CSED101. Programming & Problem Solving
Spring, 2021

Programming Assignment #2
(75 Points)

차동민 (cardongmin@postech.ac.kr)

■ **제출 마감일: 2021.05.04 23:59**

■ **개발 환경: Windows Visual Studio 2019**

■ **제출물**

- **C 소스 코드(assn2.c)**
  - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 **주석**을 붙여주세요.
- **보고서 파일** (.docx, .hwp 또는 .pdf; assn2.docx, assn2.hwp 또는 assn2.pdf)
  - 보고서는 AssnReadMe.pdf 를 참조하시면 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 소스코드와 보고서 파일을 PLMS를 이용하여 제출해 주세요.

■ **주의사항**

- 컴파일이나 실행이 되지 않는 과제는 0점으로 채점됩니다.
- 제출 기한보다 하루 늦게 제출된 과제는 최종 20%, 이틀 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: 빙고 게임

[문제]

2차원 배열을 이용하여 빙고 게임을 구현해 본다.

[목적]

- 2차원 배열의 선언과 사용을 익힌다.
- 함수에서 2차원 배열을 매개변수로 사용하는 방법을 익힌다.
- 랜덤으로 숫자를 발생시키는 난수 함수의 사용을 익힌다.
- 파일 입출력 방법을 익힌다.

[주의사항]

(1) 이 과제는 2차원 배열을 선언하고 사용하기 위한 과제이므로, 플레이어와 컴퓨터의 빙고판은 반드시 각각 2차원 배열로 선언 후 사용해야 한다. (마지막 페이지의 힌트를 참고할 것)

(2) 전역 변수, goto 문, string 관련 함수, 구조체는 사용할 수 없으며, 포인터의 경우 수업시간에 다룬 내용에 한해서 사용이 가능하다.

(3) 모든 기능을 main 함수에서 구현한 경우 감점 처리한다. 기능적으로 독립됐거나 반복적으로 사용되는 기능은 사용자 함수를 정의해서 구현하여야 한다.

(4) 프로그램 구현 시, main() 함수를 호출하여 사용하지 않는다. 즉, 소스 코드 내에 main(); 이라고 호출하지 않는다.

(5) 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.
(빙고판을 그리기 위해 +, -, ¦, # 문자를 사용, 공백의 개수는 자유)
지정된 빙고판 크기보다 더 적게 그려지거나 더 많게 그려지는 경우는 감점 사유가 된다.

(6) 프로그램에서 랜덤 시드는 프로그램 시작 시 main() 에서 srand(time(NULL)); 함수를 한번만 호출하도록 하여 한번만 초기화 한다.

[함수 요구 사항]

이 문제를 해결하기 위해 아래 기능에 해당하는 사용자 정의 함수들을 구현하여 사용하도록 한다.
아래 기능 외의 필요한 함수를 정의해서 사용할 수 있다.

#define MAX_SIZE 5 : 빙고 보드가 최대로 가질 수 있는 크기를 상수로 정의

- **void printBoard(char user_board[][MAX_SIZE], char comp_board[][MAX_SIZE], …)**: 현재의 빙고 보드를 출력
- **int checkBingo(char board[][MAX_SIZE], …)**: 빙고 보드에서 가로, 세로, 대각선 등 빙고가 존재하는지를 판단하여 있다면 1, 없다면 0을 반환

[설명 및 실행 예시]

이 과제에서 빙고 게임은 2인용 게임으로 플레이어가 컴퓨터와 대결하는 방식으로 진행된다. 각자 가지고 있는 N x N 크기의 빙고판에 채워져 있는 문자를 서로 번갈아 가면서 부른다. 문자를 부를 때마다 자신의 빙고판에 해당 문자가 있는 부분에 표시를 하여, 먼저 한 줄(가로, 세로, 대각선 중 하나)을 완성하게 되는 쪽이 승리하게 된다.

