

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

ANS:- Lambda functions are single line functions with this function You can perform the same lines of code but in a smaller way. The major difference is that lambda function does not have to start with "def" keyword.

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

ANS:- A lambda function can take any number of arguments, but can only have one expression. You can use them by using lambda function

EG .

```
add = lambda x,y: x+y
print(add(6,10))
```

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

ANS:- Lambda functions are anonymous functions i.e. without name, for this function we just use lambda keyword. Typically this function is used only when we need a small line of expressions.

Eg.

```
sub = lambda x,y: x-y
print(sub(6,10))
```

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

ANS:- The biggest advantage is that you don't have to write multiple lines with indentation and all and the same code can be written in just 2 lines with the lambda function. They don't have a name and can have only single line expressions and it is difficult to read and understand.

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

Explain with an example.

ANS: yes it can access it.

```
a = 10
sum = lambda x: x+a
print(sum(2))
```

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

ANS:-

```
sqre = lambda x: x*x
print(sqre(45))
```

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

ANS:-

```
from functools import reduce
l = [1,22,34,23,56]
r = reduce(lambda x,y: x if x>y else y, l)
print(r)
```

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

```
List1 = [1,2,3,4,5,6,7]
print(List1)
even_no = list(filter(lambda x: x%2==0,List1))
print(even_no)
```

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

ANS:-

```
list1 = ["hello","hi","good morning","hey","morning"]
sort = sorted(list1, key = lambda x:len(x))
print(sort)
```

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

ANS:-

```
l = [1,22,34,23,56]
l1 = [22,34,1,44,78]
common = lambda l,l1: list(filter(lambda x:x in l,l1))
print(common(l,l1))
```

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

ANS:-

```
def recursive(n):
    if n == 1:
        return n
    else:
        return n*recursive(n-1)

number = int(input("enter your number ::"))
if number<=0:
    print("sorry invalid no.")
else:
    print("the Factorial of",number,"is",recursive(number))

recursive(6)
```

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

ANS:-

```
def fibonacci(n):
    if n <= 2:
        return n-1
    else:
```

```

        return fibonacci(n-1)+fibonacci(n-2)
n = int(input("enter the number::"))
print(fibonacci(n))

```

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

ANS:-

```

def getSum(n):
    if len(n)==0:
        return 0
    else:
        return n[0] + getSum(n[1:])
print(getSum([1, 3, 4, 7, 5]))

```

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

ANS:-

```

def isPalindrome(s, low, high):
    if low >= high:
        return True
    if s[low] != s[high]:
        return False
    return isPalindrome(s, low + 1, high - 1)
if __name__ == '__main__':
    s = 'XYBYBYX'

    if isPalindrome(s, 0, len(s) - 1):
        print('Palindrome')
    else:
        print('Not Palindrome')

```

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

ANS:-

```

def gcd(a, b):
    if a == b:
        return a
    elif a < b:
        return gcd(b, a)
    else:
        return gcd(b, a - b)
print(gcd(45,25))

```

