

CSED101. Programming & Problem solving

Spring 2022

Programming Assignment #3 (70 points)

김동원 (kdwon@postech.ac.kr)

■ 제출 마감일: 2022.05.03 23:59

■ 개발 환경: Windows Visual Studio 2019

■ 제출물

- C 소스 코드 (assn3.c)
 - 프로그램의 소스 코드에 채점자의 이해를 돕기 위한 주석을 반드시 붙여주세요.
- 보고서 파일 (.docx, .hwp 또는 .pdf; assn3.docx, assn3.hwp 또는 assn3.pdf)
 - 보고서는 AssnReadMe.pdf를 참조하여 작성하시면 됩니다.
 - 명예 서약 (Honor code): 표지에 다음의 서약을 기입하여 제출해 주세요: “나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.” 보고서 표지에 명예 서약이 기입되어 있지 않은 과제는 제출되지 않은 것으로 처리됩니다.
 - 작성한 소스 코드와 보고서 파일은 PLMS를 통해 제출해 주세요.

■ 주의 사항

- 컴파일이나 실행이 되지 않는 과제는 0점으로 채점됩니다.
- 제출 기한보다 하루 늦게 제출된 과제는 최종 20%, 이를 늦게 제출된 과제는 최종 40% 감점됩니다. 제출 기한보다 사흘 이상 늦으면 제출 받지 않습니다 (0점 처리).
- 각 문제의 제한 조건과 요구 사항을 반드시 지켜 주시기 바랍니다.
- 모든 문제의 출력 형식은 채점을 위해 아래에 제시된 예시들과 최대한 비슷하게 작성해 주세요.
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 “POSTECH 전자컴퓨터공학부 부정행위 정의”를 따릅니다 (PLMS의 본 과목 공지사항에 등록된 글 중, 제목이 [document about cheating]인 글에 첨부되어 있는 disciplinary.pdf를 참조하세요).
- 이번 과제는 추가 기능 구현과 관련된 추가 점수가 따로 없습니다.

■ Problem: Simple 4-block

(목적)

- 2차원 배열의 선언과 사용을 익힙니다.
- 함수에서 2차원 배열을 매개변수로 사용하는 방법을 익힙니다.
- 텍스트 파일을 통한 입출력을 익힙니다.

(주의사항)

1. 이번 과제는 2차원 배열을 선언하고 사용하는 과제이므로, 게임판은 반드시 2차원 배열로 선언 후 사용해야 합니다.
2. 전역 변수, goto 문, string 관련 함수, 구조체는 사용할 수 없으며, 포인터의 경우 수업 시간에 다른 내용에 한해서 사용이 가능합니다.
3. 모든 기능을 main() 함수에 모든 기능을 구현한 경우 감점 처리 합니다. 기능적으로 독립됐거나 반복적으로 사용되는 기능은 사용자 정의 함수를 정의해서 구현하여야 합니다.
4. 프로그램 구현 시, main() 함수를 호출을 직접 하지 않습니다. 즉, 소스 코드 내에 main(); 이라고 호출하지 않습니다.
5. 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성 바랍니다.
6. 문서에 반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인 후 구현 하도록 합니다. 이외에 필요한 함수는 정의해서 사용할 수 있습니다.
7. 명시된 예러 처리 외에는 고려하지 않아도 됩니다.

(문제)

이번 과제에서는 주어진 게임판에서 타일을 제거하는 퍼즐 게임을 간소화하여 구현하려고 합니다. 해당 게임에서, 같은 무늬(숫자)를 가진 타일이 2x2 형태로 4개가 연속적으로 배치되어 있을 경우 타일들이 사라지면서 플레이어는 점수를 얻게 됩니다. 타일이 제거된 빈 자리는 위에 위치한 타일들이 내려오면서 채우게 되고, 더 이상 제거될 수 있는 타일이 남아있지 않은 경우에 게임이 끝나게 됩니다.

(퍼즐 규칙)

- 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.