1. 빙고판 생성 방법 선택

빙고 게임을 위해 빙고판 생성시, 컴퓨터가 랜덤하게 생성하는 방법과 파일 입력의 두 가지 방법으로 만들 수 있어야 한다. 아래의 예시처럼 프로그램을 시작할 때, 그 방법을 선택한다. 1(random), 2(read from file) 외의 입력에 대해서는 아래의 예시처럼 올바른 입력이 들어올 때까지 반복하여 입력 받는다.

(아래 실행 예시의 빨간색 밑줄은 사용자 입력을 나타냄)

```
Random or read from file? (1: Random, 2: Read) 3
Random or read from file? (1: Random, 2: Read)
```

(1) 랜덤(random)

사용자가 1(random)를 선택하면, 먼저 빙고판의 크기를 입력 받는다. 빙고판의 크기는 정사각형이며, 최소 3x3에서 5x5까지의 빙고판을 정할 수 있어야 한다. 3~5 사이의 값이 아닌 잘못된 값이 입력된 경우 아래의 예시처럼 올바른 입력이 들어올 때까지 반복한다.

```
Random or read from file? (1: Random, 2: Read): 1

Size of Bingo board? (3~5) 1
Size of Bingo board? (3~5) 6
Size of Bingo board? (3~5)
```

다음으로, 유효한 빙고판의 크기를 입력 받게 되면 게임을 위해 2개의 빙고판(플레이어, 컴퓨터)을 생성한다. 각 빙고판은 아래의 기준을 사용하여 각 칸에 문자를 채워 넣는다.

가. 3x3의 경우, 알파벳 A부터 I까지의 9개의 문자를 사용
나. 4x4의 경우, 알파벳 A부터 P까지의 16개의 문자를 사용
다. 5x5의 경우, 알파벳 A부터 Y까지의 25개의 문자를 사용

하나의 빙고판의 각 칸은 주어진 문자를 모두 사용하여 값을 채워야 한다. 예를 들어 5x5의 경우, 알파벳 A부터 Y까지의 25개의 문자가 한번씩 모두 들어가야 한다. (중복이나 사용하지 않는 문자는 있을 수 없다.)

2개의 빙고판은 게임을 시작할 때마다 랜덤하게 위 조건에 맞춰서 값이 채워지면 된다.

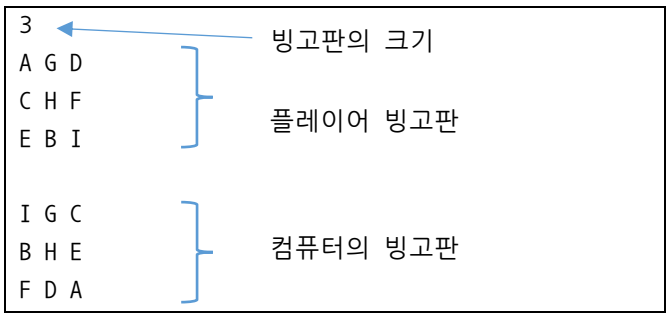
아래는 사용자가 빙고판의 크기를 5로 입력한 경우의 실행 예시로 5x5의 빙고판 2개를 랜덤하게 생성한 후, 아래처럼 출력한 후 게임을 위해 플레이어의 입력을 기다린다.

```
Random or read from file? (1: Random, 2: Read): 1

Size of Bingo board? (3~5) 5

+-+-+-+-+-+    +-+-+-+-+-+
|D|C|P|Y|V|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|W|J|R|I|H|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|O|L|Q|K|B|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|T|A|F|U|N|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|E|X|G|S|M|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+

Your choice:
```

위 예시에서 왼쪽이 플레이어의 빙고판이고 오른쪽이 컴퓨터의 빙고판에 해당한다. 플레이어의 빙고판은 위의 예시처럼 알파벳이 어디에 배치되어 있는지 표시하고, 컴퓨터의 빙고판의 내용은 표시되지 않도록 한다.

(2) 파일 입력(read from file)

사용자가 2(read from file)를 선택하면, 미리 만들어진 텍스트 파일 board.txt 를 읽어서 빙고판을 만들 수 있어야 한다. board.txt 의 구성은 아래와 같다.



