Data Structure Midterm Examination
3:30pm-5:20pm (110 minutes), April. 17, 2017

#: _____   Student ID: _____   Name: _____

✧   看清題目再作答。

✧   中英文、鉛筆原子筆做答都可以。

✧   Please answer **questions 1 - 5 on the question sheet**.   For other questions, please answer on the answer sheet in any order you want.

✧   There are 9 questions.   You can obtain up to 114 points.

---

1.   **(12pt)** Please analyze the **time complexity** of the following algorithms

```
for (int i=0; i<N; i++)
  for (int j=0; j<i*i; j++)
    for (int k=1; k<j; k=k*2)
      print("Hello");
```

```
x = 2; y = 2;
while (x<N) {
  j = random 0 or 1;
  if (j == 0 && y<x) {
    y=y+1;
  }else{
    x=x*x;
  }
}
```
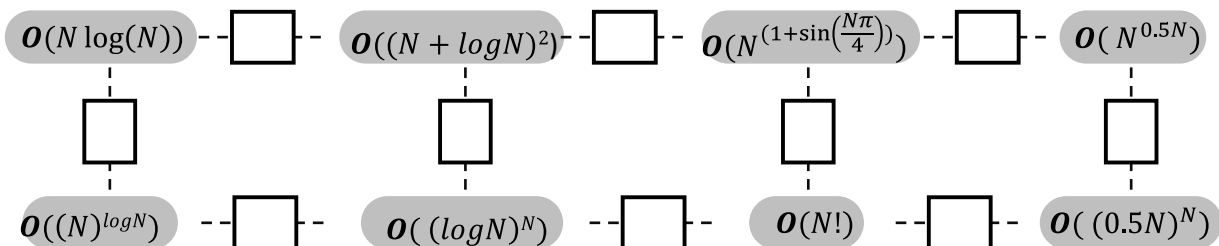
Ω(   )   Ω(   )

O(   )   O(   )

2.   **(15pt)** Please fill out the boxes with "=", "<", or ">" to show the asymptotic complexity hierarchy.   For example, **O**(n^2) $<$ **O**(n^3) = **O**($n^3/10$).

$O(N \log(N))$ --☐-- $O((N + logN)^2)$ --☐-- $O(N^{(1+\sin(\frac{N\pi}{4}))})$ --☐-- $O(N^{0.5N})$

☐   ☐   ☐   ☐

$O((N)^{logN})$ --☐-- $O((logN)^N)$ --☐-- $O(N!)$ --☐-- $O((0.5N)^N)$

3. **(12pt)** Please complete the following **recursive function** that **recursively divides an array into two halves** (i.e., 遞迴二分法) to find the maximum value in the array.

Please also show the **time complexity** using 1) a recursive definition and 2) a closed-form θ notation.

```
int RMAX(int array[], int size)
{
                                              your code goes here
    int m;



    return m;
}
```

(1) T(size) =   **(2)**

➜ **T(size) = θ(_____)**

KMP algorithm

(1) **(7pt)** Please show the 0-based failure function of the following patterns.

S	D	T	D	S	S	D	T	D	S	S	x
0											_____ if x == 'D' _____ if x == 'S' _____ if x == 'T' _____ otherwise

(2) **(8pt)** What is the next PosP index (0~5)?   A check mark (✔) stands for a match, and a cross mark stands for a mismatch.

PosS

| | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | | |

| C | D | D | C | C | D |
| 0 | 1 | 2 | 3 | 4 | 5 |

PosP

Next action:
PosS ++;
PosP = _____ ;

PosS

| | ✔ | ✔ | ✔ | ✔ | ✘ | | | |

| A | A | B | A | A | C |
| 0 | 1 | 2 | 3 | 4 | 5 |

PosP

Next action:
PosS = PosS;
PosP = _____ ;
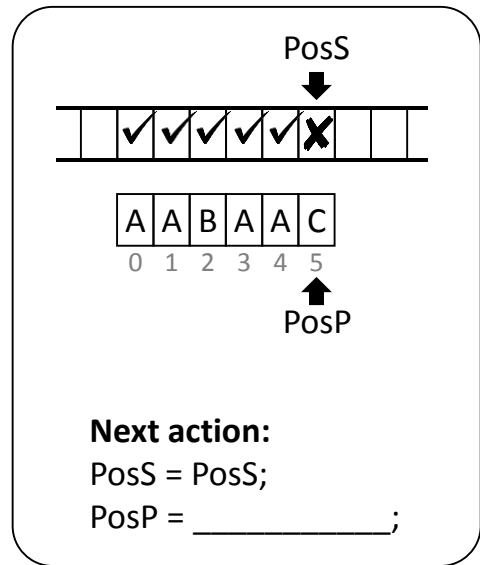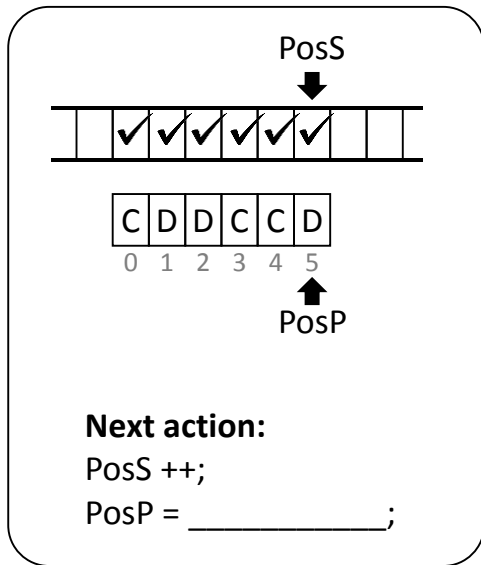
5.   **(12pt)** Please convert   **A || (A + B * C % 9) < D == T && B**   into a postfix expression.

