請照題號次序作答
Use C, C++ or Java programming language to design your computer programs.

1. (20%) The following table is the list of instructions for a simple computer A.

Instruction	Code	Operands			Action
	4-bit	4-bit	4-bit	4-bit	
	d1	d2	d3	d4	
HALT	0	0	0	0	Stops the execution of the program
LOAD	1	R_D	M_S		$R_D \leftarrow M_S$
STORE	2	M_D		R_S	$M_D \leftarrow R_S$
ADDI	3	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1} + R_{S2}$
MULI	4	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1} \times R_{S2}$
MOVE	5	R_D	R_S	0	$R_D \leftarrow R_S$
NOT	6	R_D	R_S	0	$R_D \leftarrow \neg R_S$
AND	7	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1}$ AND R_{S2}
OR	8	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1}$ OR R_{S2}
XOR	9	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1}$ XOR R_{S2}
INC	A	R	0	0	$R \leftarrow R + 1$
DEC	B	R	0	0	$R \leftarrow R - 1$
ROTATE	C	R	n	0 or 1	Rot_n R (rotate n bits,)
JUMP	D	R	n		IF $R_0 \neq R$ then PC = n, otherwise continue

- d_1, d_2, d_3, d_4: First, second, third, and four hexadecimal digits.
- R_S, R_{S1}, R_{S2}: Hexadecimal address of source registers
- R_D: Hexadecimal address of destination register
- R_0 is always set to 0x0000.
- M_S: Hexadecimal address of source memory location
- M_D: Hexadecimal address of destination memory location
- n: Hexadecimal number or hexadecimal address

The right table is the memory image of this simple computer. The program was allocated in the memory address 00 ~ 09, and the CPU executes code from memory address 00.

Memory Address	Hex Code
00	1140
01	1241
02	1342
03	3423
04	5230
05	5340
06	B100
07	D103
08	2434
09	0000
...	
40	0009
41	0001
42	0001
43	0000

a. (5%) What the object of this program?
   (a) calculate the summation
   (b) calculate the factorial number.
   (c) calculate the Fibonacci number.

b. (10%) Using the instructions of this simple computer, write a program that can compute the result of "$1^2 + 2^2 + 3^2 + ... + 9^2$".

c. (5%) If the length of every data register (R_i) is 16 bits, can you get the correct answer in question (b)? Explain your answer.

2. (20%) Sorting:
   a. (5%) Which sorting algorithm is best for sorted array? Insertion sort, selection sort, bubble sort, quick sort or merge sort? Explain your answer.
   b. (10%) Given an array of integers, write a program which implement the algorithm you selection in question (a) to sort the array.
   c. (5%) Give an example array of 8 integers to explain the worst case of your program in question (b).

見背面

3. (10%) **ROT13** ("rotate by 13 places", sometimes hyphenated ROT-13) is a simple substitution cipher used in online forums as a means of hiding spoilers, punchlines, puzzle solutions, and offensive materials from the casual glance. Applying ROT13 to a piece of text merely requires examining its alphabetic characters and replacing each one by the letter 13 places further along in the alphabet, wrapping back to the beginning if necessary.[2] A becomes N, B becomes O, and so on up to M, which becomes Z, then the sequence continues at the beginning of the alphabet: N becomes A, O becomes B, and so on to Z, which becomes M.
Table:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyzabcdefghijklm
```

Write a program that can correctly transform the following text:

```
How can you tell an extrovert from an
introvert at NSA? Va gur ryringbef,
gur rkgebireg ybbxf ng gur BGURE thl'f fubrf.
```

into the cypher text:

```
Ubj pna lbh gryy na rkgebireg sebz na
vagebireg ng AFN? In the elevators,
the extrovert looks at the OTHER guy's shoes.
```

4. (15%) A multiprogramming operating system uses **paging**. The available memory is 4096KB, and each frame is 4KB. The first program needs 235KB. The second program needs 86KB. The third program needs 1833KB. The fourth program needs 347KB
   a. How many frames are unused?
   b. How much memory is wasted in internal fragmentation?
   c. If the system has only 1024KB, how large the minimum swap space is needed in hard disk to run all the programs?
   d. If a data locate in physical memory address 0x017728, what is the frame number of this address? (hint: the size of frame/page is 4KB, the physical memory address 0x000000 is the starting address of frame 0x000)
   e. One program contains two pages. After loading to the memory, the page 0x000 is mapped to frame 0x005, the page 0x001 mapped to frame 0x022. What is the physical memory address of the 4735th instruction (count from the beginning) in this program?

5. (20%) Terminology explanations: explain the terminologies marked in bold in detail.
   a. **Thrashing** of virtual memory.
   b. **Bootstrap process** of operation system
   c. **Linking process** after compiling source code
   d. **Pass by reference** in C++
   e. **Virtual method invocation** in java.

6. (15%) In mathematics, a **perfect number** is a positive integer that is the sum of its proper positive divisors, that is, the sum of the positive divisors excluding the number itself. Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself), or $\sigma 1(n) = 2n$. The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and $1 + 2 + 3 = 6$. Equivalently, the number 6 is equal to half the sum of all its positive divisors: $(1 + 2 + 3 + 6) / 2 = 6$.
   Please write a program which can list all the perfect numbers between 1 and 100.

試題隨卷繳回