- 첫번째 줄의 숫자는 빙고판의 크기에 해당
- 다음줄부터 빙고판의 크기에 맞춰서 각 빙고판을 채울 문자가 공백과 줄바꿈으로 구분되어 있다.
- 위 예시의 board.txt 를 읽어서 빙고판을 만들게 되면 아래와 같이 출력된다.

```
Random or read from file? (1: Random, 2: Read): 2

+-+-+-+     +-+-+-+
|A|G|D|     | | | |
+-+-+-+     +-+-+-+
|C|H|F|     | | | |
+-+-+-+     +-+-+-+
|E|B|I|     | | | |
+-+-+-+     +-+-+-+

Your choice:
```

- 실행 시, 파일이 없는 경우는 고려하지 않는다.
- 파일의 내용이 틀린 경우는 없다고 가정한다. 빙고판의 크기가 3~5의 범위를 벗어나는 경우, 각 보드에 중복된 문자가 존재하는 경우 등은 없다고 가정한다.

2. 게임 진행

(1) 게임 초기 화면

빙고판을 파일 또는 랜덤하게 생성하게 되면, 생성된 빙고판이 출력되며 게임이 시작된다.
아래 실행 예시는 랜덤하게 빙고판을 생성한 경우의 예시로, 생성된 빙고판이 출력된 후 게임을 진행하기 위하여 플레이어의 입력을 기다리는 화면이다. (빙고판 5x5에 대한 실행 예시에 해당)

```
Random or read from file? (1: Random, 2: Read): 1

Size of Bingo board? (3~5) 5

+-+-+-+-+-+     +-+-+-+-+-+
|D|C|P|Y|V|     | | | | | |
+-+-+-+-+-+     +-+-+-+-+-+
|W|J|R|I|H|     | | | | | |
+-+-+-+-+-+     +-+-+-+-+-+
|O|L|Q|K|B|     | | | | | |
+-+-+-+-+-+     +-+-+-+-+-+
|T|A|F|U|N|     | | | | | |
+-+-+-+-+-+     +-+-+-+-+-+
|E|X|G|S|M|     | | | | | |
+-+-+-+-+-+     +-+-+-+-+-+

Your choice:
```

(2) 플레이어 선택

이 때, 플레이어가 먼저 알파벳을 입력하여 게임을 시작하도록 한다. 그 다음부터는 컴퓨터와 번갈아 가면서 알파벳을 선택하도록 한다. 아래의 예시는 컴퓨터가 A를 선택하여 입력한 경우로, 플레이어의 빙고판의 해당 문자는 '#'으로 표시, 컴퓨터의 빙고판은 해당 문자의 칸을 '#'으로 표시한다. (아래 실행 예시의 빨간색 #은 설명을 위한 것으로 빨간색으로 출력 되도록 구현할 필요는 없다.)

```
Your choice: A

+-+-+-+-+-+    +-+-+-+-+-+
|D|C|P|Y|V|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|W|J|R|I|H|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|O|L|Q|K|B|    | | |#| | |
+-+-+-+-+-+    +-+-+-+-+-+
|T|#|F|U|N|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|E|X|G|S|M|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+

Computer's choice:
```

(3) 컴퓨터 선택

다음으로 컴퓨터가 자신의 순서에서 알파벳을 선택한다. 이 때, 위 화면이 출력되고 1초의 지연 후(1초 지연 기능은 마지막 페이지의 힌트를 참고할 것)에 컴퓨터가 선택한 알파벳을 출력 후, 빙고판의 현황을 출력하도록 한다.

