# Section A: Theory (10 Questions)

## 1. What is the difference between a list and a tuple in Python?

| List | Tuple |
|---|---|
| List can be defined by [] | Tuple is defined by () |
| List is slower due to dynamic size and mutablity | Tuple is faster due to immutability |
| It is suitable for collection of items may change | It is suitable for collection of items that may not change |
| List having more inbuilt functions | Tuple having less inbult functions |
| List consumme more memory due to mutablity | Tuple is consume less memory due to immutablity |

## 2. Explain the immutability of strings with an example.

String immutability defines once the string is created it can not be modified
e.g
str_example = "Hello"
str_example[0] = 'h'
TypeError:String object is not support item assignment
When you try to change H to h it gave an error due to the immutability of strings.

## 3. Define a set and list two of its key properties.

Set :

1)is unordered because of this you can extract elements randomly each time when u use set

2)it only allows unique elements when you want an distinct elements then also you can use set

List:

1)list is ordered when we need continues data then we can use list

2)list is immutable that if you want to filter your data and replace that filters in original list then you can use list

## 4. How does a dictionary store its elements, and why are keys required to be immutable?

Dictionary store the elements in form of key values paires. With help of key we can extract values

# 5. What is the purpose of conditional statements in Python? Provide an example.

when you want to perform operations based on a particular condition is true in that scenario you can use conditional statements

e.g
if you want get name of employees whose experience is more than 5 years to increase there salary you can use conditional statements

```
d={'lavanya':2,'sama':6,'mira':8}
for k in v.keys():
        if d[k]>5:
                print(k)
```

output: sama & mira
these 2 are having experience more than 5 years
this result was only achieved by conditional statements

# 6. Explain the difference between `for` loops and `while` loops with examples.

for loops are used when the number of iterations is known beforehand, such as when iterating over elements in a list or any iterable object.

```
numbers = [1, 2, 3, 4, 5]

print("Iterating over the list using a for loop:")
for num in numbers:
    print(num)
```

While loops are used when the number of iterations is not known beforehand, and the loop continues as long as a certain condition is true. The loop variable is typically initialized before the loop, and the condition is checked before each iteration.

```
n='y'
while(n!='n'):

    n=input("enter y if want to continue n if not: ")
    print('welcome')
print("Bye")
```

output:
enter y if want to continue n if not: y
welcome
enter y if want to continue n if not: y

welcome
enter y if want to continue n if not: n
welcome
Bye

# 7. Describe the use of nested loops with a real-world example.
When your data having more than one columns in that scenario you can use nested loops specially when your Working with multi-dimensional data you use nested loops
Real-world example:
Clock: The second hand repeats every 60 seconds and then the minute hand moves to the next minute.
Earth: Rotates itself as it also moves around the sun.

# 8. What are the advantages of using list comprehensions in Python?
Readability: List comprehensions make code easier to understand and maintain.
Performance: List comprehensions are faster and more efficient than traditional for loops.

# 9. How do you reverse a string in Python? Write the syntax for it.
With help of slicing you can reverse string in python
S=”lava”
S=S[::-1]—reversed string

# 10. Explain the concept of pattern programs and their significance in coding practice.
a best way to learn about nested loops and control structures, transforming basic code into visually stunning designs.

involve printing different patterns on the console, usually made up of stars, spaces, numbers, or other symbols

