

Group Activity 02

(3인 혹은 4인으로 팀을 구성하여 아래의 문제를 푼다. 팀 구성은 매 시간마다 달라져도 된다.)

팀원1: _____

팀원2: _____

팀원3: _____

팀원4: _____

1. 다음의 순환 함수의 반환값을 x 와 y 의 함수로 나타내면?

```
int fun1(int x, int y)
{
    if (x > y)
        return 0;
    return y + fun1(x, y-1);
}
```

2. 다음의 순환함수의 반환값을 n 의 함수로 나타내면?

```
/* Assume that  $n \geq 1$  */
int fun2(int n)
{
    if(n == 1)
        return 0;
    else
        return 1 + fun2(n/2);
}
```

3. 다음의 순환함수가 결과적으로 하는 일은?

```
void fun3(int n)
{
    if (n == 0)
        return;
    fun3(n/2);
    printf("%d", n%2);
}
```

4. 다음의 순환함수가 결과적으로 하는 일은?

```
void fun4(char str[], int start, int end)
{
    if (start >= end)
        return;
    else {
        char temp = str[start];
        str[start] = str[end];
        str[end] = temp;
        fun4(str, start + 1, end - 1);
    }
}
```

5. 다음의 함수 fun5의 반환값을 a 와 b 에 관한 식으로 표현하면?

```
int fun(int x, int y)
{
    if (y == 0) return 0;
    return (x + fun(x, y-1));
}

int fun5(int a, int b)
{
    if (b == 0) return 1;
    return fun(a, fun5(a, b-1));
}
```

6. 다음 함수가 결과적으로 하는 일을 최대한 간명하게 설명하라.

```
int fun6(int a[], int n)
{
    if(n == 1)
        return a[0];

    int x = fun6(a, n-1);
    return (x > a[n-1] ? x : a[n-1]);
}
```

7. 다음 함수가 결과적으로 하는 일을 최대한 간명하게 설명하라.

```
double fun7(double a[], int n)
{
    if (n==1) return a[0];
    else
        return (a[n-1] + (n-1)*fun7(a, n-1))/n;
}
```

8. 다음 함수가 결과적으로 하는 일을 최대한 간명하게 설명하라.

```
int fun8(int a, int b)
{
    if (b == 0)
        return 1;
    if (b % 2 == 0)
        return fun8(a*a, b/2);
    return fun8(a*a, b/2)*a;
}
```

9. 다음 함수가 결과적으로 하는 일을 최대한 간명하게 설명하라.

```
void fun9(int arr[], int start_index, int end_index)
{
    if(start_index >= end_index)
        return;
    int min_index;
    int temp;

    /* 함수 minIndex()는 배열 arr[start_index...end_index]에서 최소값의 인덱스를 반환한다고
    가정한다. */
    min_index = minIndex(arr, start_index, end_index);

    temp = arr[start_index];
    arr[start_index] = arr[min_index];
    arr[min_index] = temp;

    fun9(arr, start_index + 1, end_index);
}
```

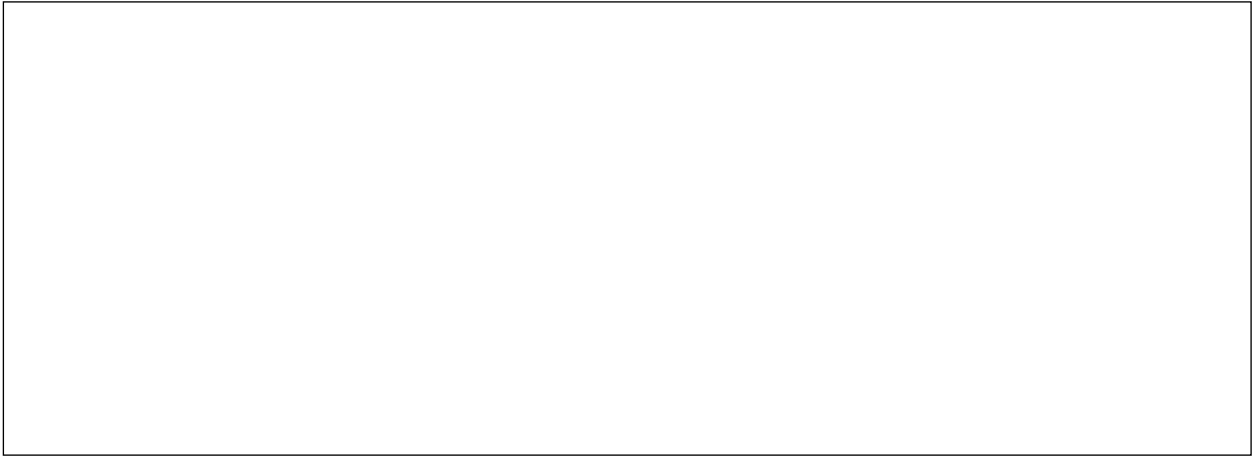
10. 다음 함수 fun10이 일을 최대한 간명하게 설명하라. 이 함수는 아래의 main함수에서 처럼 호출된다.

```
void fun10(int arr[], int n, int t) {
    if (n==0)
        arr[0] = t;
    else if (t>=arr[n - 1])
        arr[n] = t;
    else {
        arr[n] = arr[n-1];
        fun10(arr, n-1, t);
    }
}

int main() {
    int data[] = {7, 1, 3, 8, 1, 9, 2, 10};
    int result[8];
    for (int i=0; i<8; i++)
        fun10(result, i, data[i]);
    ...
}
```

