Python Test

Q1. Write a Python program that defines a function called "add_numbers" that takes two arguments (i.e.,
numbers) and returns their sum. Within the function, add the two numbers together and return the result
using the return statement. Call the function with the values 5 and 6, and print out the returned result.
This will result in the addition of 5 and 6, with the output of the program being the sum of these two
numbers.

Q2. Write a Python program that calculates the square root of a given number using a built-in function.
Specifically, the program should take an integer or float input from the user, calculate its square root
using the 'sqrt()' function from the 'math' module, and print out the result to the user. As an example,
calculate the square root of the number 625 using this program, which should output the value of 25.

Q3.Write a program that prints all prime numbers between 0 to 50.

Q4.How can we swap the values of three variables (let's say a, b, and c) without using a fourth variable?
For example, if we have a=5, b=8, and c=9, how can we obtain a=9, b=5, and c=8? The challenge is to
perform this operation without using an additional variable to store any of the values during the swapping process.

Q5. Can you write a program that determines the nature of a given number (in this case, 87) as being
positive, negative, or zero? The program should be designed to take the number as input and perform the
necessary calculations to determine if the number is positive (i.e., greater than zero), negative (i.e., less
than zero), or zero (i.e., equal to zero). The output of the program should indicate which of these three
categories the given number falls into.

Q6. How can you create a program that determines whether a given number (in this case, 98) is even or
odd? The program should be designed to take the number as input and perform the necessary calculations to determine whether it is divisible by two. If the number is divisible by two without leaving a
remainder, it is an even number, and if there is a remainder, it is an odd number. The output of the
program should indicate whether the given number is even or odd.

Q7.Write a program for sum of digits.The digits are 76543 and the output should be 25.

Q8.Write a program for reversing the given number 5436 and the output should be 6345.

Q9.Write a program to check if a given number 371 is an Armstrong number?

Q10.Write a program the given year is 1996, a leap year.
Q11. Create a list in python using the followings: 2,3,4,5,6,7 with variable ‘a’

Add ‘mango to the above list

Also add banana, grapes & orange in the list
insert apple in the 5th position of a variable ‘a’

Remove last item from the list
Q12.

L = [1,2,3,4,5,6,7]

Using the above list slice from 1:4

Q13. Reverse the order of given string L = [4,5,6,8,3] Without using reverse() function.

Q14. Use list comprehension to square the given list L=[2,4,7,3,6,8]

Q15. Create a function that takes in a tuple of integers and returns the sum of the integers. Test the

function with a tuple of your choice.
Q16. Create two sets of your favorite fruits, and use the union() method to combine them into a single set.
Print the resulting set to the console.

Q17. Create a set of random words, and use the add() method to add a new word to the set. Print the
resulting set to the console.

Q18. Create a set of your favorite animals, and use the remove() method to remove one animal from the
set. Print the resulting set to the console.

Q19. favorite_books = {"1984", "To Kill a Mockingbird", "Pride and Prejudice"}

favorite_movies = ["The Shawshank Redemption", "The Godfather", "The Dark Knight"]

Use the zip() function to combine the book set and movie list into a list of tuples representing book/
movie pairs. Print the resulting list.

Q20. Write a Python program to find the difference between consecutive numbers in a list.
Solutions:
Q21. Create a dictionary called fruits with the following key-value pairs:

"apple": 0.75

"banana": 1.25

"orange": 0.90

Then, print out the price of a banana.

Q22. Create an empty dictionary called ages. Add the following key-value pairs to the dictionary:

"Alice": 30

"Bob": 25

"Charlie": 35

Then, print out the age of Charlie.

Q23. Write a function called word_count(text) that takes a string as input and returns a dictionary where
Each key is a word in the text and its value is the number of times that word appears in the text. For
for example, word_count("hello world hello") should return {"hello": 2, "world": 1}.

Q24. Create a dictionary called phone_book with the following key-value pairs:

"Alice": "555-1234"

"Bob": "555-5678"

"Charlie": "555-9012"

Then, prompt the user to enter a name and print out the corresponding phone number. If the name is not
in the phone book, print out a message saying that the name was not found.

Q25. Write a program that prompts the user to enter a number between 1 and 10. If the number is less than
5, print out "Too low!", otherwise print out "Too high!".

Q26. Write a program that prompts the user to enter a password. If the password is "password123", print
out "Access granted", otherwise print out "Access denied".

Q27. Write a program that prompts the user to enter a positive integer. Then, use a loop to print out all the
odd numbers from 1 to that integer.

Q28. Write a program that generates a random number between 1 and 100 and then prompts the user to
guess the number. If the user's guess is too low, print out "Too low!", if the guess is too high, print out "Too

high!", and if the guess is correct, print out "You win!".

Q29. Write a program that generates a random number between 1 and 10 and then prompts the user to
guess the number. The user has three attempts to guess the number. If the user guesses correctly within
three attempts, print out "You win!", otherwise print out "You lose!".

