Question 1 (30 points): Provide a **brief** answer to each of the following questions.

1. (**5 points**) In a MIPS assembly program, given the instruction `beq $s3 $s4 Target`, the pseudo code for the execution of this instruction is:

```
PC = PC + 4
if ($s3 = $s4)
   PC = PC + signExtended(address << 2)
```

In this pseudo code `address` stands for the 16 least-significant bits of the instruction word.

During the execution of a MIPS program, a branch instruction is fetched from the address `0x8000 FABC`. The binary representation of this instruction is `0x1274 FFF0`. When this branch is taken, what is the address, expressed in hexadecimal, of the next instruction that is executed?

Target address in hexadecimal: `0x8000 FA80`

```
 signExtended(address << 2) = 0xFFFF FFC0
                       PC+4 = 0x8000 FAC0
   0x8000 FAC0
+  0xFFFF FFC0
---------------
   0x8000 FA80
```

2. (**5 points**) Given a jump instruction in a MIPS program, the target address for the jump is obtained by concatenating the four most significant bits of PC+4 with the 26 least-significant bits of the instruction word and then shifting the resulting 30 bits to the left by two.

During the execution of a MIPS program, a jump instruction is fetched from the address `0x4FFF FFFC`. Which is the lowest memory address and which is the highest memory address where the target of this jump instruction can be placed in memory. Express both addresses in hexadecimal.

Lowest address: `0x5000 0000` **Highest Address:** `0x5FFF FFFC`

3. (**5 points**) In the translation of a C program to MIPS assembly, assume that the base address for the integer arrays `A`, `B`, and `C` are already in registers `$s0`, `$s1` and `$s2`, respectively. The indexes variables `i` and `j` are in registers `$t0` and `$t1`, respectively. How many load and how many store instructions must be executed in MIPS assembly to execute the following C statement?

```
A[B[i+3]] = C[B[j+1]];
```

Number of loads: $\boxed{3}$ **Number of stores:** $\boxed{1}$

4. (**5 points**) The register `$s0` contains the binary representation of a branch instruction. Write a sequence of MIPS assembly code that produces in `$v0` the value that should be added to the PC register to determine the target address of the instruction. Your code should not change the value of any other register in the MIPS processor.

```
sll  $v0, $s0, 16
sra  $v0, $v0, 14
addi $v0, $v0, 4
```

5. (**5 points**) Steamroller is a third-generation of the family 15th microprocessor from AMD, which was released in 2014. The L2 cache of the Steamroller has a capacity of 2 M bytes, each cache block stores 64 bytes, and this L2 cache is 16-way set associative. A memory address in the Steamroller has 64 bits. The drawing below represents a memory address in this architecture and it has been divided into three fields: tag, index and offset. Inside the six small boxes drawn above the address, indicate which bits are used for tag, index, and offset.

63		17	16		6	5		0
tag			index			offset		

Offset: $2^6 = 64$ bytes ⇒ bits 5-0

Index: 16-way ⇒ each index selects 16 blocks ⇒ Set size $= 16 \times 64 = 2^4 \times 2^6 = 2^{10}$

Number of sets: $\frac{2MB}{2^{10}} = \frac{2 \times 2^{20}}{2^{10}} = 2^{11}$

Index bits: 16-6

6. (**5 points**) The following code is used to synchronize access to a memory location by multiple processors and to ensure that the increment of a counter is atomic. Assume that when this code starts executing `$s1 = 0x8000 4000`, and `$t1 = 0x0000 0008`.

```
ll      $t1, 0($s1)        # load linked

addi    $t1, $t1, 1        # increment counter

sc      $t1, 0($s1)        # store conditional
```

If another processor stores to the memory location whose address is `0x8000 4000` after the `ll` instruction is executed and before the instruction `sc` is executed, what will be the value in `$t1` after the execution of the `sc` instruction?

`$t1`: $\boxed{0}$