CSE340: Computer Architecture

Assignment 2
Chapter 2 (Instructions: Language of the Computer)

Question - 1:

We have 8 argument registers, X10 - X17 in RISC-V, Suppose you created a function that requires 10 parameters. Now how will you pass these 10 arguments to your function?

Question - 2:

Given a machine code, what do we understand by looking at the **Opcode** field?

Question - 3:

Suppose you have converted LD X9, 10[X21] to its equivalent RISC-V machine code. Now based on which **field** machine will understand the size of the data being loaded into register X9?

Question - 4:

What is a Program Counter? Explain the importance of a Program Counter with example(s).

Question - 5:

Given the machine code: `0110 1111 1101 1010 0110 0110 0110 0111`,

It is identified as an SB-type instruction based on its Opcode. To determine the offset from this machine code, one must first construct the 12-bit immediate value and then multiply this immediate value by 2.

Explain why you need to multiply the immediate value by 2 to obtain the offset value?

Question - 6:

BEQ X0, X0, Label; Why do we call this instruction an unconditional jump?

Question - 7:

```
Main () {
    int x = 0;
    int y = 9;
    int z = addition(x, y);
}

int addition (int a, int b) {
    int c = a + b;
    return c;
}
```

In the above code snippet, Main() is the **caller** function and addition is the **callee** function. When we **transfer control** from Caller(Main) to Callee(addition) we use Jal X1, label

Again when we **transfer control** from Callee(addition) to Caller(Main) we use Jalr X0, label.

In both cases we are transferring control, so why do we need two different instructions to do so?

Question - 8:

```
if ( A[3] != A[6]){
    if (A[3] == 0) {
        A[3] = A[3] + 2;
    }else{
        A[6] = A[6] / 16;
    }
}else{
    A[6] = A[6] * 8
}
```

a) Construct the equivalent RISC-V code of the above mentioned C code.

b) Base addresses of array A and B are in register X20 and X21.
From your converted codes, pick any instruction from each of the three formats(R, I, S) and convert them into their corresponding binary values.