컴퓨터가 자신의 순서에서 알파벳을 선택할 경우 컴퓨터는 빙고판에 있는 문자 중에서 지금까지 선택되지 않은 문자들 중에서 랜덤하게 선택하도록 구현을 해야 한다. 아래는 컴퓨터가 E를 선택한 예시이다.

```
+-+-+-+-+-+    +-+-+-+-+-+
|T|#|F|U|N|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|E|X|G|S|M|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+

Computer's choice: E

+-+-+-+-+-+    +-+-+-+-+-+
|D|C|P|Y|V|    | | | | |#|
+-+-+-+-+-+    +-+-+-+-+-+
|W|J|R|I|H|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|O|L|Q|K|B|    | | |#| | |
+-+-+-+-+-+    +-+-+-+-+-+
|T|#|F|U|N|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|#|X|G|S|M|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+

Your choice:
```

(4) 게임 종료

플레이어와 컴퓨터가 번갈아가면서 자신의 순서에 알파벳을 선택하며 게임을 진행하다가, 둘 중 가로, 세로, 대각선의 빙고가 하나라도 완성이 되면 게임이 종료된다.

```
Your choice: Q

+-+-+-+-+-+    +-+-+-+-+-+
|D|C|P|Y|V|    | | | | |#|
+-+-+-+-+-+    +-+-+-+-+-+
|W|J|R|I|H|    | | | | |#|
+-+-+-+-+-+    +-+-+-+-+-+
|O|L|#|K|B|    | | |#| | |
+-+-+-+-+-+    +-+-+-+-+-+
|T|#|F|U|N|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|#|X|G|S|M|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+

Computer's choice: V

+-+-+-+-+-+    +-+-+-+-+-+
|D|C|P|Y|#|    | | |#| |#|
+-+-+-+-+-+    +-+-+-+-+-+
|W|J|R|I|H|    | | | | |#|
+-+-+-+-+-+    +-+-+-+-+-+
|O|L|#|K|B|    | | |#| | |
+-+-+-+-+-+    +-+-+-+-+-+
|T|#|F|U|N|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|#|X|G|S|M|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+

Your choice: I

+-+-+-+-+-+    +-+-+-+-+-+
|D|C|P|Y|#|    | | |#| |#|
+-+-+-+-+-+    +-+-+-+-+-+
|W|J|R|#|H|    | | | | |#|
+-+-+-+-+-+    +-+-+-+-+-+
|O|L|#|K|B|    | | |#| | |
+-+-+-+-+-+    +-+-+-+-+-+
|T|#|F|U|N|    | | | | | |
+-+-+-+-+-+    +-+-+-+-+-+
|#|X|G|S|M|    | | | | |#|
+-+-+-+-+-+    +-+-+-+-+-+

You Win!
계속하려면 아무 키나 누르십시오...
```

게임 종료 전, 게임의 결과를 출력 후 프로그램을 종료하도록 한다.
- 플레이어가 먼저 빙고를 완성한 경우에는 앞의 예시처럼 ‘You Win!’을,
- 컴퓨터가 먼저 빙고를 완성한 경우에는 ‘Computer Wins!’을,
- 동시에 빙고를 완성한 경우에는 ‘Draw’를

출력 후 프로그램을 종료하면 된다.

(5) 예외 처리

플레이어가 알파벳을 선택하는 부분에서, 만약 잘못된 입력을 받는 경우, 아래의 예시처럼 올바른 입력이 들어올 때까지 반복한다. 잘못된 값이 들어오는 경우는 다음과 같다.
- 현재 빙고 판에서 나올 수 없는 알파벳이 입력되는 경우 (ex: 4x4 크기의 보드판에서는 A~P까지

의 알파벳만 사용 가능한데, 그 외의 알파벳 (Q, X 등)이 입력되는 경우)
- 앞에서 이미 선택한 알파벳이 입력되는 경우

```
+-+-+-+-+    +-+-+-+-+
|A|H|F|E|    | | |#| |
+-+-+-+-+    +-+-+-+-+
|P|I|B|D|    | | | | |
+-+-+-+-+    +-+-+-+-+
|K|J|#|C|    | | |#| |
+-+-+-+-+    +-+-+-+-+
|L|#|N|M|    | | | | |
+-+-+-+-+    +-+-+-+-+

Your choice: S
Your choice: G
Your choice:
```

3. 파일 출력(게임 기록)

