2021년 2학기

프로그래밍과 문제해결

Assignment #2

담당교수: 윤은영

학번: 20210370

학과: 무은재학부

이름 권민경

POVID ID: mkkwon

명예서약(Honor Code)

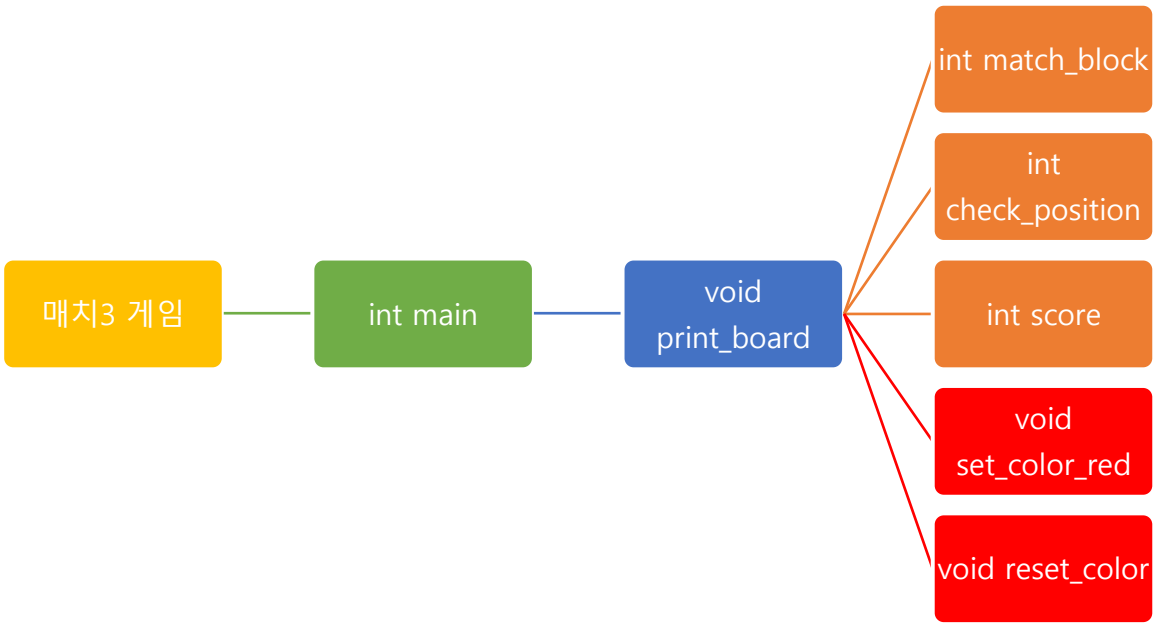
"나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다."

매치3 게임

1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- 6 by 6 보드에서 5턴동안 블록 3개 이상을 맞추는 게임이다.
- 사용자는 바꿀 보드의 위치를 입력하고 입력대로 보드가 교환된다.
- 이때 교환은 상 하 좌 우 블록끼리만 교환이 가능하다.
- 최종 점수를 표시하고 게임이 종료된다.



입력부: 게임을 진행하기 위해 보드판에서 바꾸고 싶은 위치 두 곳을 입력받는다.

처리부:

Print_board: 만약 입력값이 상하좌우에 위치한 유효한 블록이라면 두 배열의 원소를 바꾼다. 빨간색/삭제 처리하고 다시 랜덤으로 원소를 배열한다.

Match_block: 3개이상이 연속하는 것을 검사하는 함수

Score: 점수를 누적해서 업데이트 한다.

