E) Find 3 differences between a compiler and an interpreter.

BASIS FOR COMPARISON	COMPILER	INTERPRETER
Input	It takes an entire program at a time.	It takes a single line of code or instruction at a time.
Output	It generates intermediate object code.	It does not produce any intermediate object code.
Working mechanism	The compilation is done before execution.	Compilation and execution take place simultaneously.

F) Find the difference between Python 2 and 3?

Python 2

print functional brackets optional.
Prefix string with u to make unicode string.
Division of integers always return integer – 5/2=2.
Raw_input () reads string.
input() evaluates data read.
generator .next().

Python 3

print functional brackets compulsory.
String unicode by default.
Division of integers may result in float – 5/2=2.5.
Raw_input() not available.
Input always reads string.
Next (generator).
Py2 to py3 utility.
Dictionary .keys() and .values() returns a view not a list.
Can no longer use comparison operators on non natural comparisons.
Eg. None < None will raise a TypeError instead of returning false.
Percent (%) string formatting operator is deprecated use the .format()
Function or concatenation.

G) What is ASCII and UTF-8?

ASCII
In ASCII, every letter, digits, and symbols that mattered (a-z, A-Z, 0–9, +, -, /, “, ! etc.) were represented as a number between 32 and 127.

ASCII uses 7 bits to represent a character. By using 7 bits, we can have a maximum of 2^7 (= 128) distinct combinations. Which means that we can represent 128 characters maximum.

UTF-8
In UTF-8, every code-point from 0–127 is stored in a single byte. Code points above 128 are stored using 2, 3, and in fact, up to 6 bytes.

We would have needed an entirely new character set... that's the rational behind Unicode. Unicode doesn't contain every character from every language, but it sure contains a gigantic amount of characters