1. Consider three different processors P1, P2, and P3 executing the same instruction set. And, their clock rate and CPI are like below:

	P1	P2	P3
CPI	4GHz	3.4GHz	2.4GHz
Clock Rate	2.5	2.0	1.2

a. Which processor has the highest performance expressed in instructions per second?

Answer 1:
P1= 4GHz / 2.5= 1.6* 10^9 instructions per second
P2= 3.4GHz/2.0= 1.7* 10^9 instructions per second
P3= 2.4 GHz/1.2= 2 * 10^9 instructions per second
P3 has the highest Performance.

b. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.

Answer 2:
Cycles:
P1= 4GHz * 10 = 4 * 10^10
P2= 3.4GHz *10 = 3.4 * 10^10
P3= 2.4GHz *10 = 2.4 * 10^10
Number of Instructions:
P1= (4GHz * 10) / 2.5 =1.6 * 10^10 instructions
P2= (3.4GHz * 10) / 2 =1.7 * 10^10 instructions
P3= (2.4GHz * 10) / 1.2 =2 * 10^10 instructions

c. We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

Answer 3:

Execution time= (No. of instruction * CPI) / Clock Rate
Execution time * 0.7 = (No. of instruction * CPI * 1.2) / Clock Rate
New Clock Rate = Clock Rate * 1.2 / 0.7
= 1.71 * Clock rate
P1= 4GHz * 1.71 = 6.8 GHz
P2= 3.4GHz * 1.71 = 5.814 GHz
P3= 2.4GHz * 1.71 = 4.104

2. In this problem, we compare the efficiency of four ISAs with respect to code compaction and to memory traffic:

- Accumulator-based
- Stack-based
- Memory-to-memory (all operands are located in main memory)

• Register-based (pure Load/Store)

1. Accumulator-based
   Load D
   Add B
   Store C
   Load A
   Sub C
   Store B

Code size = # instructions × size of each instruction
=6 * 4
=24

Data Memory Traffic= # data addresses given to the memory × data address size +
data received from the memory × data size
= 6 * 4 + 6 * 4
=48

Instruction traffic = # instruction addresses given to the memory × address size +
instruction received from the memory × instruction size
= 6 * 4 + 6 * 4
= 48

2. Stack-based
   Push D
   Push B
   Add
   Pop C
   Push A
   Push B
   Sub
   Pop B

Code size = # instructions × size of each instruction
=6 * 4 + 2 * 1
=26

Data Memory Traffic= # data addresses given to the memory × data address size +
data received from the memory × data size
= 6 * 4 + 6 * 4
=48

Instruction traffic = # instruction addresses given to the memory × address size +
instruction received from the memory × instruction size
= 8 * 4 + (6 * 4 + 2 * 1)
=58

3. Memory-to-Memory
   Add C, D, B

Sub B, A, C

Code size = # instructions × size of each instruction
=2 * 10
=20

Data Memory Traffic= # data addresses given to the memory × data address size +
data received from the memory × data size
= 6 * 4 + 6 * 4
=48

Instruction traffic = # instruction addresses given to the memory × address size +
instruction received from the memory × instruction size
= 2 * 4 + 2 * 10
= 28

4. Register Based
   Load R1, D
   Load R2, B
   Add R1, R1, R2
   Store C, R1
   Load R2, A
   Sub R2, R2, R1
   Store B, R2

Code size = # instructions × size of each instruction
=5 * 5 + 2 *3
= 31

Data Memory Traffic= # data addresses given to the memory × data address size +
data received from the memory × data size
= 5 * 4 + 5 * 4
=40

Instruction traffic = # instruction addresses given to the memory × address size +
instruction received from the memory × instruction size
= 7 * 4 + (5 * 5 + 2 * 3)
= 59

	Accumulator based	Stack based	Memory to memory	Register based
Code size	24	26	20	31
Data traffic	48	48	48	40
Instruction traffic	48	58	28	59