Assignment 1 (Computing Foundations)

Q 1.

a)

- A CPU structure consists of registers, the ALU, and the control unit. The function of a CPU is to perform operations specified by the program.
- The memory of a computer, or otherwise known as the RAM, is connected to the CPU so the information is accessible.
- The I/O refers to anything that transfers data to and from the computer, like a keyboard or printer. The function of the I/O is to transfer information from the user to the computer and to output data for the user to access.
- The ALU is used to perform arithmetic and logic operations in the CPU.
- Each of these components needs each other to fully execute tasks. The I/O gets a command from the user and it goes through the CPU which accesses the memory. And the ALU which is a very important subsystem in the CPU will execute operations that are needed.

b) The main difference between the Von Neumann Architecture and the human brain is that the Von Neumann Architecture for a computer relies on the instruction given by the user and executes the problem with an already set of rules that it follows, it cannot come up with new rules. The processing speed rarely differs each time. A brain is vastly different. It does not follow the order in which the Von Neumann Architecture follows to execute commands. A brain can come up with its own instructions and can act on them in different amounts of time. A brain can also pull up memories if triggered by a word, it happens automatically even though pulling up memory is not needed. A computer cannot go look at memories for no reason. Brains and computers do have quite a bit in similarities, like the general way of executing a command, the way they get the instructions and do the command are noticeably different.

Q 2. The computer uses binary numbers because it is much easier to handle 2 states rather than 9 to 10 states if it were to use decimals. 1 and 0 can be translated with the electricity being turned on and off, and it could easier do true and false statements. If the computer were to use decimals which would need about 10 or so states, it would take much longer, would take more resources to create and it would be much harder in every aspect compared to using binary numbers.

Q 3.

A && C || !(B && C) && C

Inputs			***Outputs***			
A	**B**	**C**	**A && C**	**!(B && C)**	**A && C || !(B && C)**	**A && C || !(B && C) && C**
0	0	0	0	1	1	0
0	0	1	0	1	1	1
0	1	0	0	1	1	0
0	1	1	0	0	0	0
1	0	0	0	1	1	0
1	0	1	1	1	1	1
1	1	0	0	1	1	0
1	1	1	1	0	1	1

C || A && !(A) || B && C

Inputs			*Outputs*			
A	**B**	**C**	**!(A)**	**A && !(A)**	**B && C**	**C || A && !(A) || B && C**
0	0	0	1	0	0	0
0	0	1	1	0	0	1
0	1	0	1	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Is P && (Q || R) the same as (P && Q) || (P && R)?

Inputs			*Outputs (LHS)*		*Outputs (RHS)*		
P	**Q**	**R**	**(Q || R)**	**P && (Q || R)**	**(P && Q)**	**(P && R)**	**(P && Q) || (P && R)**
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

P && (Q || R) is the same as (P && Q) || (P && R)

Q 4. *Convert 52 to binary*

52/2 = 26 R 0
26/2 = 13 R 0
13/2 = 6 R 1
6/2 = 3 R 0
3/2 = 1 R 1
1/2 = 0 R 1

52 = 110100

Another way that I like converting decimals to binary is to minus the number by the closest base number which has a lower value than the decimal

128, 64, 32, 16, 8, 4, 2, 1

52 is closer to 32 so we minus it by that

52 - 32 = 20
20 - 16 = 4
4 - 4 = 0

So we only used 32, 16, and 4 we mark these numbers with 1s and the rest with 0s

32, 16, 8, 4, 2, 1
1   1 0 1 0 0 -> 110100

To double-check we can reverse the binary to decimal
We list down the base numbers and see which ones are labeled with 1

1 1 0 1 0 0
32 16 8 4 2 1

And we see that 31, 16, and 4 have the one. So now we add the three numbers together and if it adds to 52 to we are correct

32 + 16 + 4 = <u>52</u> <------ so now we can see that we were right