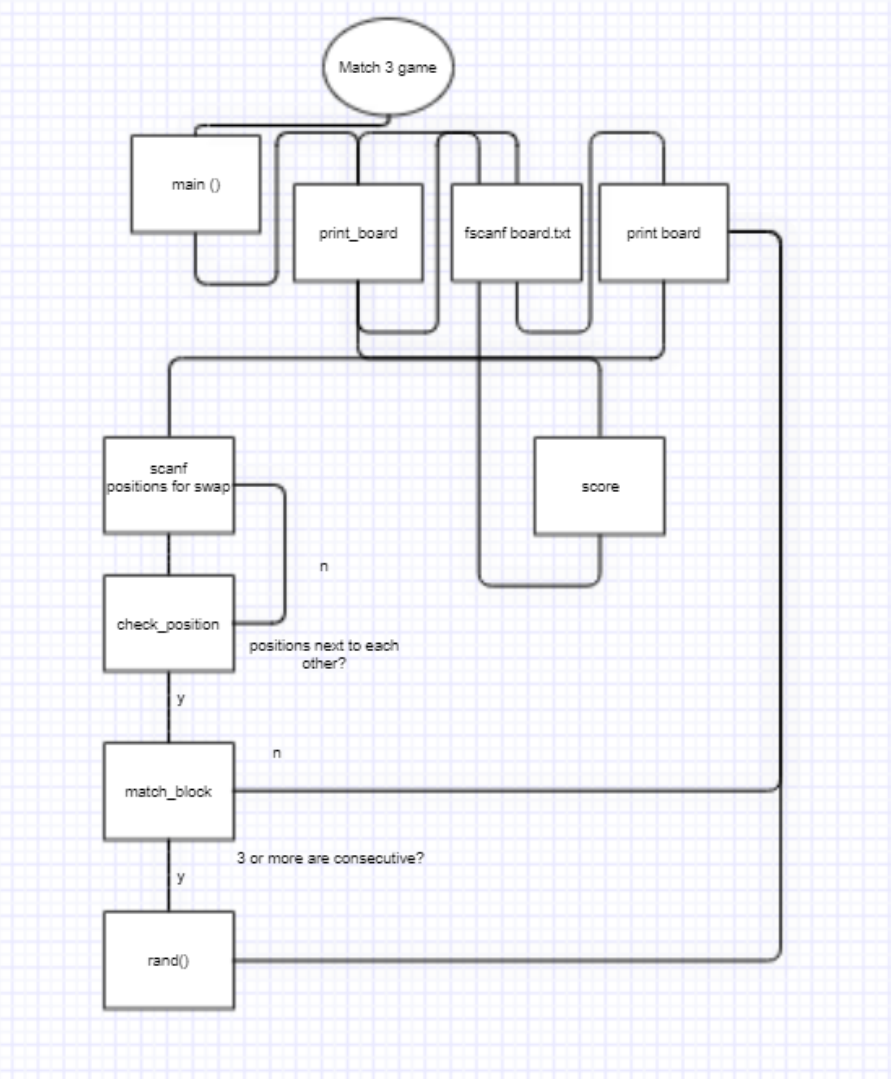
출력부: 이때 바뀐 부분을 빨간색으로 처리하고 3개 이상의 블록이 연속할 경우 다시 빨간색으로 표시하고 원소를 " (스페이스)"로 표시해 삭제 처리한다.

2. 알고리즘

본 프로그램 작성을 위해 Pseudo코드 형태로 나타내면 다음과 같다.

Pseudo-algorithm for match 3 game
//변수는 선언된 것으로 가정 1. Include essential headers <stdio.h>, <stlib.h>, <windows.h> and <time.h> 2. Print starting screen 3. Print a 6 × 6 board, ask users to input the positions 4. Print two positions in red, wait 1 sec 5. Check and print consecutive numbers (3 or bigger) in red and wait 1 sec 6. Print " " for the consecutive numbers, wait 1 sec 7. Put new numbers into the deleted positions randomly 8. For score, Use score+=arrb[i][j]

위의 의사 알고리즘을 flow chart를 통해 표현하면 아래와 같다.



3. 프로그램 구조 및 설명

a. 시작화면 실행

b. fscanf를 통해 board.txt에서 배열을 입력 받음

파일의 원소를 한 개씩 읽어 이중 for문과 arr[i][j]식을 이용해서 배열에 넣는다. 이 배열을 처음에 출력하도록 한다. 만약 board.txt라는 식이 존재하지 않는다면 적절한 메시지를 출력하도록 한다.

c. 블록의 위치를 바꿈

사용자로부터 바꾸고 싶은 블록의 위치를 입력 받는다. Check_positon함수를 불러와 입력된 위치가 배열의 범위 (0,1,2,3,4,5)가 아닌 수가 입력되면 'out of range' 라는 메시지를 출력하고 배열의 위치가 상하좌우가 아니거나 위치가 같다면 'they are not adjacent' 메시지를 출력하고 위치를 다시 입력 받는다. 만약 유효한 수라면 함수에서 0을 return 하고 다음 단계를 진행한다.

d. 바뀐 곳 빨간색으로 표현

유효한 입력이라면 배열을 빨간색으로 1초동안 표시한다. Set_color_red() 함수와 reset_color() 함수를 불러와서 바뀐 블록 두개를 빨간색으로 표시한다.