Only the four boxes with thick borders will be graded (只有粗黑框格子計分).

bottom

Token	Stack	Output So Far
A		
||		
(		
A		
+		
B		
*		
C		
%		
9		
)		
<		
D		
==		
T		
&&		
B		
	Final output	

Priority	Operator
High	In-coming (
	*, /, %
	+, -
	<, <=, >=, >
	==, !=
	&&
	||
Low	In-stack (

6. **(12pt) Asymptotic notations**

$$\big(F(N) + G(N)\big) = \boldsymbol{O}(N!) \Rightarrow \big(F(N) \times G(N)\big) = \boldsymbol{O}((2N)!)$$

   (1) **(4pt)** Is the above statement **true or false**?

   (2) **(8pt)** Please prove your answer **according to the definition of Big-O**.

   Hint: $\frac{x+y}{2} \geq \sqrt{xy}$

7.

   (1) **(4pt)** What are the three basic structures of **structured programming**?

   (2) **(4pt)** To solve a problem, a structured program can always achieve **the same or better speed** than a non-structured program can.   Is this statement **true or false**? Please briefly explain.

   (3) **(4pt)** Can the access levels of object members affect the function of a program? Why don't we always use *public*?

8. **(12pt)** We want to design a data structure to store an lower-triangular matrix (LTM). The following figure shows a 3x3 LTM example.

$$\begin{bmatrix} T_{00} & 0 & 0 \\ T_{10} & T_{11} & 0 \\ T_{20} & T_{21} & T_{22} \end{bmatrix}$$

T_{00}	T_{10}	T_{11}	T_{20}	T_{21}	T_{22}
0	1	2	3	4	5

   (1) **(4pt)** How many integers are exactly needed for an NxN LTM if only the triangular part is stored in a 1-D array, row by row, as shown above?

   (2) **(4pt)** Following the above question, let f(i, j) be the array index of term T_{ij}.   Please show f(i, j).

   (3) **(4pt)** Let **R** be the probability that $T_{ij} = 0$.   If a data structure treats a LTM as a **sparse matrix** and stores only non-zero terms, how many integers are exactly needed for an NxN LTM?

9. **(12pt)** Please design a simple Tetris (俄羅斯方塊) game.　 The skeleton code is shown below.　 **Make reasonable assumptions and simplification** yourself if appropriate. Please write C/C++ style pseudo code and focus on the following two functions.

(1) **(6pt) Tetromino::tryMove( )** which tries to move the incoming tetromino piece, updates the tetromino's (x, y) coordinates, and returns whether the movement is successful.

(2) **(6pt) Playground::tryClearLines( )** which calculates how many horizontal lines on the playground are full, clears full lines, and moves other lines downward.

Tetromino

Playground

22

10
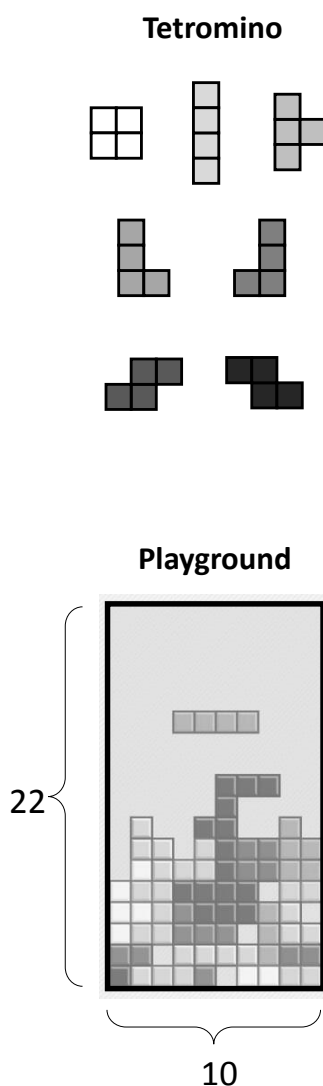
```cpp
class Playground;

class Tetromino{
friend class Playground;
    Tetromino();
    void reset(); // reset x, and y.  randomize type.
    int tryMove(int direction, const Playground & pg);
                    // 0: cannot move, 1: successfully move
private:
    int type;     // 1~7 for the seven types
    int x, y;     // Tetromino's position in the playground
    int score;
};

class Playgrond{
friend class Tetromino;
    Playground();
    void fix(const Tetromino & t); // fix the tetromino
    int tryClearLines();
    void draw(const Tetromino & curr_tetromino);
    void addScore(int s);
private:
    int grid[10][22];  // 0 for empty.  1~7 for tetrominos
};

int main()
{
  Tetromino t;
  Playground pg;
  While(1){
    int k = GetKey();  // LEFT, RIGHT, DOWN, or NONE
    if(k == NONE) {  k = DOWN; }
    if(k == DOWN && t.tryMove(k, pg) == 0) {
        pg.fix(t);
        pg.addScore(pg.tryClearLines());
        t.reset();
    }else{
        pg.draw(t);
    }
    sleep(0.5);  // sleep 0.5 second
  }
}
```