1. (25 points) The following table, represents the value of some numbers in decimal, binary and hex. Fill in the empty cells with the missed values. The detail of your conversion is not required.

| Decimal | Binary | Hex |
|---|---|---|
| 243 | 11110011 | 0xF3 |
| 172 | 10101100 | 0xAC |
| 175 | 10101111 | 0xAF |
| 123 | 01111011 | 0x7B |
| 64 | 1000000 | 0x40 |
| 169 | 10101001 | 0xA9 |

2. (20 points) Using the approximation of 1024 as 1 Kilo, how much are each of the following numbers?

a. $2^{34}$ **~16,000,000,000 (16 Giga)**
b. $2^{21}$ **~2,000,000 (2 Mega)**
c. $2^{17}$ **~128,000 (128 Kilo)**
d. $2^{46}$ **~64,000,000,000,000 (64 Tera)**

3. (5 points) L1 cache is …… and ….. than main memory.

**a.** faster and cheaper
**b.** **faster and more expensive**
**c.** slower and cheaper
**d.** slower and more expensive

4. (5 points) Main Memory is cache for …….

**a.** Main memory
**b.** **On chip caches**
**c.** Secondary storages
**d.** Register file

5. (15 points) Briefly (in one or two sentences), describe each one.

a. Register File - **They are storage in the CPU that hold data which the ALU performs operations on.**
b. PC – **Keeps track of which commands to execute, and executes them.**
c. Heap – **Responsible for balancing the memory management of different programs that are running; dynamically allocated.**

6. (20 points) In Figure 1.5, 1.6 and 1.7 (from B&O Book), describe the operations in which Main Memory and Disk are involved.

**1.5**: **The Main memory holds each character when "./hello" is written to it by the processor. The disk is not involved.**

**1.6: The disk sends out the executable for "hello" and it is written into the main memory.**

**1.7: The main memory sends out the "hello world/n" character from the main memory to the processor, which prints it out to the display. The disk is not involved.**

For questions 1-6, submit only one PDF file on BlackBoard.

7. (20 points) You learned *getchar()* and *EOF* in the class. You use them in an example to count the number of digits from input[1]. Extend[2] that C program, so it will count the number of various individual English alphabets (both lowercase and uppercase). Also, count the total number of characters.

You program should prints these:

- count of each individual digit (0-9)
- count of each individual alphabet letter (a-z and A-Z)
- total count of any kind of characters

So basically the output will be the values stored in (10+26+26+1=) 63 counter.

**Optional** - Count the number of words (i.e., terms that are separated by whitespaces: space, tab, newline). Note that there could be more than one whitespaces between two terms.

Submit only your C program on BlackBoard.

---

[1] As we showed in the class, the *input* can be received from a keyboard and terminated with *Ctrl-D*
or the *input* can be a file and send to the program using < operator.

[2] only use *getchar()* to read the characters from input. Don't use other functions for reading the input.