빙고 게임에서 매 턴마다 어떤 알파벳들을 선택했는지, 각 빙고 보드의 상황이 어떻게 되어가는지를 파일로 출력하도록 한다. 출력을 저장할 파일 이름은 result.txt 로 한다.
다음과 같이 출력하며, 첨부한 'result.txt'를 참조하여 동일하게 출력하도록 한다. 출력파일에는 '현재 빙고판', '선택된 알파벳', '게임 결과' 등이 기록된다. 파일에 출력할 때는 컴퓨터의 빙고판도 모두 가려지는 것이 없이 출력되도록 한다. (단, 플레이어의 입력의 경우, 유효한 입력에 대해서만 기록하면 된다.)

```
B E P D      O M C G
G I N F      F J N I
K M H O      B H K L
A L C J      P D A E

User's choice: E
B # P D      O M C G
G I N F      F J N I
K M H O      B H K L
A L C J      P D A #

Computer's choice: N
B # P D      O M C G
G I # F      F J # I
K M H O      B H K L
A L C J      P D A #

User's choice: G
B # P D      O M C #
# I # F      F J # I
K M H O      B H K L
A L C J      P D A #

Computer's choice: C
B # P D      O M # #
# I # F      F J # I
K M H O      B H K L
A L # J      P D A #

User's choice: A
B # P D      O M # #
```

```
# I # F      F J # I
K M H O      B H K L
# L # J      P D # #

Computer's choice: M
B # P D      O # # #
# I # F      F J # I
K # H O      B H K L
# L # J      P D # #

User's choice: B
# # P D      O # # #
# I # F      F J # I
K # H O      # H K L
# L # J      P D # #

Computer's choice: O
# # P D      # # # #
# I # F      F J # I
K # H #      # H K L
# L # J      P D # #

Computer Wins!
```

[힌트]

- ***배열 선언 및 사용***

  (1) 배열의 크기를 선언할 때 변수를 사용하면 안됩니다(int array[i][i];). 충분한 크기로 선언한 뒤(int array[4][4];), 필요한 부분만 사용하면 됩니다.

  (2) 플레이어와 컴퓨터의 빙고판은 각각 2차원 배열로 정의하여 사용하도록 합니다. 이 배열 외에도 사용한 알파벳을 기억하기 위한 1차원 배열을 선언해서 사용하면 좋습니다.

  (3) 배열을 parameter 로 다른 함수에 넘겨주었을 경우, 그 함수에서 배열의 값을 바꾸면 main 함수에서도 바뀐 상태가 유지됩니다. 이는 배열이 포인터 형태를 띄고 있기 때문입니다.

```c
#include <stdio.h>
void foo(int x[]);
int main()
{
    int a[10];
    a[0] = 0;
    foo(a);
    printf("%d\n", a[0]); //prints 1
    return 0;
}
void foo(int x[])
{
    x[0] = 1;
    return;
}
```

- ***지연 효과 주기***

지연 효과는 Sleep()함수를 통해 구현 가능합니다. 해당 함수는 인자로 지연시간을 millisecond 단위로 받으며, 사용하기 위해서는<Windows.h> 헤더파일을 포함해야 합니다. 사용 예는 다음과 같습니다.

```
#include <stdio.h>
#include <Windows.h>
int main()
{
    printf("첫 번째 메시지 입니다\n");
    Sleep(1000);
    printf("이 메시지는 1초뒤에 출력됩니다\n");
    return 0;
}
```

참고) 리눅스 또는 맥에서 프로그래밍을 하는 경우

```
#include <stdio.h>
#include <unistd.h>  // sleep 함수 사용을 위해
int main()
{
    printf("첫 번째 메시지 입니다\n");
    sleep(1);   // 매개변수는 초 단위. 1초 지연
    printf("이 메시지는 1초뒤에 출력됩니다\n");
    return 0;
}
```

- ***어싸인 팁***

(1) 프로그램 작성 중 무한 루프에 의해 프로그램이 끝나지 않을 경우, Ctrl+C를 누르세요.