

2022년 1학기

프로그래밍과 문제해결

Assignment #3

담당교수: 윤은영

학번: 20220100

학과: 무은재학부

이름: 박기현

POVIS ID: kihyun

명예서약(Honor Code)

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

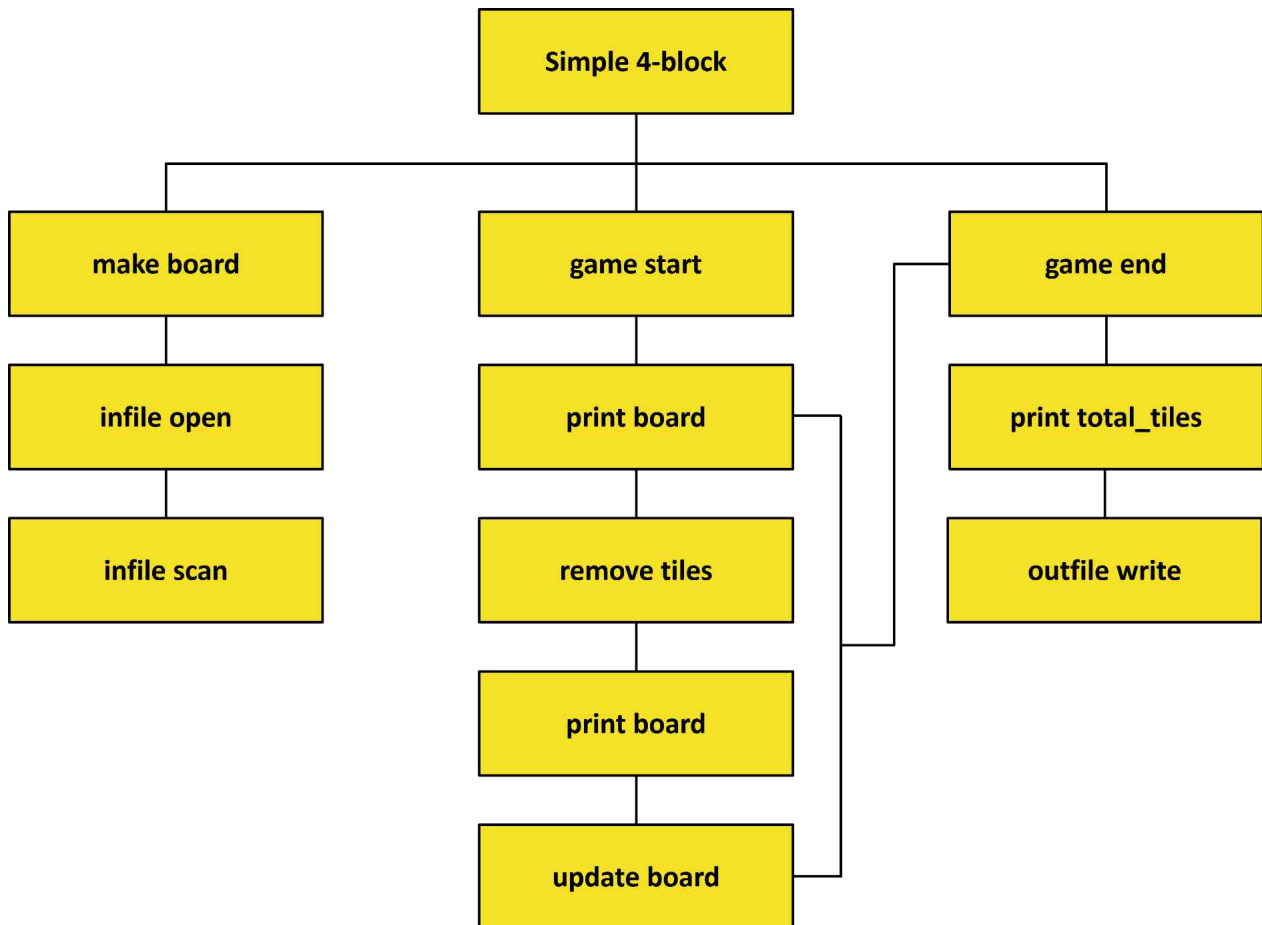
Problem. Simple 4-block

1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- 게임판은 최대 10X10의 크기를 가지며, 각 행과 열의 길이는 5 이상이다. 즉, 게임판을 이루는 총 타일의 개수는 25 이상, 100 이하이다.
- 각 타일은 8가지의 무늬 중 하나를 가질 수 있으며, 무늬는 1~8까지의 숫자로 총 8종류이다.