(설명)

1. 프로그램을 실행하면, 함께 주어진 board.txt 파일을 읽어 아래 그림 1과 같이 게임판을 출력한다. 파일의 첫 줄에는 게임판의 새로 길이(행)과 가로 길이(열)이 함께 주어지며, 다음 줄부터는 게임판을 이루는 타일들의 무늬가 주어진다. 출력형식은 아래 주어진 그림 1의 예시를 참고하여 작성한다.

board.txt 예시

```
6 6
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
```

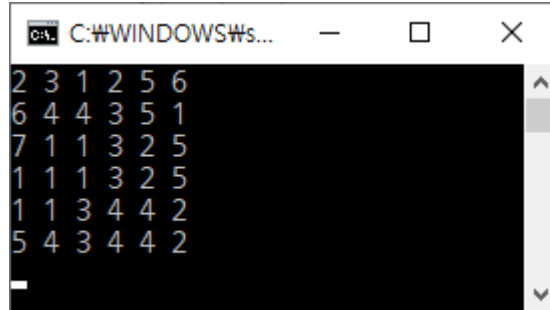


그림 1

예외처리)

- (1) 파일이 없거나, (2) 게임판이 1~8 이외의 숫자를 가지거나, (3) 게임판 타일의 개수가 주어진 행과 열의 곱과 다를 경우에는 적절한 에러 메시지를 출력하고 프로그램을 종료한다.
2. 그림 1과 같이 게임판을 출력 후, 사용자의 입력을 기다린다. 사용자가 [Enter]키를 입력하면, 화면을 지운 후 게임판에서 연속된 무늬가 2x2로 배치되어 있는 타일들을 모두 찾아 제거한 후, 아래의 그림 2-1과 같이 출력한다. 타일이 제거된 빈 칸은 0으로 채워 출력하도록 한다.
 - 모든 게임판은 출력하기 전에 이전 화면을 지우도록 한다. (화면 지우기는 5쪽의 "화면 지우기"를 참고)
 - 사용자 입력으로 [Enter]키 입력 외에는 고려할 필요 없음.

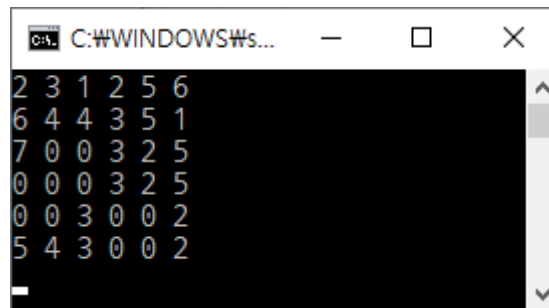


그림 2-1

- 연속된 무늬의 2x2 형태가 여럿일 시, 이들은 동시에 제거된다. 또한, 하나의 무늬가 여러 2x2 형태에 동시에 포함될 수 있다 (최대 4개). 아래의 그림 2-2 을 참고하라.

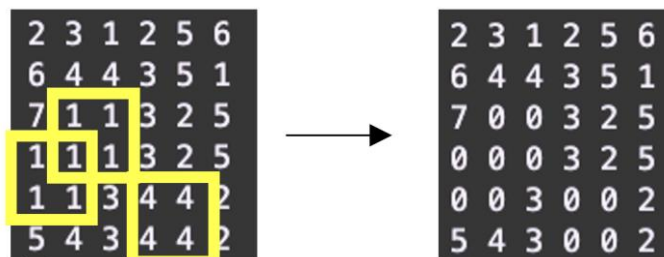


그림 2-2

(힌트) 해당 기능 구현 시, 게임판을 저장하고 있는 배열 외에 다른 배열을 선언하여 지울 칸을 표시하는 용도로 사용하면 좋습니다.

3. 제거된 화면을 출력한 후, 사용자의 입력을 기다린다. 사용자가 [Enter]키를 입력하면, 제거된 자리(0으로 채워진 위치)를 위의 타일들이 차례로 내려와 채우게 된다(그림 3). 이때 생기는 빈 칸은 이전과 같이 0으로 채워서 출력한다. 이때도 이전 화면을 지우고 새로운 화면을 출력하도록 한다.

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

그림 3

4. 더 이상 제거 가능한 형태가 없을 때까지 2~3까지의 과정을 반복한다(그림 4).

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	→	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	→	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	→	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	→	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
--	---	--	---	--	---	--	---	--

그림 4

5. 더 이상 제거 가능한 형태가 없으면, 게임 과정에서 제거된 타일의 총 개수를 출력한다.

