**LAB 4**

1.A) (111)₂ = (1 × 2²) + (1 × 2¹) + (1 × 2⁰) = (7)₁₀

= (7)₁₀

1.B) 777 = (7 × 8²) + (7 × 8¹) + (7 × 8⁰) = (511)₁₀

= (511)₁₀

1.C) (FEC)₁₆ = (15 × 16²) + (14 × 16¹) + (12 × 16⁰) = (4076)₁₀

= (4076)₁₀

2.A) Convert each hex digit to 4 binary digits and then convert each 3 binary digits to octal digits:

A9

= A 9

= 1010 1001

= 10 101 001

= 251(base8)

2.B) Convert each hex digit to 4 binary digits and then convert each 3 binary digits to octal digits:

E7

= E 7

= 1110 0111

= 11 100 111

= 347(base8)

2.C) Convert each hex digit to 4 binary digits and then convert each 3 binary digits to octal digits:

6E

= 6 E

= 110 1110

= 1 101 110

= 156(base8)

3.A) Convert every octal digit to 3 binary digits, then convert every 4 binary digits to 1 hex digit:

777

= 7 7 7

= 111 111 111

= 1 1111 1111

= 1FF

= 1FF(base16)

3.B) Convert every octal digit to 3 binary digits, then convert every 4 binary digits to 1 hex digit:

605

= 6 0 5

= 110 000 101

= 1 1000 0101

= 1 8 5

= 185(base16)

3.C) Convert every octal digit to 3 binary digits, then convert every 4 binary digits to 1 hex digit:

1

= (1) 16

4.A) Divide by the base 16 to get the digits from the remainders:

| **Division by 16** | **Quotient** | **Remainder**  **(Digit)** | **Digit #** |
| --- | --- | --- | --- |
| (1066)/16 | 66 | 10 | 0 |
| (66)/16 | 4 | 2 | 1 |
| (4)/16 | 0 | 4 | 2 |

= (42A)16

4.B) Divide by the base 16 to get the digits from the remainders:

| **Division by 16** | **Quotient** | **Remainder**  **(Digit)** | **Digit #** |
| --- | --- | --- | --- |
| (1939)/16 | 121 | 3 | 0 |
| (121)/16 | 7 | 9 | 1 |
| (7)/16 | 0 | 7 | 2 |

= (793)16

4.C) Divide by the base 16 to get the digits from the remainders:

| **Division by 16** | **Quotient** | **Remainder**  **(Digit)** | **Digit #** |
| --- | --- | --- | --- |
| (1)/16 | 0 | 1 | 0 |

= (1)16

5. Addition of two 8-bit numbers

MVI A, 06H ;Add value 06H to register A.

MVI B, 05H ;Add value 05H to register B.

ADD B ;Add value A and value B.

STA 3000H ;Store the final value to address 3000H.

HLT ;Halt the process.

**CAT2**

1.) (Instruction/Code-cache and Data-cache) in a processor?

Answer.

Cache Memory stores program instructions and data that are used repeatedly in the operation of programs or information that the CPU is likely to need next. We need it to improve the efficiency and speed of data retrieval by the CPU.

The split design enables us to place the instruction cache close to the instruction fetch unit and the data cache close to the memory unit, thereby simultaneously reducing the latencies of both

2.A.) Computer architecture focused on improving performance of the CPU

I would improve the overall the performance pf the CPU by adding specialized chips rather than having only general-purpose cores and memory. The specialized chips are as follows;

* Central processing unit (CPU) — the “brains” of the SoC. Runs most of the code of the operating system and your apps.
* Graphics processing unit (GPU) — handles graphics-related tasks, such as visualizing an app’s user interface and 2D/3D gaming.
* Image processing unit (ISP) — can be used to speed up common tasks done by image processing applications.
* Digital signal processor (DSP) — handles more mathematically intensive functions than a CPU. Includes decompressing music files.
* Neural processing unit (NPU) — used in high-end smartphones to accelerate machine learning (A.I.) tasks. These include voice recognition and camera processing.
* Video encoder/decoder — handles the power-efficient conversion of video files and formats.
* Secure Enclave — encryption, authentication, and security.
* Unified memory — allows the CPU, GPU, and other cores to quickly exchange information.

B.) The evolution of computers based on organization, architecture, and technology.

Transistors made of semiconductors replaced vacuum tubes in the construction of computers. By replacing this bulky and unreliable vacuum tubes with transistors, computers could now perform the same functions, using less power and space.

This was then followed by the creation of Integrated Circuits. The concept was to take the whole circuit, with all its elements and the relations between them, and regenerate the entire thing in an imperceptible tiny structure on the area of an element of silicon. The number of transistors on a chip currently is 114 billion. (On the Apple M1 ultra.)

Recently the invention of integrated SoC (System on chip) where all the pieces talk to each other quickly and efficiently and where the controllers and processes between them are built inside.

3."A is n times faster than B"  
PerformanceA / PerformanceB = n

.Computer A runs a program in 7 microseconds

.Computer B runs the same program in 11 microseconds

.How many times faster is Computer A?

11/7= 1.571

A is 1.571 times faster than B.

4.

**Answer:**  
Easiest is to convert to binary: 0011 1111.

Then convert to decimal: 32 + 16 + 8 + 4 + 2 + 1 = 63, which is the  
answer.

5. The biggest difference is cost. A mobile device must be cheap device(Enough for many users to acquire) while a server is less cost sensitive. Another difference is reliability. The bank demands high reliability if the server breaks(or silently calculates wrongly) millions of Kenyan shilling are at stake while a mobile device doesn’t have to be that reliable. The last difference is energy efficiency, efficiency is more important for battery life of the mobile device while a single server doesn’t require much efficiency but if it were a data center efficiency it would’ve been of great importance.

6. A process is basically a program in execution. The execution of a process must progress in a sequential fashion.

**1.Process ID**

When a process is created, a unique id is assigned to the process which is used for unique identification of the process in the system.

**2.Program counter**

A program counter stores the address of the last instruction of the process on which the process was suspended. The CPU uses this address when the execution of this process is resumed.

**3.Process State**

The process, from its creation to the completion, goes through various states which are new, ready, running and waiting. We will discuss about them later in detail.

**4.Priority**

Every process has its own priority. The process with the highest priority among the processes gets the CPU first. This is also stored on the process control block.

**5.General Purpose Registers**

Every process has its own set of registers which are used to hold the data which is generated during the execution of the process.

**6. List of open files**

During the Execution, every process uses some files which need to be present in the main memory. OS also maintain a list of open files in the PCB.

7. **Three types of process scheduling queues**

**1. Job queue -** This queue keeps all the processes in the system.

**2.Ready queue –** This queue keeps a set of all processes residing in main memory, ready and waiting to execute. A new process is always put in this queue.

**3.Device queue** – The processes which are blocked due to unavailability of an I/O device constitute this queue.

I/O

I/O Waiting queue

Ready queue

Job Queue

**Exit**