- 2X2 형태의 연속된 타일들이 같은 무늬를 가진다면, 타일들이 지워진다. 하나의 타일은 여러 2X2에 포함될 수 있으며, 해당되는 2X2 형태가 여럿 있다면 동시에 지워진다.
- 지워진 2X2 형태의 빈 자리는 위쪽의 타일들이 내려와서 채우게 된다.
- 더 이상 제거할 수 있는 2X2 형태가 없을 때까지 제거와 채우는 동작을 반복한다.
- 더 이상 제거 가능한 타일이 없다면, 지금까지 제거된 타일의 수를 기반으로 점수를 계산하고 출력한다.

이 때 사용되는 구상 가능한 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.



2. 알고리즘

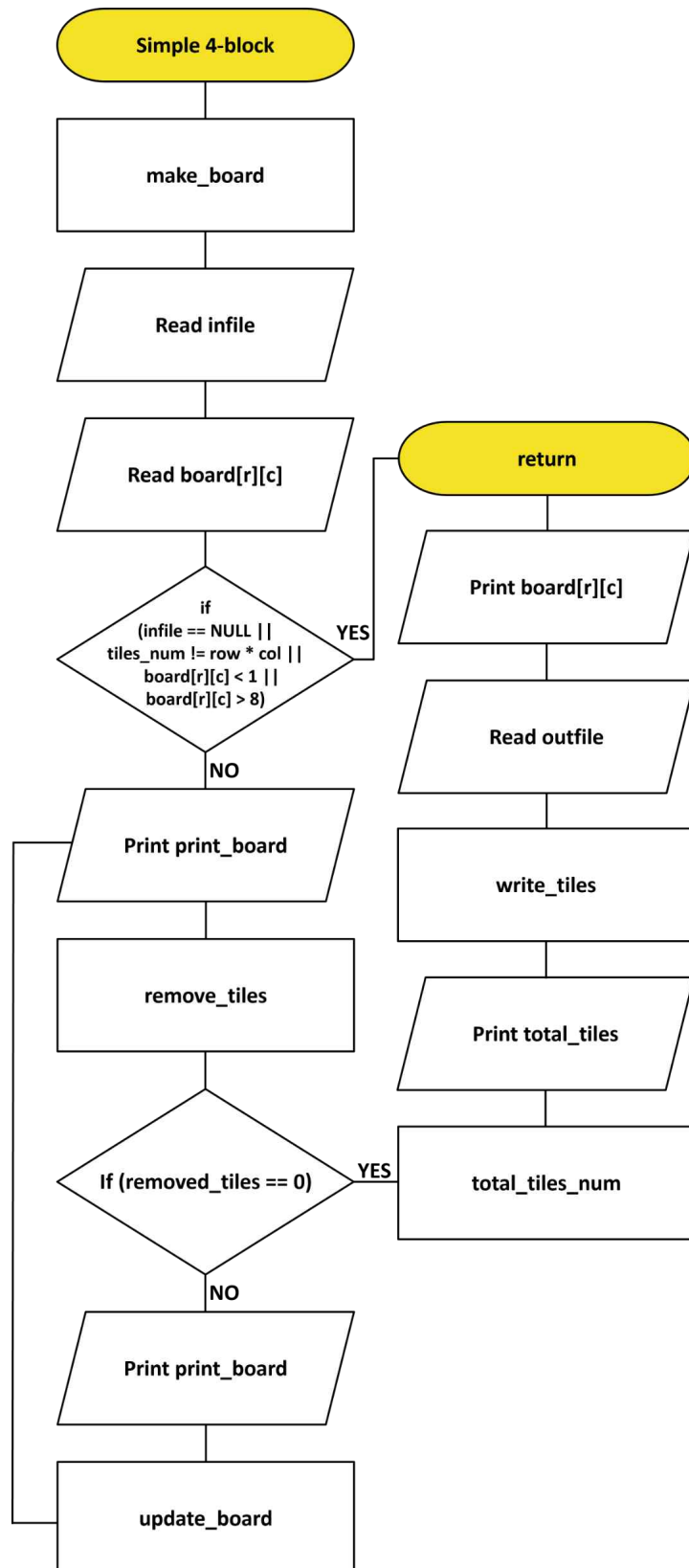
본 프로그램 작성을 위한 알고리즘을 Pseudo 코드 형태로 나타내면 다음과 같다.