Q30. Write a program that prompts the user to enter their age and then prints out whether they are a child
(age 0-12), a teenager (age 13-19), an adult (age 20-59), or a senior (age 60+)

Q31. Create a class called "Person" with properties for "name", "age", and "gender". Create an object of this
class and print out its properties.

Q32. Write a Python class called "Rectangle" with attributes for "width" and "height". Implement methods
to calculate the area and perimeter of the rectangle. Create an instance of this class and use it to print out
the rectangle's area and perimeter.

Q33. Write a Python class called "BankAccount" with attributes for "balance" and "interest_rate".
Implement methods to deposit and withdraw money from the account, as well as to calculate the interest
earned on the balance based on the interest rate. Create an instance of this class and use it to test out the
implemented methods for deposit, withdrawal, and interest calculation.

Q34. Write a Python class called "Animal" with attributes for "name" and "species". Create a subclass of
"Animal" called "Dog" with additional attributes for "breed" and "owner". Implement both classes with
appropriate methods and constructors to initialize their properties. Create instances of both classes and
use them to print out the various properties of the animals and dogs, such as their names, species,
breeds, and owners.

Q35. Create a class called "Car" with properties for "make", "model", and "year". Create a subclass of "Car"
called "ElectricCar" with additional properties for "battery_size" and "range". Create objects of both
classes and print out their properties.

Q36. Create a class called "Student" with properties for "name" and "id". Create a subclass of "Student"
called "GraduateStudent" with additional properties for "advisor" and "research_area". Create objects of
both classes and print out their properties.

Q37. Create a class called "Shape" with methods to calculate the area and perimeter of the shape. Create
subclasses of "Shape" for "Rectangle", "Circle", and "Triangle" with their own methods for calculating area
and perimeter. Create objects of each class and print out their area and perimeter.

Q38. Create a class called "Employee" with properties for "name", "id", and "salary". Add methods to give the employee a raise and to calculate their annual salary. Create objects of this class and test out the
methods.

Q39. Create a class called "Book" with properties for "title", "author", and "publisher". Create a subclass of
"Book" is called "Ebook" with additional properties for "file_format" and "file_size". Create objects of both
classes and print out their properties.

Q40. Create a class called "Bank" with properties for "name" and "accounts". Add methods to create new
accounts, deposit and withdraw money from accounts, and to calculate the total balance of all accounts.
Create an object of this class and test out the methods.
Q41. Can you create a Python program that reads a text file and counts the number of words contained
within it? The program should be designed to read the file and break it down into individual words, using

spaces, punctuation marks, and other delimiters to separate the words. It should then count the number
of words found in the file and display this count as output. The program should be flexible enough to work
with different text files and should be able to handle a variety of formatting and punctuation styles.

Q42. Could you help me create a Python program that can find and display the longest word in a text file?
The program should be able to read any text file and separate its contents into individual words, taking
into account various delimiters like spaces, punctuation marks, and other characters. Then, it should
compare the length of each word and determine which one is the longest, and finally, print that word as
output. The program should be versatile enough to work with various text files and be able to handle
different formatting styles and punctuation.

Q43. Write a Python program to read a text file and print out the most frequent word(s) in the file.

Q44. How can you use Python to count the number of rows in a CSV file?

Q45. How can you use Python to calculate the average of a specific column in a CSV file?

Q46. Write a Python program to read a JSON file and print out the value of a specified key.

Q47. Can you provide a Python code snippet to write a list of strings into a text file where each string is
written on a new line?

Q48. Can you provide a Python code to read a binary file and display the data in hexadecimal format.

Q49. Can you write a Python code to read a Comma-Separated Values (CSV) file, apply a specific condition
to each row, and create a new CSV file that contains only the rows that satisfy the condition? For example,
if the CSV file contains information about products and their prices, you may want to create a new CSV
file that only includes the products that are within a certain price range. The program should be able to
read the CSV file, compare the values in each row to the specified condition, and write the rows that meet
the criteria to a new CSV file.

Q50. Write a Python program to read a text file, count the frequency of each word, and write the results to
a new text file.

Q51. Write a program that prints the first 10 even numbers using a for loop.

Q52. Write a program that takes a list of strings and prints out each string in reverse order using a for loop.

Q53. Write a program that prints the multiplication table of a given number using a for loop.

Q54. Write a program that takes a list of integers as input and returns the sum of all the numbers in the list
using a for loop.

Q55. Write a program that prompts the user for a positive integer and then prints out all the prime
numbers up to that number using a for loop.

Q56. Write a program that prompts the user to enter a password until the correct password is entered
using a while loop.

Q57. Write a program that takes a list of strings and prints out each string in reverse order using a while
loop.

Q58. Write a program that prompts the user to enter a positive integer and then prints out all the Fibonacci
numbers up to that number using a while loop.

Q59. Write a program that takes a list of integers as input and returns the product of all the numbers in the
list using a while loop.

Q60. Write a program that prompts the user to enter a positive integer and then prints out the factorial of
that number using a while loop.
Q61. Write a program that spawns two threads. One thread should print even numbers between 0 and 10,
and the other thread should print odd numbers between 1 and 9.

