1) Discuss different types of addressing in MIPS ISA, show example of instruction use it

---

2)

Intel 8085 is a little endian microprocessor. what is the correct memory state after perform the following operation. copy the bytes 0xBBFF2211 to memory (Assume start memory address is 10000F)

Address	Content
10000F	

---

3)

Determine the following computer hardwire parameters to be designed:

1. Total number of instructions that can be implemented in the processor for (all types R, LS type and J)
2. Size of the register file (number of registers)
3. Maximum size of main memory (assume that a program can access data from any place in it)
4. How many bits are needed for the program counter (PC) and the instruction register (IR)

4) Show how in MIPS assembly to copy immediate value 0xFFFF7643 into register $s0

5) Translate this code into MIPS:

```
int sum = 0;

for(int i = 0; i < n; ++i)
    sum += A[i];
```

6) Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

```
f = g - A[B[4]];
```

a. For the C statement above, what is the corresponding MIPS assembly code?

b. For the C statement above, how many MIPS assembly instructions are needed?

c. For the C statement above, how many different registers are needed?

---

7) Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

```
lw $s0, 4($s6)
```

a. For the MIPS assembly instructions above, what is the corresponding C statement?

b. For the MIPS assembly instructions above, rewrite the assembly code to minimize the number of MIPS instructions (if possible) needed to carry out the same function.

c. How many registers are needed to carry out the MIPS assembly as written above? If you could rewrite the code above, what is the minimal number of registers needed?

---

8)

a. Implement the “branch if memory and register are equal” (bmre) instruction, which uses the i-type

Field	op = 0	rs	rt	offset
Bits	31-26	25-21	20-16	15-0

The bmre instruction has the following semantics:

```
if (R[rt] == M[R[rs]]) {
                        // PC = PC + 4 + 4*offset;
} else {
                        // PC = PC + 4
}
```

Show what changes are needed to support bmre instruction and write (next to the signal's name) values of all control signals for this instruction. You should only add wires, muxes, and a comparator (as shown above) to the datapath; do not modify the main functional units themselves (the memory, register file and ALU). Try to keep your diagram neat!