# Section B: Correct the Error (10 Questions)

```
# 1. #    Identify and fix the error.
# my_list = [1, 2, 3# error is syntaxErro
# print(my_list)

# ANS:
my_list=[1,2,3]
print(my_list)


# 2.#    Identify and fix the error.
tuple_example = (1, 2, 3)
# tuple_example[1] = 5# TypeError tuple not support item assinment

# ANS:
tuple_example=list(tuple_example)
tuple_example[1]=5
print(tuple_example)


# 3. What will the output be, and why? Fix if needed
my_set = {1, 2, 3, 3}
print(len(my_set))

# ANS:
#No Erro outpu will be length od set which means how many elements a set have

# 4.Identify and fix the error.
my_dict = {1: 'a', 2: 'b'}
# print(my_dict[3])#keyError

# ANS:
print(my_dict[2])







# 5.Identify and fix the error.
# if 5 > 3#Syntax Error
   #    print("Five is greater than three")
```

```
# ANS:
if 5 > 3:
    print("Five is greater than three")

# 6. Identify and fix the error.
# for i in range[5]:#TypeError Range is not subscriptable
  #    print(i)

# ANS:
for i in range(5):
    print(i)


# 7.  What is the potential issue with this code? Fix it.
# while x < 5:# NameError name x is not defined
#       print("Hello")

# ANS:
x=1
while x<5:
    print('Hello')
    x+=1


# 8. Identify and fix the indentation error.
# nested_list = [[1, 2], [3, 4]]
# for i in nested_list:
# print(i)

# ANS:
# IndentationError:expected an indeted block after for
nested_list = [[1, 2], [3, 4]]
for i in nested_list:
 print(i)



# 9.   Identify and fix the error.
# str_example = "Hello"
# str_example[0] = 'h'
```

```
# print(str_example)

# ANS:
# TypeError:String object is not support item assignment
str_example = "Hello"
str_example='h'+str_example[1:]
print(str_example)


# 10.#    Fix the alignment issues in the nested loop.
#    for i in range(1, 6):
#        for j in range(i):
#        print('*', end=' ')
#        print()

# ANS:
for i in range(1, 6):
    for j in range(i):
        print('*', end=' ')
    print()
```

# Section C: Coding (10 Questions)

```
# 1. Write a program to reverse a given string.
```

```
s=input('Enter String: ')
r=''
for i in s:
    r=i+r
print(r)

# output:
# Enter String: lavanya
# aynaval


# 2. Create a list of numbers and find their sum using a loop.
l=list(map(int,input("Enter elements: ").split(" ")))
sum=0
for i in l:
    sum+=i
print(sum)

# output;
# Enter elements: 10 20 30 40
# 100

# 3. Write a Python program to generate the Fibonacci sequence up to `n` terms.
n=int(input("Enter Number: "))
n1=0
n2=1
print(n1,n2,end=" ")
for i in range(n-2):
      c=n1+n2
      print(c,end=" ")
      n1=n2
      n2=c
 output:
Enter Number: 5
0 1 1 2 3
Enter Number: 7
0 1 1 2 3 5 8
# 4. Create a dictionary to store the names and marks of 5 students, then print all students scoring above 80.
d={'lavanya':85,'samarth':53,'saranya':90,'pritu':34}
l=[]
for k,v in d.items():
```

```
    if v>80:
        l.append((k,v))
print(l)
# output:
# [('lavanya', 85), ('saranya', 90)]


# 5. Write a program to find the maximum element in a list without using the `max()` function.
 l=[10,20,30,400,50,1,2,3,45]
max=0
for i in l:
    if i>max:
        max=i
print("Maximun: ",max)

# output:
# Maximun:  400

# 6. Generate a pattern using nested loops:
#    1
#    3 5
#    7 9 11
#    13 15 17 19
z=1
for i in range(4):
    for j in range(i+1):
        print(z,end=" ")
        z+=2
    print()

# output:
# 1
# 3 5
# 7 9 11
# 13 15 17 19
# 7. Write a Python program to find the factorial of a given number using a loop.
n=5
fact=1
for i in range(1,n+1):
    fact=fact*i
print(fact)
```

```
# output:
# 120

# 8. Create a set of integers and check if a user-provided number exists in the set.
s={1,2,3,4,10,20,30}
n=int(input("enter no. to check: "))
print(n in s)
# output:
# enter no. to check: 3
# True

# 9. Write a program to count the occurrences of each character in a string using a dictionary.
s='acodenera'
d={}
for i in s:
    d[i]=s.count(i)
print(d)
# output:
# {'a': 2, 'c': 1, 'o': 1, 'd': 1, 'e': 2, 'n': 1, 'r': 1}

# 10. Develop a program to calculate the sum of squares of the first `n` natural numbers using a loop.
n=int(input("Enter number: "))
sum=0
for i in range(n):
    sum+=i**2
print("sum=",sum)
# output:
# Enter number: 4
# sum= 14
```