1) A 500-MHz processor was used to execute a program with the instruction mix

Instr. type	CPI	Instr. count
ALU operations	3	40000
Memory accesses	4	25000
Branches	2	15000
Floating-point	5	20000

(a) Determine the **average CPI**.

(b) Assume we build an optimizing compiler which discards 50% of ALU instruction. What's the **average CPI** of this improved version? **How much faster** is the improved version than the original one?

2) Suppose a program runs in 100 seconds on a machine, with floating-point operations responsible for 60 seconds of this time. How much do we have to improve the speed of floating-point operation if we want this program run two times faster? How about making it 2.5 times faster?

1.

(a) Average CPI = (3 * 40000 + 4 * 25000 + 2 * 15000 + 5 * 20000) / (40000 + 25000 + 15000 + 20000) = **3.5**

Execution time = 100000 * 3.5 / (500 * 10^6) = 0.0007

= 7 * 10^{-4} (sec)

(b) Average CPI = (3 * 20000 + 4 * 25000 + 2 * 15000 + 5 * 20000) / (20000 + 25000 + 15000 + 20000) = **3.625**

Execution time = 80000 * 3.625 / (500 * 10^6)

= 5.8 * 10^{-4} (sec)

Speedup = (7 * 10^{-4}) / (5.8 * 10^{-4}) = **1.207** (times)

The average CPI only cannot be used as performance metric. ∵ Only the execution time can be the performance metric.

2. (a) Speedup = 2 = 100 / (40 + 60/n)

$\Rightarrow$ ***n* = 6**

(b) Speedup = 2.5 = 100 / (40 + 60/n)

$\Rightarrow$ n = infinity

So It is **impossible** to make the program 2.5 times faster.

Amdahl's Law: The performance enhancement possible with a given improvement is limited by the amount that the improved feature is used.