PYTHON TECHNICAL QUESTIONS:

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

Datové typy v Pythonu jsou:
1, string - řetězce znaků
2, int - celá čísla,
3, float - reálná čísla,
4, boolean - nese stavovou hodnotu True, False
5, pole bajtů - např. soubor jpg.
6. list seznam hodnot
7. N-tice neměnný seznam hodnot
8. slovníky seznam hodnot uspořádaných podle klíčů klíč = hodnota

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

List je seznam proměnných, který můžeme na rozdíl od tuples měnit.
Tuples - n-tice měnit nemůžeme.

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

Pomocí self odkazujeme na objekt.
Příklad:

```
class Trida:
    def funkce(self, atribut):
        self.atribut = atribut
        return self.atribut
trida = Trida()
print(trida.funkce("vrací hodnotu"))
```

Self odkazuje na třídu Trida.

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

__init__ je speciální metoda volaná při vytvoření, inicializaci objektu. Předává hodnoty atributů objektu.

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

Break - Při splnění podmínky ukončí průběh cyklu.
Continue - Při splnění podmínky nezpracuje následující obsah cyklu, ale pokračuje v provádění od začátku.
pass - Pokračuje v provádění smyčky bez ohledu na podmínku.

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

slicing je získávání vybrané části n-tice nebo seznamu.

```
seznam = [ "a", "b", "c", "d"]
seznam[slice(start, stop, krok)]
seznam[slice(0, 4, 2)]
```

Stejný výsledek lze získat i pomocí následující syntaxe.

seznam[0:4:2] - vrací hodnotu v rozsahu indexu 0 - 4 po kroku 2

seznam[start, stop, krok]

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

Dekorátor bere jako argument dekorovanou funkci a vrací zase funkci.

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

list - Seznam hodnot nebo proměnných s automaticky přiděleným indexem. První hodnota má index 0 každá další +1.
Dict - slovník je seznam hodnot nebo proměnných ve vztahu klíč - hodnota.

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

Lambda je jednoduchá anonymní funkce s jedním výrazem. Hodí se použít uvnitř jiné funkce.
lambda x: funkce

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

Vytvoření třídy:

```
class Trida:
    def __init__(self, x, y) -> None: # inicializuje se automaticky při volání třídy
        self._x = x
        self._y = y
    def metoda(self): # funkce třídy
        return (self._x + self._y)
trida = Trida(2, 3) # konstruktor třídy. Volá-inicializuje třídu.
trida.metoda() # volá funkci třídy.
```

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

Použijeme znak * před argument.
def fce(*argument):

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

```
cisla = 4, 12, 1, 3, 5, 6, 3.15, 2.12, 2.12, 14 #testovaný řetězec čísel
for x in cisla: # iteruje jednotlivá čísla řetězce
    if cisla.count(x) != 1: #testuje, zda se číslo nevyskytuje v řetětci vícekrát než jednou
        print("Číslo {0} se v řetězci opakuje!".format(x) )
        break #ukončí testování pokud se číslo v řetězci opakuje
    else:
        print("Číslo {0} je unikátní.".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

False

3. 1
4. 0

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] **- OK**

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) **- OK**
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 **-OK**
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] **-OK**
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.