e. 연속하는 수 처리

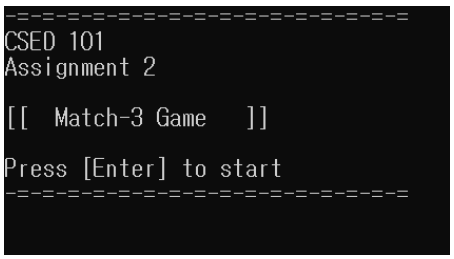
연속하는 수를 처리하기 위해서 arrb라는 배열을 만든다. Match_block 함수에 arr과 초기화 값이 모두 0인 arrb를 함께 보내고 만약 arr에서 3개 이상 연속하는 블록이 존재한다면 arr의 값을 arrb에 모두 복사한다. 이후 print_board 함수로 돌아와서 arrb에 수가 존재하면 그 부분은 발간색으로 표시하고 1초뒤 공백으로 처리한다. 이후 arr에 1~5까지의 수를 랜덤 배정한후 출력한다.

f. 점수 처리

arrb함수의 원소를 누적해서 다 더하면 점수가 된다. Score 함수에 arrb 값과 이전 단계에서의 점수를 보내고 이전 단계에서의 점수에 현재 arrb값을 더한다.

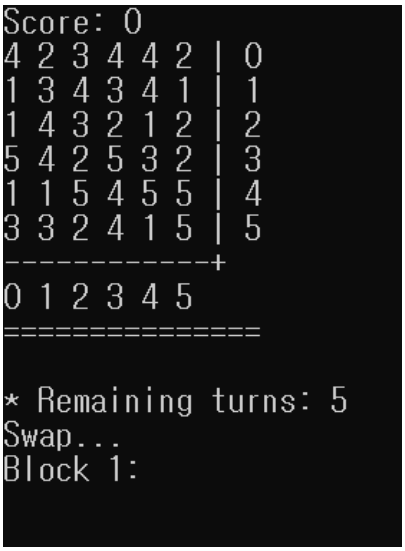
4. 프로그램 실행방법 및 예제

1.



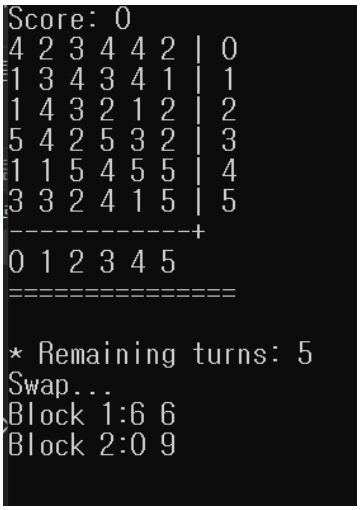
시작화면이다. Enter를 누르면 다음 화면으로 넘어간다.

2.

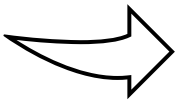


Board.txt 파일로부터 배열을 읽어와서 그대로 프린트 한다. 이때 상단에 점수가 나오며 우측과 하단에 배열의 위치를 띄워준다. 남은 턴의 수를 프린트하고 사용자로부터 바꾸고 싶은 위치의 배열을 입력 받는다.

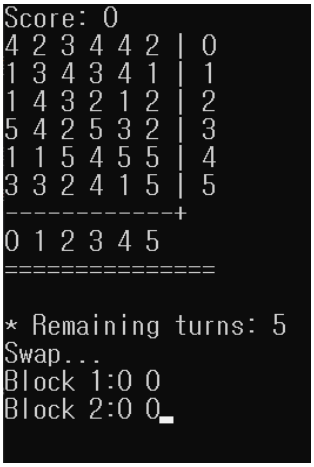
3.



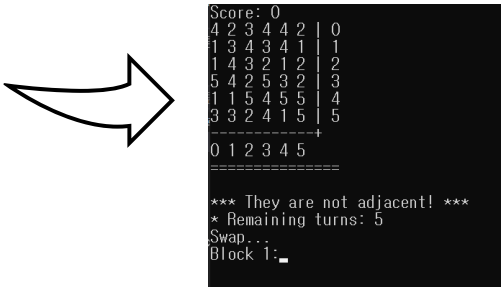
입력이 배열 범위를 벗어난 경우 실행 예시이다.



4.