Q62. Write a program that generates a list of random numbers and sorts them using multithreading. Use
two threads to sort the first half and the second half of the list in parallel.

Q63. Write a program that simulates a bank account transaction. The program should create two threads,
one for a deposit and one for a withdrawal. The deposit thread should add
100 to the account,and the withdrawal thread should withdraw

50 from the account. Run the program for 10 iterations.

Q64. Write a Python program that creates a thread to print out the current date and time every 5 seconds.
The program should continue running until the user presses the 'q' key.

Q65. Write a Python program that creates two threads. Each thread should print out the numbers from 1 to
10. The two threads should run concurrently and print out the numbers in an interleaved fashion.

Q66. Write a Python program to create a shared variable between two threads and increment its value in
each thread.

Q67. Here's a Python program that creates a thread to count down from 5 to 0 and prints "Blastoff!" when
the count reaches 0:

Q68. Write a program that creates a thread to print the Fibonacci sequence up to a certain number n. The
main thread should prompt the user for the value of n.

Q69. Write a program that creates two threads to add and subtract numbers from a shared variable. The
shared variable should start at 0, and each thread should perform 10 iterations of adding or subtracting a
random integer between 1 and 10. The program should print the final value of the shared variable.

Q70. Write a program that creates a thread to calculate the sum of the numbers from 1 to 100 and prints the result.

Q71.Write a Python program to create two processes that print out the numbers from 1 to 10 simultaneously.

Q72.Write a Python program that creates four processes and computes the sum of the first 1000 integers using multiprocessing

Q73.Write a Python program that creates two processes and communicates between them using a Queue. The first process should send a list of numbers to the second process, which should calculate the sum of the numbers and send it back to the first process.

Q74.Write a program to find the maximum number in a list of lists using multiprocessing.

Q75.Write a Python program that uses the Pool class from the multiprocessing module to calculate the squares of a list of integers.

Q76.How do you create a new process using the multiprocessing module in Python?

Q77.How do you use a Pool to execute a function with multiple arguments in parallel?

Q78.Write a Python program to calculate the sum of squares of numbers in a list using multiprocessing.

Q79.Write a Python program that uses the multiprocessing module to calculate the sum of a large list of integers.

Q80.Write a Python program that uses the multiprocessing module to parallelize the computation of the factorial of a number.

Q81.Write a Python program that prompts the user for two integers and divides the first integer by the second integer. If the either variable are none integer then print the value error and second integer is zero, catch the ZeroDivisionError and print an error message to the user.

Q82.Write a Python program that prompts the user for a password and checks whether the password meets certain criteria (e.g., at least 8 characters, contains at least one uppercase letter, etc.). If the password does not meet the criteria, raise a custom exception called PasswordError with a custom error message.

Q83.Write a Python program that prompts the user for a number and calculates the square root of the number using the math.sqrt() function. If the number is negative, raise a ValueError with a custom error message.

Q84.Write a Python program that prompts the user for a number and calculates the square root of the number using the math.sqrt() function. If the number is negative, raise a ValueError with a custom error message.

Q85.Write a Python program that prompts the user for a list of integers and calculates the average of the list. If the list is empty, raise a ValueError with a custom error message.

Q86.Write a function that takes a string as input and returns the number of vowels in the string. If the input is not a string, raise a TypeError with a custom error message.

Q87.write a python program how to handle simple runtime error?

Q88.write a program how to handle multiple errors with one answer:# Program to handle multiple errors with one

Q89.write a program to print the reciprocal of even numbers note:we might want to run a certain block of code if the code block inside try runs without any errors. For these cases, you can use the optional else keyword with the try statement.

Q90.How to handle exceptions using the try, except, and finally statements write code

Q91. Write a Python program to create a MySQL database and a table.

Q92. Write a Python program to insert data into a MySQL table.

Q93. Write a Python program to create an index on a MySQL table.

Q94. Write a Python program to join two tables in MySQL.

Q95. Write a Python program to handle MySQL errors using exception handling.

Q96. Write a Python program to connect to a MongoDB database and insert data.

Q97. Write a Python program to update data in a MongoDB database.

Q98. Write a Python program to handle MongoDB errors using exception handling.

Q99. Write a Python program to query a MongoDB database using aggregation.

Q100. Write a Python program to perform a text search on a MongoDB database.
Q101. Create a Flask application that displays "Hello, World!" on the homepage.

Q102. Write a Flask route that takes a name parameter and returns "Hello, [name]!" as plain text.

Q103. Write a Flask route that takes a number parameter and returns the square of that number as plain
text.

Q104. Write a Flask route that displays a simple HTML form that asks for a name and returns "Hello,
[name]!" when submitted.

Q105. Write a Flask route that displays a list of names in an HTML unordered list.

Q106. Write a Flask route that displays a list of names in a table.

Q107. Write a Flask route that displays a list of names in a dropdown menu.

Q108. Write a Flask route that receives data through a POST request and returns the data in JSON format.

Q109. Write a Flask route that redirects the user to a different URL.