PYTHON TECHNICAL QUESTIONS:

1. What are the common ***built-in data types*** in Python?

   Data types in Python:
   1, string – character string
   2, int – integers
   3, float – decimal numbers
   4, boolean – True/False
   5, list – list of values and variables
   6, tuples – immutable list of values and variables
   7, dict – key-value dictionary
   8, array – array of bits

2. What are ***lists and tuples?*** What is the key difference between the two?

   List and tuple are lists of values and variables. Tuples is impossible change. Lists possible change.

3. What is the use of ***self*** in Python?

   Self refer to the object. In the example self refer Trida.
   Example:

```
Class Trida:
        def fce(self, attrib):
        self.attrib = attrib
        return self.attrib
trida = Trida()
```

4. What is   ***init_?***

   __int__ is special method called at inicialize obejct. Passes values for object.

5. What is ***break, continue*** and ***pass*** in Python?

   Break – stop loop
   Continue – Does not process after the content of the cycle, but skips and continues its execution.
   Pass – continue in process

6. What is ***slicing*** in Python?

   Slicing give part of list, string or tuple.

7. What are ***decorators*** in Python?

   Decorator takes the decorated function and return a function.

8. What are ***Dict*** and ***List comprehensions?***

   List – A list of values or variables with automatically assigned index. First value has index of 0 and each other +1.
   Dict – A dictionary is a key-value relationship.

9. What is ***lambda*** in Python? Why is it used?

A lambda is a simple anonymous function with a single expression. Use it inside another function. Lambda x: fce.

10. How do you create a ***class*** in Python?

```
class T:
    def __init__(self, x, y) -> None: # initialized automatically when the class is called
        self._x = x
        self._y = y
    def method(self): # class function
        return (self._x + self._y)


our_class = T(2, 3) # class constructor
our_class.method() # called class method
```

11. Write python function which takes a variable number of arguments.

```
def function(*argument):
```

12. Write a program, which takes a sequence of numbers and check if all numbers are unique.

```
numbers = 4, 12, 1, 3, 5, 6, 3.15, 2.12, 2.12, 14 #tested numbers
for x in numbers: # iterates all number
    if numbers.count(x) != 1: #tested if number is only one time in numbers
        print("Number {0} is not unique.".format(x) )
        break #stop process if number is repeats
    else:
        print("Number {0} is unique.".format(x))
```

PYTHON TECHNICAL TEST:

1. What will be the result of executing the code:

```
lst = [1, 2, 3]

pri.nt(i.d(lst) == i.d(lst[::J]))

```

1. True
2. False
3. 1
4. 0

False

2. What will be the result of executing the code:

```
. def func(par):
    par = [1]

lst = [0]
func(lst)

pri.nt(func)
```

a) (1, O]

b) (0, 1]

c) [O]

d) [1] **- this**

3. What will be the result of executing the code:

```
. def add(nu 1, nu 2):
    return nu 1+5, nu 2+5

res= add(3, 4)
ori.nt(res)
```

a) 17

b) 8

c) (8, 9) **- this**

d) Error

4. What will be the result of executing the code:

```
varone = 10
varTwo = ++varone

ori.nt(f"varOne=fvarOne1. varTwo=fvarTwo1")
```

a) varOne=l0, varTwo=lO **-this**

b) varOne=ll, varTwo=lO

c) varOne=lO, varTwo=ll

d) varOne=ll, varTwo=ll

5. What will be the result of executing the code:

```
arr=[4,3,2,1]
arr[0], arr[arr[0]-1] = arr[arr[0]-1], arr[0]

pri.nt(arr)
```

a) (1,3,2,3]

b) (1,3,2,4]

c) (4,3,2,1] **-this**

d) (1,3,2,1]

PYTHON TECHNICAL TASKS:

! ! !Important using only standard libraries of python

Problem 1

This is a CLI application where you can build a program that intakes some words from the user and then generates a random password using those words.
This password must be strong (min. length 8 char and must contain all of the symbols)

Problem 2

Implement a program that will launch a specified process and periodically collect the following data:

- CPU usage(%);
- Memory usage (bytes);
- Number of open handles (for Windows OS) or file descriptors (for Linux OS).

Data collection should be performed all the time the process is running. Path to the executable file for the process and time interval between data collection iterations should be provided by user. Collected data should be stored on the disk. Format of stored data should support automated parsing to potentially allow, for example, csv, json.

Problem 3

This is a command-line application where you will design a contact book application that users can use to save and find contact details.
Write a mini programm which is called a contact book to save contact details, including name, address, phone number, and even email address.
The application should also allow users to update contact information, delete contacts, and list saved contacts.
The SQLite database is the ideal platform for saving contacts.