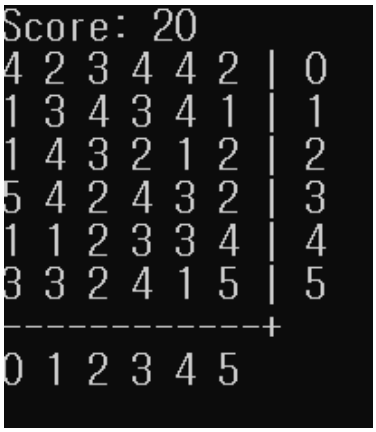
위치를 바꾸고 싶은 블록이 상하좌우에 위치하지 않을 경우 출력화면이다



5.

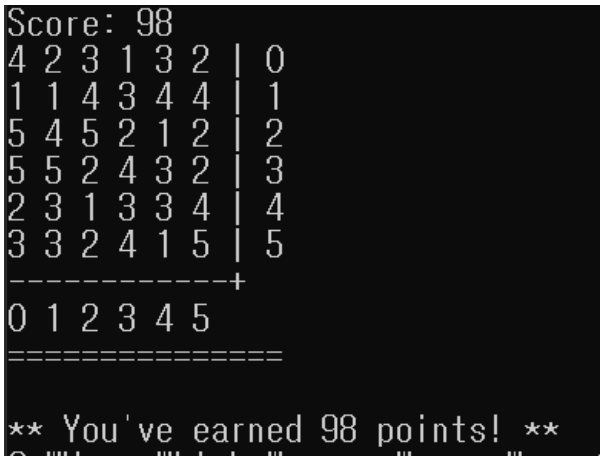
입력이 유효할 경우 바꾼 블록 두개를 빨간색으로 표시하고 블록 검사를 통해 3개 이상 연속하는 블록을 1초후 빨간색으로 표시한다. 1초 후 연속하는 블록을 삭제한다.

6.



지워진 블록 위치에 새로운 값을 랜덤하게 배정하고 1초후 출력한다. 이후 while구문을 사용해 다시 겹치는 수가 있는지 검사하고 있다면 이전 단계인 5번을 반복한다. 겹치는 수가 없다면 새로운 입력을 받고 이때 turn수는 줄어든다.

7.



만약 turn이 0 이 된다면 보드를 다시 출력하고 최종 점수를 알려준 후 프로그램을 종료한다.

5. 토론

- 재귀함수의 사용이 메모리를 차지해 에러를 일으킬 수 있다는 것을 배웠기 때문에 코드를 짜면서 재귀함수가 사용하지 않았다.
- 처음에 3개 이상의 블록이 연속할 경우 어떻게 처리해야 할지 고민했고, 겹치는 수를 모두 7로 바꾸어서 처리하는 방법 등 여러가지를 생각했지만 처리가 쉽지 않았다. 그러던 중 다른 배열 arrb를 만들어 연속할 경우 arrb에 arr 원소를 복사하여 처리하면 점수 합과, 연속하는 수를 동시에 처리할 수 있을 것이라는 생각이 들어 arrb 배열을 사용하였다.

6. 결론

- 이번 어싸인을 통해 2차원 배열을 처리하는 방법을 알게 되었다. 배열의 주소를 받아오는 것, 배열의 원소 값에 접근하는 방법을 배워서 자료를 처리하는데 도움이 될 것이다.
- 처음 board.txt 파일을 읽어오고 파일의 값을 배열에 대입할 때 읽음과 동시에 프린트해야 할 지 아니면 배열에 저장한 후 프린트해야 할 지 고민되었다. 하지만 이후 arr 배열의 처리를 위해 저장하는 것이 좋겠다고 생각되어 저장을 먼저 했으며 성공적인 판단이었다.
- 배열을 처리할 때 for문을 사용하는 방법을 익히게 되었다. 이중 for 문이 사용되었다.

7. 개선 방향

- 처음 코드를 짰을 때 반복되는 구문이 매우 많았다. 반복을 줄이기 위해 코트 수정을 많이 해야 했고, 특정 함수내에서 똑같은 함수를 불러내는 재귀 함수의 사용이 있어서 그 부분을 해결하기 위해 노력했다.

- 포인터 변수의 사용이 있었는데, 수업 노트를 복습하고 수업을 다시 들으니 배열을 포인터로 어떻게 처리하는지 알게 되어 6 ×6 배열이 아닌 더 큰 범위의 배열도 처리할 수 있을 것 같다.