本試題是否可以使用計算機： ☐可使用 ， ☑不可使用 （請命題老師勾選）

1. Consider the following performance measurements for running a benchmark program on three different computers (each with only one CPU):

Measurement	Computer A	Computer B	Computer C
Instruction Count	4.5×10^{10}	4×10^{10}	4.2×10^{10}
Clock Rate	3.2 GHz	4 GHz	2.8 GHz
Cycles Per Instruction	1.6	2.0	1.4

(A) Please calculate the execution time in seconds for running this benchmark program on Computer A, B and C, respectively. (10%)
(B) Please calculate the MIPS (i.e., million instructions per second) rating for Computer A, B and C, respectively. (10%)

2. For transforming C programs into executable files, several utilities are required, including the *compiler*, the *assembler*, and the *linker*. Please explain the functionalities of these three utilities, respectively. (15%)

3. When we say that the Java language is portable, a major reason is that there are different software interpreters on many different platforms. What is the usual name of these interpreters? What is the common side effect (i.e., drawback) of utilizing interpreters for running Java programs? Why? (15%)

4. Please make a comparison between the static RAM and the dynamic RAM in terms of their major characteristics. Also, please spell out the full names of "RAM". (15%)

(背面仍有題目,請繼續作答)

本試題是否可以使用計算機： □可使用 ， ☑不可使用 （請命題老師勾選）

5. Assume that a processor has a five-stage pipeline (as shown below) with each stage taking one clock cycle to finish.

IF	ID	EXE	MEM	WB

IF: instruction fetch

ID: instruction decode

EXE: execution or address calculation

MEM: data memory access

WB: write back to register file

(A) Suppose that we have an add instruction followed immediately by a subtract instruction as follows:

```
add $2, $3, $4  // Register $2 = $3 + $4
sub $5, $2, $6  // Register $5 = $2 - $6
```

Please explain why this situation causes the pipeline stall. Also, how many clock cycles will this pipeline stall last if there is no *forwarding*? (15%)

(B) It is known that the data hazard in (A) can be resolved by *forwarding*. Please describe how *forwarding* works with above pipeline stages. (5%)

(C) Consider the similar situation in which a memory read occurs after a memory write:

```
sw  $7, 100($2)  // Store data from register $7 to memory
lw  $8, 100($2)  // Load data from memory to register $8
```

Write a short paragraph describing how this situation differs from the one in (A), and describe how the potential *read-after-write* problem is resolved. (10%)

6. What are the reasons for programmers at early years to write programs in assembly languages rather than in high-level languages? (5%)