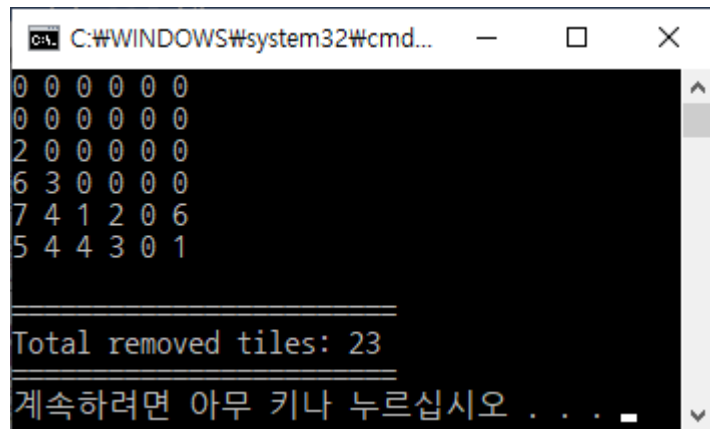


그림 5

6. 최종 게임판과 5의 출력 결과를 result.txt에 저장하고 게임을 종료한다.

result.txt 예시

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					
=====					

(함수 요구 사항)

다음 함수들을 반드시 작성하여 프로그램을 구현해야 합니다. 이 함수들을 제외하고, 필요한 함수가 있다면 자유롭게 정의하여 사용하도록 합니다. 아래 명시된 함수 이름은 변경하지 말아주세요. 매개변수의 개수와 자료형, 반환 자료형 등은 자유롭게 변경이 가능합니다. 그러나 동일한 기능을 하는 함수는 반드시 있어야 하며, 변경 시 무엇을 어떻게 변경해서 구현했는지 보고서에 기록하도록 합니다.

- **#define MAX_SIZE 10**
게임판 구성을 위해 최대로 가질 수 있는 크기를 상수로 정의
- **int main()**
프로그램이 시작되면 make_board 함수를 호출하여 게임판 배열을 생성한다. 이때, 게임판은 int board[MAX_SIZE][MAX_SIZE] 배열로 표현된다. 이 배열은 main함수 안에서 선언되어야 함.
- **int make_board(int board[][MAX_SIZE], int *row, int *col)**
board.txt 파일을 읽고 게임판의 행의 크기와 열의 크기를 저장한다. 파일이 없는 경우, 적절한 정수를 반환하여 프로그램을 종료하기 위해 활용한다.
- **void print_board(int board[][MAX_SIZE], int row, int col)**
게임판(2차원 배열)을 전달받아 위에 주어진 그림 1과 같이 게임판을 출력하는 함수
- **int remove_tiles(int board[][MAX_SIZE], int row, int col)**
게임판에서 제거 가능한 2x2 형태를 제거하고 제거된 무늬의 존재 여부에 따라 적절한 정수 값을 반환하며, 이는 프로그램을 종료하는 데에 사용되도록 한다.
- **void update_board(int board[][MAX_SIZE], int row, int col)**
2x2 형태가 제거된 빈 칸에 위쪽의 무늬들이 순서대로 내려와 채우도록 하는 함수

[Tip]

(a) 화면 지우기

화면을 지우기 위해서는 stdlib.h에 포함된 system 함수를 사용합니다. 윈도우의 경우에는 매개변수로 "cls"를 넘겨주면 되고, Linux나 Mac 환경에서는 "clear"를 넘겨주면 됩니다. 다음 예제 코드를 실행해 보면, "Hello world"가 먼저 출력된 뒤, [Enter] 키를 입력하면 이전의 출력이 삭제되고 "Bye world"가 출력되는 것을 확인할 수 있습니다.

```
#include <stdio.h>
#include <stdlib.h> // system() 사용을 위해 포함

int main()
{
    printf("Hello world\n");
    getchar();

    system("cls"); // 리눅스 또는 맥의 경우 system("clear");
    printf("Bye world\n");

    return 0;
}
```

(b) 프로그램 작성 중 무한 루프에 의해 프로그램이 끝나지 않을 경우, [Ctrl]+[c]를 누르면 강제 종료됩니다.