parentheses, truncation, and phrases.

Question 19: What will the output of the following?



Question 20: What are conditional statements in Python?

Conditional Statements In Python
1) if statements
2) if-else statements
3) elif statements
4) Nested if-else statements
5) elif Ladder

Question 21: What is use of 'if', 'elif' and 'else' keywords?

The if/elif/else structure is a common way to control the flow of a program, allowing you to execute specific blocks of code depending on the value of some data.

if statement

If the condition following the keyword if evaluates as true, the block of code will execute. Note that parentheses are not used before and after the condition check as in other languages.

else statement

You can optionally add an else response that will execute if the condition is false.

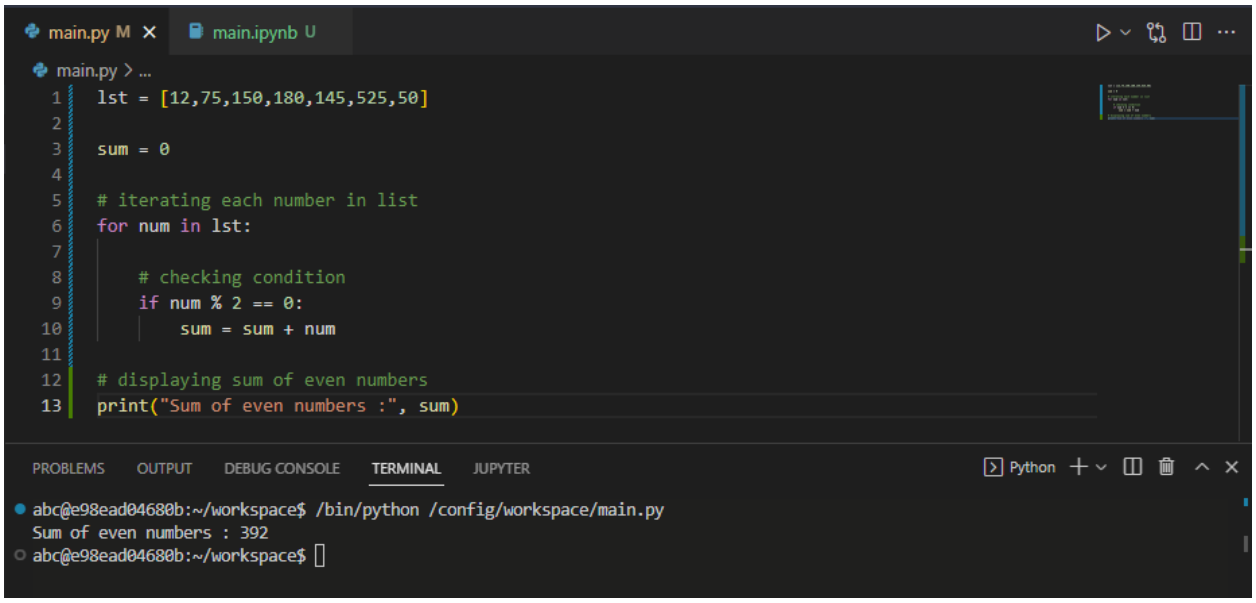
elif statement

Multiple conditions can be checked by including one or more elif checks after your initial if statement. Just keep in mind that only one condition will execute.

Question 22: Write a code to take the age of person as an input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote".

Question 23: Write a code that displays the sum of all the even numbers from the given list.

```
numbers = [12, 75, 150, 180, 145, 525, 50]
```



Question 24: Write a code to take 3 numbers as an input from the user and display the greatest no as output.

Question 25: Write a program to display only those numbers from a list that satisfy the following conditions

- **The number must be divisible by five**
- **If the number is greater than 150, then skip it and move to the next number**
- **If the number is greater than 500, then stop the loop**