QUIZ #2

1. [알고리즘 분석] (15점)

1) 다음 code에서 문장 x=x+1 가 수행되는 시간 복잡도(time complexity) "θ notation"으로 구하시오

```
J:=n
while (j≥1) {
    for (i = 1; i ≤ j;  i++)
                x=x+1;
    j = ⌊j/2⌋;
}
```

(답) 1st loop, the statement is executed n times, → t(n) ≥ n and t(n) = Ω(n)

. .. x = x+1 is executed at most n + n/2 + n/4 + ... + $n/2^{k-1}$

t(n) ≤ $\frac{n(1-1/2^k)}{1-1/2}$ = $2n(1-1/2^k)$ ≤ 2n, so t(n) = O(n),

therefore t(n) = Θ(n)

2) 다음 전체 segment code의 시간 복잡도(time complexity)를 Big-O notation으로 구하시오

```
for i = 2 to n do
   a[i] = 0
   for i = 1 to n do
       for j = 1 to n do
          a[i] := a[i] + a[j]
```

(답) MAX(O(n) + $o(n^2)$) = $O(n^2)$

2. [배열] (15점)

1) 다음 행렬에 대한 전치행렬(transposed matrix)을 구하시오.

A = 1 3 5 7
    5 7 9 4
    4 9 5 9

A^T = 1 5 4
        3 7 9
        5 9 5
        7 4 9

2) 다음의 희소 행렬을 2차원 배열의 논리적 구조를 **행 우선순위** 와 **열 우선순위로** 각각 표현 하시오).

A = 0 0 0 9
    0 1 0 0
    0 0 0 0
    0 0 7 0
    0 0 0 0
    3 0 0 0
    0 0 0 0

< Row major>

row	col	value
4	4	7
0	3	9
1	1	1
3	2	7
5	0	3

<col. Major>

row	col	value
4	4	7
0	5	3
1	1	1
2	3	7
3	0	9

3. Magic Square 는 1에서 n2까지의 정수로된 nxn행렬로서, 각행의 합, 열의 합, 주 대각선의 합이 모두 같다. 크기가 5인(5x5) Magic Square를 작성하시오 (10점) (조건: 첫번째 행의 중앙에 1을 넣고, 이동은 **왼쪽 위로 이동한다**. 표를 작성할 것) * 오른쪽위로 이동은 감점입니다.

15	8	1	24	17
16	14	7	5	23
22	20	13	6	4
3	21	19	12	10
9	2	25	18	11

4. [stack] (10점)

1) 스택을 활용할 수 있는 예를 하나 설명하시오.

부페식당의 접시?

2) 스택 메모리에 대한 정보의 입/출력 방식은?

가. FIFO 나. FILO 다. LILO ***라. LIFO***

3) 스택의 응용 분야와 거리가 먼 것은?

가. 운영체제의 작업 스케줄링 나. 함수 호출의 순서 제어

다. 인터럽트의 처리 라. 수식의 계산

5. [stack] 만약, STACK operation 이 Array 의 끝에서부터 시작된다면 (Array size 10), A,B,C 를 PUSH 한 결과는 다음 그림과 같을 것이다. 이러한 상황에 적합한, STACKCREATE, PUSH, POP ADT 를 작성하라. (10점)

index	1	2	3	4	5	6	7	8	9	10
stack								C	B	A

```
void stackcreate( )  {top = 11};

void  PUSH (int newdata) { stack[--top] = newdata};

void  POP ( ) {return stack[top++]};
```

6. [Queue] (10점)

1) 일상생활에서 발견할 수 있는 큐의 예를 설명하여라.

예) 은행에서 번호표 순서대로 처리하기, 식당에서 줄서기

2) 운영체제의 작업 스케줄링 등에 응용되는 것으로 가장 적합한 자료구조는?

① 스택 *② 큐* ③ 연결리스트 ④ 트리

3) 1차원 배열의 선형 큐에서 잘못된 포화상태 문제(한 개의 공간이 남아있는 문제)를 해결할 수 있는 방법을 두가지만 설명하시오.

방법1) 원소들을 비어있는 앞자리로 이동하기 *방법2) 원형큐로 만들기*

➔ flag 사용하기, count 사용하기

7. 배열을 이용하여 다음 사항을 만족하는 'QueueAscending' 함수를 작성하시오 (10점)

(가정: Queue is not empty;
결과: Returns TRUE, if the elements in Q are in ascending order(오름차순). Otherwise returns FALSE.)

```
int queascending ( )
    bool check = true;   int count;   int front = 0;
    while ((count <= maxquesize) && check)   {
          if (Queue[front] >   Queue [front+1]) then
                return check = false;     // exit
          else      {front= front +1;
                count = count + 1};
         }
    return check;
}
```

8. 크기가 5인 선형 큐에서 다음의 연산을 수행한다가, 큐가 포화상태가 되어 더 이상 작업을 할 수 없게 되는 시점을 답하시오. (5점)

A 삽입 → B 삽입 → 삭제 → C 삽입 → 삭제 → 삭제 → D 삽입 → E 삽입 → 삭제 → ***F 삽입*** → 삭제 → 삭제 → 삽입G

⇨ *큐가 포화상태이기 때문에 F삽입연산을 수행할 수 없다.*

9. 다음 수식표현에 관하여 답하라. (15점) (연산 과정을 강의노트와 같이 테이블로 작성할 것)

1) 다음 수식을 postfix notation으로 바꾸라. **수식: A+B-C*(D*E+F-G)**

Case:

1) operand -> print **2) "(" -→ PUSH**

3) operator 3) ")" → while (stack[top]!="(")

Stack[top] > token → print stack[top]; PUSH(token) print(POP(top))

≤ token → PUSH (token) POP(top)

5) 스택처리: while ((token=POP(top)) != eos) print (token)

(답)

Token	stack	postfix
A		A
+	+	A
B	+	AB
-	[- +]	same
C	same	ABC
*	[* - +]	same
(	[(* - +]	ABC
D	same	ABCD
*	[* (* - +]	same

Token	stack	postfix
*	[* (* - +]	ABCD
E	same	ABCDE
+	[+ (* - +]	ABCDE*
F	same	ABCDE*F
-	[- + (* - +]	same
G	same	ABCDE*FG
)	[* - +]	ABCDE*FG-+
$end	→ ABCDE*FG - + * - +	

답: ABCDE*FG - + * - +

2) 다음 수식이 (A =7.0, B = 4.0, C= 3.0, D = -2.0 일때) stack에서 계산(evaluation) 되는 과정을 설명하시오 **수식: ABC+/D*** **결과값: -2**

(답)

A	B A	C B A	B + C A	A/(B+C)	D A/(B+C)	A/(B+C)*D