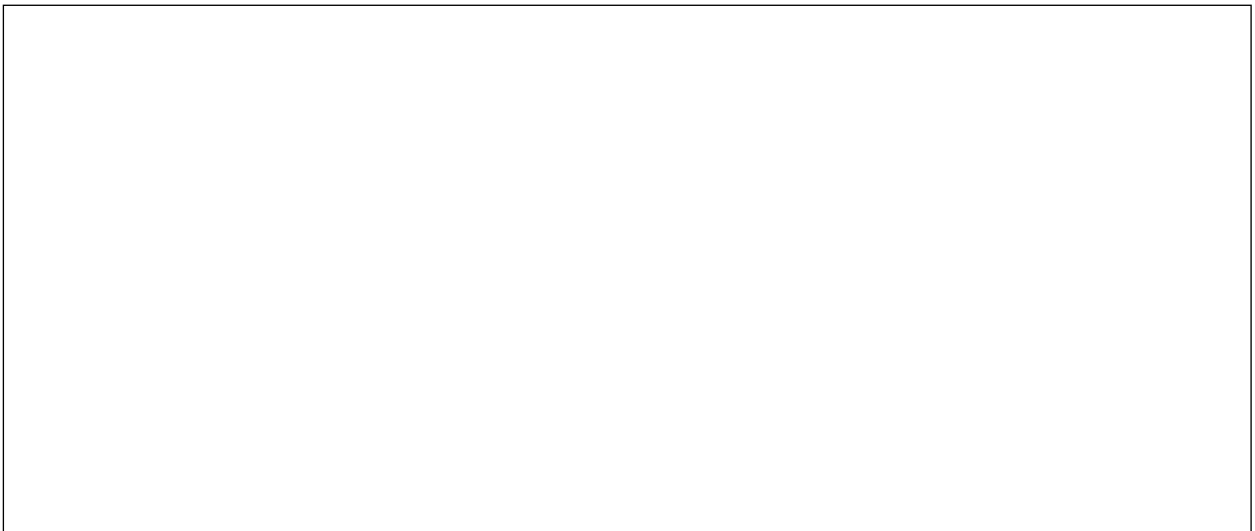
11. 다음은 N-queen 문제를 푸는 순환함수이다. 이 함수를 변형하여 N-queen 문제의 서로 다른 해의 개수를 구하는 함수를 작성하라. promising 함수는 작성할 필요가 없다.

```
int cols[N+1];
bool queens( int level )    {
    if (!promising(level))
        return false;
    else if (level==N)
        return true;
    for (int i=1; i<=N; i++) {
        cols[level+1] = i;
        if (queens(level+1))
            return true;
    }
    return false;
}
```



12. 다음은 강의 슬라이드에 있는 미로에서 출구에 도달하는 경로가 존재하는지 검사하는 함수이다. 이 함수를 부분적으로 수정하여 출구까지 가는 서로 다른 경로의 개수를 카운트하여 반환하는 함수를 작성하라.

```
bool findMazePath(int x, int y) {
    if (x<0 || y<0 || x>=N || y>=N || maze[x][y] != PATHWAY_COLOUR)
        return false;
    else if (x==N-1 && y==N-1) {
        maze[x][y] = PATH_COLOUR;
        return true;
    }
    maze[x][y] = PATH_COLOUR;
    if (findMazePath(x-1,y) || findMazePath(x,y+1)
        || findMazePath(x+1,y) || findMazePath(x,y-1)) {
        return true;
    }
    maze[x][y] = BLOCKED_COLOUR;
    return false;
}
```



13. 입력으로 주어진 N 개의 정수들을 합이 동일한 두 집합으로 분할할 수 있는지 물어보는 문제를 PARTITION 문제라고 부른다. 이 문제를 해결하는 되추적기법 알고리즘을 고안하라. 상태공간트리를 기술하고 알고리즘을 말이나 pseudo 코드의 형태로 제시하라.

14. 하나의 그래프와 사용할 색의 개수 m 이 입력으로 주어진다. 그래프의 각각의 노드를 m 개의 색 중 하나로 칠하면서 어떤 인접한 두 노드도 동일한 색으로 칠해지지 않도록 할 수 있는지 검사하는 문제이다. 예를 들어 아래의 그림은 $m = 3$ 개의 색으로(red, green, white) 칠한 결과이다. 되추적(backtracking) 기법으로 이 문제를 해결하려고 한다. 적절한 상태공간트리를 구상하고 알고리즘을 pseudocode의 형태로 기술하라.