Pseudo-algorithm for Simple 4-block

// 프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

```
1  include an essential header <stdio.h> / <stdlib.h>
2  make_board
    fscanf infile and read board[MAX_SIZE][MAX_SIZE]
3  while(1)
    print_board
    remove_tiles
    update_board
4  if ((remove_tiles) == -1)
    break
5  total_tiles_num
6  printf total_tiles
7  write_tiles
```

위의 의사 알고리즘을 flow chart를 통해 표현하면 아래와 같다.



3. 프로그램 구조 및 설명

a) 게임 시작

- 프로그램을 실행하면, 함께 주어진 board.txt 파일을 읽어 화면에 게임판을 출력한다.

b) 사용자의 입력 후 타일 제거

- 사용자가 [Enter] 키를 입력하면, 화면을 지운 후 게임판에서 연속된 무늬가 2X2로 배치되어 있는 타일들을 모두 찾아 제거한다.
- 타일이 제거 된 빈 칸은 0으로 채워 출력한다.

c) 사용자의 입력 후 타일 채우기

- 사용자가 [Enter] 키를 입력하면, 화면을 지운 후 제거 된 자리(0으로 채워진 위치)를 위의 타일들이 차례로 내려와 채운다.
- 이때 생기는 빈 칸은 이전과 같이 0으로 채워서 출력한다.

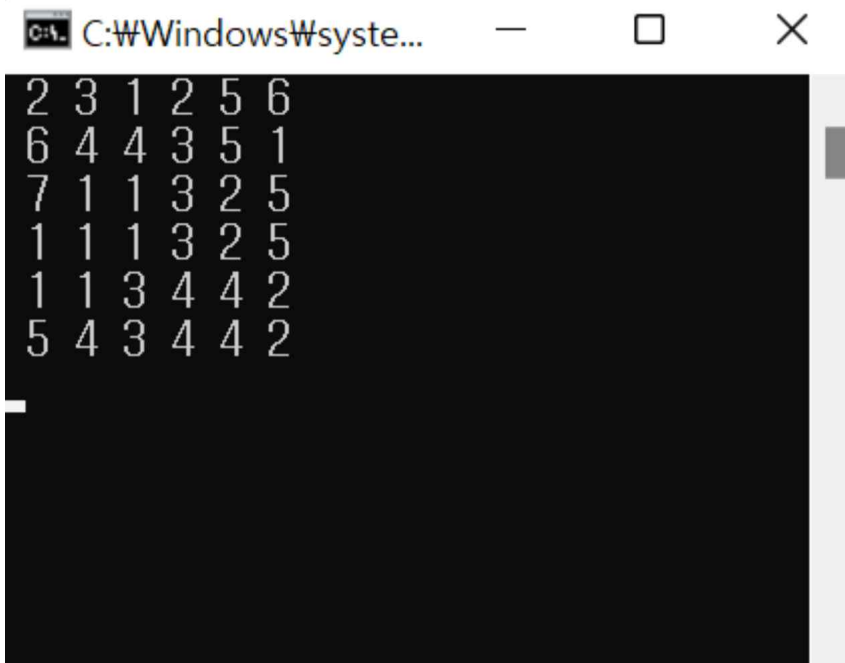
d) 게임 반복

- 더 이상 제거 가능한 형태가 없을 때까지 (b) ~ (c)의 과정을 반복한다.

