Mohammad ElKholy 900202159

Q1) Integer data type: integers are stored in two's complement form in the machine, which the programmer is not aware of how it is implemented.

String data type: strings are stored as a series of characters. Each character being encoded in a certain format, example: Unicode, utf-8, etc which is hidden from the programmer.

Q2) for-loop: In assembly we have to write instructions for incrementing the loop counter & checking loop condition, whereas in a high level language that is done implicitly. Example: In python we can just specify the number of times we want to loop:

```
for i in range(1, 10):
```

If-statements: In assembly we need to specify the address we want to branch to if the if condition is met whereas in a high level language we just need to specify the condition. Additionally

if the condition is complex (multiple conditions) we need to specify more instructions for those.

Q3) An abstract data type is a logical description of the data and the operations that manipulate the data inside it. A data structure is the actual implementation of the abstract data type.

Example #1

List (ADT): A finite number of ordered elements which may or may not contain duplicates.

Array (DS): A list of elements stored consecutively in memory

Linked-list: A list of nodes, where each contains an element and a pointer to the next node

Example #2:
Map (ADT): A collection of records, where each record has a key to access it.

unordered_map & map (DS in C++): Both are implementations of the map ADT, but unordered-map is implemented using a hash table whereas map is implemented using a self balancing BST and both have different time complexity.

Q4) Python:

```
def factorial(n):
    if(n==0): return 1
    return n * factorial(n-1)
```

Q5)

Compilers	Interpreters	Pseudo Interpreters
1. Code scanned in one go	Code read line by line	code scanned in one go
2. executed on processor	executed on virtual machine	executed on virtual machine
3. errors shown in one go	errors are shown once line is translated	errors shown in one go
4. compiled into machine language	compiled into byte code	compiled into intermediate language
5. Very fast	Slow	intermediate
6. Performs lexical analysis, syntax analysis & semantic Analysis	Performs lexical analysis, syntax analysis & semantic Analysis	Performs lexical analysis, syntax analysis & semantic Analysis

Q6) I downloaded Cling which is an interpreter built on top of LLVM & Clang libraries. It comes with installing ROOT which is a data analysis framework provided by CERN. I found it through stackoverflow. I installed it by downloading the latest .exe installer for my machine. The interpreter reads line by line.

Experience using it: runs mostly like any python interpreter. I do not have to use semi-colons at the end of each statement. I also do not have to specify the variable type when assigning a value to it. Errors do not stop program execution and you can continue prototyping similar to a jupyter notebook.