Q1. Cron jobs equivalent in windows?
--> On Microsoft Windows, cron jobs are known as Scheduled Tasks. They can be added through the Windows Task Scheduler user interface, by using PowerShell or with help of schtasks.exe . Running a task at specific time or at recurring dates is one of the common administrative tasks on all operating systems.

Q2. What are hyperparameters?
--> A hyperparameter is **a parameter whose value is set before the machine learning process begins**.

Q3. How to write do..while in Python?
-->

```
secret_word = "python"
counter = 0

while True:
word = input("Enter the secret word: ").lower()
counter = counter + 1
if word == secret_word:
break
if word != secret_word and counter > 7:
break
```

Q4. what is Clean Code?
--> Code is clean if it can be understood easily – by everyone on the team. Clean code can be read and enhanced by a developer other than its original author. With understandability comes readability, changeability, extensibility and maintainability.
--> https://gist.github.com/wojteklu/73c6914cc446146b8b533c0988cf8d29

Q5. Mention 3 sets in C++.
-->

Q6. Sets in memory in Python different from C++, How they stored in memory?
--> Hashing, a popular technique to perform insertion, deletion and traversal in O(1) on average. The operations on Hash Table are somewhat similar to Linked List. Sets in python are **unordered list with duplicate elements removed**.
-->

Q7. How to print Error in Python?

Built-in Python Exceptions

Here is the list of default Python exceptions with descriptions:

1. **AssertionError:** raised when the assert statement fails.
2. **EOFError:** raised when the input() function meets the end-of-file condition.
3. **AttributeError:** raised when the attribute assignment or reference fails.
4. **TabError:** raised when the indentations consist of inconsistent tabs or spaces.
5. **ImportError:** raised when importing the module fails.
6. **IndexError:** occurs when the index of a sequence is out of range
7. **KeyboardInterrupt:** raised when the user inputs interrupt keys (Ctrl + C or Delete).
8. **RuntimeError:** occurs when an error does not fall into any category.
9. **NameError:** raised when a variable is not found in the local or global scope.
10. **MemoryError:** raised when programs run out of memory.
11. **ValueError:** occurs when the operation or function receives an argument with the right type but the wrong value.
12. **ZeroDivisionError:** raised when you divide a value or variable with zero.
13. **SyntaxError:** raised by the parser when the Python syntax is wrong.
14. **IndentationError:** occurs when there is a wrong indentation.
15. **SystemError:** raised when the interpreter detects an internal error.

Example.

```
try:
print(1/0)
except ZeroDivisionError:
print("You cannot divide a value with zero")
except:
print("Something else went wrong")
```

the output will be

```
You cannot divide a value with zero
```

link.. https://www.datacamp.com/tutorial/exception-handling-python