

Python practice question

❖ Python Basics:

1. Write a program to check whether a given number is even or odd.
 2. Write a program to add two numbers entered by the user.
 3. Write a program to find the largest of three numbers.
 4. Write a program to calculate the factorial of a number.
 5. Write a program to reverse a given number.
 6. Write a program to count the number of vowels in a given string.
 7. Write a program to print the first n Fibonacci numbers.
 8. Write a program to check whether a given string or number is a palindrome.
 9. Write a program to find the sum of all elements in a list.
 10. Write a program to count the number of positive, negative.
-

❖ Conditional statement:

1. Write a program to check whether a given number is **positive, negative, or zero**.
 2. Write a program to check whether a person is **eligible to vote** based on age.
 3. Write a program to find the **greater of two numbers**.
 4. Write a program to check whether a given year is a **leap year**.
 5. Write a program to find the **largest of three numbers** using conditional statements.
 6. Write a program to determine whether a given character is a **vowel or consonant**.
 7. Write a program to determine the **type of triangle** (Equilateral, Isosceles, Scalene, or Not a Triangle) based on three sides.
 8. Write a program to calculate **income tax** based on salary slabs using conditional statements.
-

❖ Loops:

1. Write a program to print numbers from **1 to n** using a loop.
 2. Write a program to print the **multiplication table** of a given number.
 3. Write a program to find the **sum of first n natural numbers** using a loop.
 4. Write a program to **count the number of digits** in a given number using a loop.
 5. Write a program to **reverse a number** using a loop.
 6. Write a program to check whether a given number is an **Armstrong number**.
 7. Write a program to print all **even numbers between 1 and n** using a loop.
 8. Write a program to generate the **Fibonacci series up to n terms** using a loop.
 9. Write a program to check whether a given number is a **prime number** using a loop.
-

❖ Function:

1. Write a function to add two numbers and return the result.
 2. Write a function to check whether a given number is even or odd.
 3. Write a function to find the square of a number.
 4. Write a function to find the maximum of two numbers.
 5. Write a function to calculate the factorial of a number.
 6. Write a function to check whether a given string is a palindrome.
 7. Write a function that takes a list of numbers and returns the sum of all elements.
 8. Write a function to check whether a given number is a prime number.
 9. Write a function to generate the first n Fibonacci numbers.
 10. Write a function that takes a number as input and returns the count of digits in that number.
-

❖ Exception Handling:

1. Write a program to handle a `ZeroDivisionError` when dividing two numbers.
 2. Write a program that handles an exception when a user enters invalid input instead of an integer.
 3. Write a program to handle a `ValueError` while converting user input to an integer.
 4. Write a program that safely accesses an element from a list and handles an `IndexError`.
 5. Write a program to handle a `KeyError` when accessing a value from a dictionary.
 6. Write a program that uses `try`, `except`, and `else` blocks to divide two numbers.
 7. Write a program that uses multiple `except` blocks to handle different types of exceptions.
 8. Write a program that uses a `finally` block to close a file after reading data.
 9. Write a program to raise a custom exception when a user enters a negative number.
 10. Write a program that handles multiple possible runtime errors in a calculator program.
-

❖ Lists and Tuples:

1. Write a program to find the **sum of all elements** in a list.
 2. Write a program to find the **largest and smallest elements** in a list.
 3. Write a program to **remove duplicate elements** from a list.
 4. Write a program to **reverse a list**.
 5. Write a program to **count the occurrences** of an element in a list.
 6. Write a program to convert a **list into a tuple**.
 7. Write a program to **find the index** of an element in a tuple.
 8. Write a program to **merge two lists** into a single list.
 9. Write a program to **check whether an element exists** in a tuple.
 10. Write a program to unpack elements of a tuple into individual variables.
-

❖ **Strings and Dictionaries:**

1. Write a program to count the number of characters in a string.
 2. Write a program to reverse a given string.
 3. Write a program to check whether a string is a palindrome.
 4. Write a program to count the number of vowels and consonants in a string.
 5. Write a program to find the frequency of each character in a string.
 6. Write a program to create a dictionary from two lists (keys and values).
 7. Write a program to count the frequency of each word in a given sentence using a dictionary.
 8. Write a program to find the key with the maximum value in a dictionary.
 9. Write a program to remove a key from a dictionary safely.
 10. Write a program to check whether a key exists in a dictionary
-

❖ **Set:**

1. Write a program to **find the union** of two sets.
 2. Write a program to **find the intersection** of two sets.
 3. Write a program to **find the difference** between two sets.
 4. Write a program to **check whether one set is a subset** of another set.
 5. Write a program to **remove duplicate elements** from a list using a set.
-

❖ **OS Module Questions:**

1. Write a program to get the current working directory using the os module.
2. Write a program to create a new directory using the os module.
3. Write a program to rename a file using the os module.
4. Write a program to check whether a file or directory exists using the os module.
5. Write a program to list all files and folders in a given directory using the os module.

❖ Oops:

1. Write a program to create a class and object and display a message.
 2. Write a program to create a class with instance variables and display their values.
 3. Write a program to create a class with a method that prints student details.
 4. Write a program to demonstrate the use of a constructor (`__init__`).
 5. Write a program to create a class and access its methods using an object.
 6. Write a program to demonstrate single inheritance.
 7. Write a program to demonstrate method overriding.
 8. Write a program to demonstrate encapsulation using private variables.
 9. Write a program to create a class method and a static method.
 10. Write a program to demonstrate multiple objects of a class.
 11. Write a program to demonstrate multiple inheritance.
 12. Write a program to demonstrate multilevel inheritance.
 13. Write a program to demonstrate polymorphism using method overriding.
 14. Write a program to create an abstract class and implement its methods.
 15. Write a program to demonstrate operator overloading using OOP concepts.
-

❖ File Handling:

1. Write a program to **create a file** and write some text into it.
2. Write a program to **read the contents** of an existing file.
3. Write a program to **append data** to an existing file.
4. Write a program to **count the number of lines** in a file.
5. Write a program to **count the number of words** in a text file.
6. Write a program to **count the number of characters** in a file.
7. Write a program to **search for a specific word** in a file.
8. Write a program to **copy the contents** of one file into another file.
9. Write a program to **read a file line by line** using a loop.