

### Question 1

- a. Machine language, or machine code, is the fundamental language of the 1<sup>st</sup> generation computers. Hence, it's also denoted to be the 1<sup>st</sup> generation programming language by some. Machine language is categorized as lower level language. Basically, it's a set of coded instructions in a sequence of binary bits comprised of only "0" and "1", usable and understandable only by the machines.
- b. Like machine language, assembly language or assembly code is categorized as a type of lower level language which uses mnemonic to signify each operation code (opcode). As such, the difference between machine language and assembly language is the latter uses symbolic equivalent as substitution to binary numbers, allowing codes more readable by humans. Still, it's a difficult language and tie to specific machine.
- c. High level language made programming simpler since it uses more natural and ordinary words as well as mathematics notation in which, contrary to binary or symbolic equivalent as in machine and assembly languages. Eventually, this coding method required a complier to translate them into machine language before the machine able to understand them.

**Commented [U1]:** Very good flow of writing but need to provide examples for each language and quote the sources

**Question 2:**

- i. **Edit:** The very first step where program was created by the programmer in the editor, follow by storing it to the storage.
- ii. **Preprocess:** Preprocess is the stage where preliminary operations or codes were performed prior the compiler able to process it. Output from preprocess stage remain as text form.
- iii. **Compile:** Compilation is a process converts source code (.c) into object code (.o) and stores it to the storage.
- iv. **Link:** Link or linking is referring to generation of an executable file from multiple object files. Link process is done by linker whichlinking the object code with the libraries.
- v. **Load:** Load process is done by loader which places program into the memory.
- vi. **Execute:** CPU processes each instruction and executes it. There is possibility of storing new values or data when the program executes.

**Commented [U2]:** You need to elaborate more for each step as the mark weighting for this question is high.

**Question 3:** Please refer to Codes folder for source code and executable file.

**Commented [U3]:** Very good answer but need to provide comments of code

**Question 4:** Please refer to Codes folder for source code and executable file.

**Commented [U4]:** Good use of function

**Question 5:** Please refer to Codes folder for source code and executable file.

**Commented [U5]:** Good answer but need to write in function

**Question 6:** Please refer to Codes folder for source code and executable file.

(a)

`!(3 + 3 >= 6)`

`!((3 + 3) >= 6)`

`!(6 >= 6)`

`!(TRUE)`

Answer: **FALSE**

(b)

`1 + 6 == 7 || 3 + 2 == 1`

`(1 + 6) == 7 || (3 + 2) == 1`

`7 == 7 || 5 == 1`

`(TRUE) || (FALSE)`

Answer: **TRUE**

(c)

`1 > 5 || 6 < 50 && 2 < 5`

`(1 > 5) || (6 < 50) && (2 < 5)`

`FALSE || TRUE && TRUE`

`FALSE || TRUE`

Answer: **TRUE**

(d)

`14 != 55 && !(13 < 29) || 31 > 52``(14 != 55) && !(13 < 29) || (31 > 52)``TRUE && !(TRUE) || FALSE``TRUE && FALSE || FALSE``FALSE || FALSE`*Answer:* **FALSE**

(e)

`6 < 7 > 5``((6 < 7) > 5)``((TRUE) > 5)`*Answer:* **FALSE****Commented [U6]:** perfect**Question 7:** *Please refer to Codes folder for source code and executable file.***Commented [U7]:** Very good answer**Question 8:** *Please refer to Codes folder for source code and executable file.***Commented [U8]:** Good answer, can simplify the logic further**Commented [U9]:** Excellent! You scored quite well in all the programming questions. It shows that you have very solid programming knowledge in basic C language as well as function. You should provide comments of code. You should have inline text referencing and the referencing format please follow WOU's student guide book. Keep up your good works.