e) 게임 종료

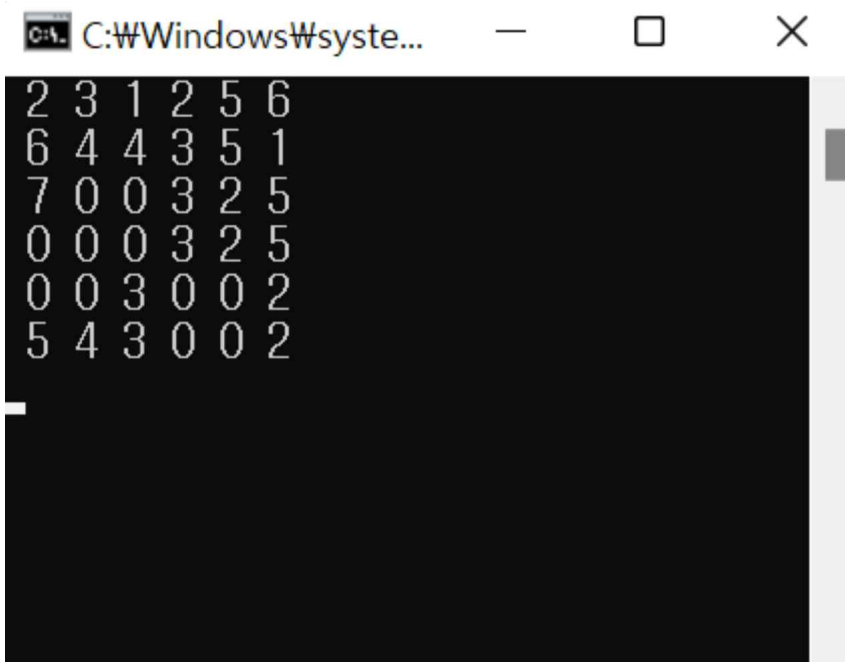
- 더 이상 제거 가능한 형태가 없으면, 게임 과정에서 제거된 타일의 총 개수를 출력한다.
- 최종 게임판과 (e)의 출력 결과를 result.txt에 저장하고 게임을 종료한다.

4. 프로그램 실험방법 및 예제



A screenshot of a Windows command prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The window contains a 6x6 grid of numbers:

2	3	1	2	5	6
6	4	4	3	5	1
7	1	1	3	2	5
1	1	1	3	2	5
1	1	3	4	4	2
5	4	3	4	4	2



A screenshot of a Windows command prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The window contains a 6x6 grid of numbers:

2	3	1	2	5	6
6	4	4	3	5	1
7	0	0	3	2	5
0	0	0	3	2	5
0	0	3	0	0	2
5	4	3	0	0	2

```
C:\Windows\system32>
0 0 0 0 0 6
0 0 0 0 0 1
2 0 1 2 5 5
6 3 4 3 5 5
7 4 3 3 2 2
5 4 3 3 2 2
```

```
C:\Windows\system32>
0 0 0 0 0 6
0 0 0 0 0 1
2 0 1 2 0 0
6 3 4 3 0 0
7 4 0 0 0 0
5 4 0 0 0 0
```



```
C:\Windows\system32\cmd.exe
0 0 0 0 0 0
2 0 0 0 0 0
6 3 0 0 0 0
7 4 1 2 0 6
5 4 4 3 0 1

=====
Total removed tiles: 23
=====
계속하려면 아무 키나 누르십시오 . . .
```

```
result.txt - 메모장
파일 편집 보기
0 0 0 0 0 0
0 0 0 0 0 0
2 0 0 0 0 0
6 3 0 0 0 0
7 4 1 2 0 6
5 4 4 3 0 1

=====
Total removed tiles: 23
=====

줄 11, 열 1 | 100% | Windows (CRLF) | UTF-8
```

5. 토론

- 파일을 읽고 게임판을 출력하는 과정에서 게임판 타일의 개수가 올바르지 않는 경우, 게임판이 1 ~ 8 이외의 숫자를 가지는 경우에 따라 오류메세지를 출력하도록 구현하는 것이 힘들었으나, 조건문과 반복문의 원리를 살펴보며 해결할 수 있었다.
- 타일을 제거한 후 빈 칸을 위의 타일이 순서대로 채우게 하기 위해서 어떤 식으로 구현해야 할지 막막하였고, 가장 많은 시간이 소요되었다. 타일을 제거할 때와 같이 새로운 배열을 선언하여 구현해보려고 노력하였고, 그 과정에서 배열 대신 새로운 변수를 선언하여 새로운 열을 만들면 그 열에 값을 차례대로 입력함으로써 해결할 수 있을 것이라 판단하였다. 최종적으로 많은 고민과 수정 끝에 프로그램을 완성할 수 있었다.

6. 결론

- 본 과제는 사용자 정의 함수와 반복문, 조건문, 포인터, 2차원 배열, 텍스트 파일을 통한 입출력 등 C언어의 다양한 기능을 정확히 이해하고 있어야 하는 문제였다. 문제를 해결하는 과정에서 어려움을 겪었던 2차원 배열, 포인터 등을 다시 한 번 이해하는 시간을 가질 수 있었다.

7. 개선방향

- 앞으로 사용하게 될 배열과 텍스트 파일을 통한 입출력, 포인터 등을 더 정확히 이해하여 다음 과제를 수행하는 데에 있어 조금 더 수월히 할 수 있도록 노력해야겠다고 생각하였다.