1. A program executes in 10 sec on a particular computer. The computer's branch-processing subsystem is upgraded so that branch instructions execute twice as fast as before. After the upgrade, the original program runs in 8 sec. What fraction of the program's instructions is branch instructions?

From Amdahl's Law, \( 8 = 10 \left(1 - \frac{x}{2}\right) \implies x = 0.4 \)

2. You are evaluating two 'C' compilers, A and B, for your computer. You compile the same test program using each compiler. The compiled code from compiler A has an instruction count of 1000\( _{10} \) instructions, and average CPI of 2. The compiled code from compiler B has an instruction count of 1200\( _{10} \) instructions. When you run the two programs, you find that the CPU time for the two versions is equal. What is the average CPI for the code compiled with compiler B?

\( 1000 \times 2 = 1200 \times CPI_B \implies CPI_B = \frac{5}{3} \)

3. A data word is stored by a MIPS processor at memory address 1000\( _{10} \). What is the memory address of the least-significant byte of the data word?

MIPS is “big-endian” and memory words occupy 4 bytes so the least-significant byte is at address 1003

4. Write a MIPS code segment for the 'C' statement: 
\[ A = 5 + B[C] \]
Assume that A is in $s0, the base address of B is in $s7, and the address of the memory variable C is in $t3.

\[
\begin{align*}
\text{lw} & \quad \$t4, 0(\$t3) \quad \# \$t4 \leftarrow C \\
\text{sll} & \quad \$t4, \$t4, 2 \quad \# \$t4 \leftarrow 4 \times \$t4 \\
\text{add} & \quad \$t5, \$t4, \$s7 \quad \# \$t5 \text{ now contains the address of } B[C] \\
\text{lw} & \quad \$t6, 0(\$t5) \quad \# \$t6 \leftarrow B[C] \\
\text{addi} & \quad \$s0, \$t6, 5 \quad \# A = 5 + B[C]
\end{align*}
\]

5. Encode the MIPS instruction:
\[
\text{addi} \quad \$s1, \$s2, -17_{16}
\]
Show the contents of each field in binary.

\[\text{Note that you needed to convert the immediate data to 2's complement form}\]
\[
001000 \mid 00010 \mid 00001 \mid 1111 \quad 1111 \quad 1110 \quad 1001
\]

6. Encode the MIPS instruction:
\[
\text{lw} \quad \$s3, 65_{10}(\$t0)
\]
Show the contents of each field in hexadecimal.

\[\text{23 | 8 | 13 | 41 \ hex}\]

7. Encode the MIPS instruction:
\[
\text{bne} \quad \$s1, \$s2, \text{L1}
\]
where the label L1 refers to memory address 1000\( _{10} \) and the bne instruction is located at memory address 500\( _{10} \). Show the contents of each field in binary.

\[\text{The offset is } 1000 - (500 + 4) = 496_{10} \]
\[\text{So the answer is: } 00 \quad 0101 \mid 1 \quad 0001 \mid 1 \quad 0010 \mid 0000 \quad 0000 \quad 0111 \quad 1100\]
8. Encode the MIPS instruction:

    j L2

where the Jump instruction is located at address 12345678_{16} and the label L2 refers to address 76543210_{16}. Show the contents of each field in hexadecimal.

Target /4 = 01 1101 1001 0101 0000 1100 1000 0100
Dropping h/o 4 bits gives 01 1001 0101 0000 1100 1000 0100 = 1950C84 hex
So the answer: 2 | 1950C84 hex

Since the instruction attempts to jump outside of the current page of memory, this instruction might result in an assembler error, so I also accepted the answer that the jump could not be encoded.