1. Please explain the functionality of the *program counter (PC)* when we executing a program. Moreover, please explain how *PC-relative addressing* works. (15%)

2. Please spell out the full names of the *DLL*. What is the advantage of utilizing DLLs instead of static linking when generating an executable file? Why? (15%)

3. Suppose an identical program is complied using four different optimization methods (Opt1~Opt4) on a specific computer system with a single 400MHz CPU, respectively. Corresponding values of executing the respective program are recorded as shown in the following table.

Optimization method	Clock cycles (millions)	Instruction count (millions)
Opt1	158,615	114,938
Opt2	66,900	37,470
Opt3	66,521	39,993
Opt4	65,747	44,993

(A) Which optimization method generates the program with the highest CPI (clock cycles per instruction) value? (8%)
(B) Which optimization method generates the program with the highest MIPS (million instructions per second) value? (8%)
(C) Which optimization method generates the fastest program? How many seconds (in terms of CPU time) are required to execute this program? (8%)
(D) If your answers in (A) to (C) are identical, please explain the relationships among the three corresponding performance metrics. If not, please explain why the contradiction exists. (6%)

（背面仍有題目，請繼續作答）

4. IEEE 754 floating-point standard utilizes the following formula to represent a single-precision floating-point number:

$$(-1)^S \times (1+F) \times 2^{(E-127)}.$$

Please answer the following questions according to this standard.

(A) Given the following bit pattern, what is the value of this floating-point number? (10%)

S	E	F
1	01111110	10000000000000000000000

(B) Please show the bit pattern for the floating-point number $-5/6_{ten}$. (10%)

5. (A) Please explain what the *deadlock* is in terms of the relationships among *processes* and *resources*. (5%)

(B) Consider a system consisting of m resources of the same type, being shared by n processes. Resources can be requested and released by processes only one at a time. Show that the system is deadlock free if the following two conditions hold:

i. The maximum need of each process is between 1 and m resources.

ii. The sum of all maximum needs is less than $m + n$. (15%)