1. What is a lambda function in Python, and how does it differ from a regular function?

Ans – lambda function is the small anonymous function in python , it is used when you need a quick , one line function, you don’t need to use def or give it a name , however regular function start with def and I have a name , it can have a multiple lines.

1. Can a lambda function in Python have multiple arguments? If yes, how can you define and use them?

Ans – lambda function in python have a multiple arguments , we can use them by sperating the commas , the expression after : this must return a single value

1. How are lambda functions typically used in Python? Provide an example use case.

Ans – lambda function is typically used in python for short , quick and one time operations , specially when defininig a full def function feels unnecessary

1. What are the advantages and limitations of lambda functions compared to regular functions in Python?

Ans = in lambda function there is only one line function but in regular we can use a multiple lines allowed , in lambda function name is a optional but in function must have a name ,, lambda function return keyword not required , but in regular function required for output, lambda function is good for small cases , however regular function is better for logic complex .

1. Are lambda functions in Python able to access variables defined outside of their own scope?

Explain with an example.

Ans – yes lambda function in python is able to access variables ,

Ex -

1. Write a lambda function to calculate the square of a given number.

Ans - square = lambda x :x \*\*2

print(square(5))

25 #output

1. Create a lambda function to find the maximum value in a list of integers.

Ans – na I don’t know ans

1. Implement a lambda function to filter out all the even numbers from a list of integers.

Ans – even\_numbers = lambda list : list (filter (lambda x:x%2 == 0 , list))

Numbers = [1,2,3,4,5,6,7,8,9]

Print(even\_numbers(numbers))

1. Write a lambda function to sort a list of strings in ascending order based on the length of each string.

Ans – sort\_by\_length = lambda 1st: sorted(1st key = lambda x:len(x))

Words = [“apple”, “kiwiw” , “banana”]

Print(sort\_by\_length(words))

1. Create a lambda function that takes two lists as input and returns a new list containing the common elements between the two lists.

Ans - common\_elements = lambda list1, list2: [x for x in list1 if x in list2]

list\_a = [1, 2, 3, 4, 5]

list\_b = [3, 4, 5, 6, 7]

print(common\_elements(list\_a, list\_b)) # Output: [3, 4, 5]

1. Write a recursive function to calculate the factorial of a given positive integer.

Ans = def factoroial(n):

If n ==0 or n =1:

Return 1

Else:

Return n+ factorial(n-1)

Print(factorial(5))

1. Implement a recursive function to compute the nth Fibonacci number.

Ans – idk

1. Create a recursive function to find the sum of all the elements in a given list.

Ans – idk

1. Write a recursive function to determine whether a given string is a palindrome.

Idk

1. Implement a recursive function to find the greatest common divisor (GCD) of two positive integers.

idk