

# CSED101. Programming & Problem solving

## Fall, 2020

### Programming Assignment #1 (75 points)

황제영(capricorn116@postech.ac.kr)

- **Due:** 2020.05.11 23:59
- **Development Environment:** Windows Visual Studio 2019
- **제출물**
  - **C Code files (assn1.c)**
    - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것.
  - **보고서 파일** (assn1.docx, assn1.hwp 또는 assn1.pdf)
    - AssnReadMe.pdf 를 참조하여 작성할 것.
    - **명예서약(Honor code):** 표지에 다음의 내용을 포함한다. "나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다." 보고서 표지에 명예서약이 없는 경우는 과제를 제출하지 않은 것으로 처리한다.
    - 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.
- **주의사항**
  - 문제의 요구사항을 반드시 지킬 것.
  - 모든 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.
  - 문제에 제시되어 있는 파일이름으로 제출 할 것. 그 외의 다른 이름으로 제출하면 감점 또는 0점 처리된다.
  - 컴파일 & 실행이 안되면 무조건 0점 처리된다.
  - 하루 late시 20%가 감점되며, 3일 이상 지나면 받지 않는다. (0점 처리)
  - 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 'POSTECH 전자컴퓨터공학부 부정행위 정의'를 따른다. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 disciplinary.pdf를 참조할 것.)
  - 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.

## ■ Problem: 미니 야구 게임

### (목적)

이번 과제를 통하여 조건문, 반복문, 사용자 정의 함수 및 라이브러리 함수 사용법을 익힌다.

### (주의사항)

- 이번 과제는 함수를 정의하고 사용하는 방법을 익히는 문제이므로 main함수에 모든 기능을 구현한 경우 감점 처리 함.
- 과제 설명에서 반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인한 후, 구현하도록 한다. 이 때, 설명에서 지정한 사용자 정의 함수의 매개변수의 개수와 자료형, 함수 이름, 리턴 자료형 등 자유롭게 변경이 가능하다. 그러나 동일한 기능을 하는 함수는 반드시 있어야 하며, 변경 시 무엇을 어떻게 변경해서 구현했는지 보고서에 포함하도록 한다. 이외의 필요한 함수를 정의해서 사용할 수 있다.
- 프로그램 구현 시, main() 함수를 호출하여 사용하지 않는다. 즉, 소스 코드 내에 main(); 이라고 호출하지 않는다.
- 전역 변수, 배열 및 goto 문은 사용할 수 없으며, 포인터의 경우 수업시간에 다룬 내용에 한해서 사용이 가능하다.
- 사용자로부터 메뉴 선택, 스윙, 거르기를 위해 숫자를 입력 받을 때, 정수 외의 입력에 대해서는 고려하지 않아도 된다.
- 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.

### (설명)

본 게임은 야구의 모든 룰을 적용하지 않은 '미니 야구 게임'이다. 컴퓨터와 플레이어가 플레이를 하게 된다. 총 3회로 게임을 진행하며, 플레이어는 타자가 되어 컴퓨터가 던지는 공을 치거나 거를 수 있으며 3회가 종료되고 특정 점수 이상을 획득하면 플레이어가 이기는 게임이다.

다음으로 '미니 야구 게임'의 규칙은 다음과 같다.

- 컴퓨터는 일정한 확률로 스트라이크와 볼인 공을 던진다.
- 플레이어(타자)는 컴퓨터가 던진 공을 치기 위해 스윙을 하거나 공을 거를 수 있다. 아래의 기준으로 '스윙' 혹은 '거르기'를 선택할 수 있다.
  - 스윙(1), 거르기(2)
- 플레이어가 스윙을 하여 안타를 치거나 볼넷의 경우, 타자는 진출한다. 안타를 친 경우, 특정한 확률로 1루타, 2루타, 3루타, 홈런 그리고 플라이 아웃이 발생하며, 각각 안타에 따라 주자들이 이동(진루)한다.
  - 이동 및 점수 득점 규칙
    - ① 1루타의 경우 모든 주자가 하나의 베이스만큼 이동한다.
      - A. 1루에 주자가 존재하면 2루로 진루(이동)한다.
      - B. 2루에 주자가 존재하면 3루로 진루(이동)한다.
      - C. 3루에 주자가 존재하면 홈베이스로 진루(이동)한다.
    - ② 2루타의 경우 모든 주자는 두 개의 베이스만큼 이동한다.
    - ③ 3루타의 경우 모든 주자는 세 개의 베이스만큼 이동한다.
    - ④ 홈런의 경우 모든 주자(타자 포함)가 홈 베이스로 들어온다.

- ⑤ 볼넷의 경우 타자가 1루로 진출한다.
  - A. 1, 2, 3루에 모두 주자가 있는 상황에서 타자가 볼넷을 얻은 경우라면, 모든 주자가 하나의 베이스만큼 이동한다. 타자는 1루에, 1루 주자는 2루에, 2루 주자는 3루에, 3루 주자는 홈으로 들어와 득점한다.
  - B. 진루하고자 하는 베이스에 주자가 없는 경우라면, 그 베이스 다음 주자들은 이동하지 않는다. (ex. 주자가 2, 3루에 있는 경우, 현재 타자가 볼넷으로 출루하게 되면 2, 3루의 주자는 다음 베이스로 이동하지 않는다.)
- ⑥ 한 명의 주자가 1, 2, 3루를 순서대로 거쳐 홈베이스로 들어온 경우, 1점 득점한다. 즉, 홈베이스로 들어온 주자의 수만큼 점수를 획득하게 된다.
- 플레이어가 스윙을 하여 공을 쳐서 플라이 아웃 또는 삼진(쓰리 스트라이크)이 된 경우, 그리고 공을 걸렀을 때 스트라이크가 되어 삼진이 되면 타자가 아웃된다.
  - 아웃 규칙
    - 플레이어가 컴퓨터가 던진 공을 타격하면 일정한 확률로 플라이 아웃 처리가 된다.
    - 플레이어가 삼진(3 스트라이크)을 당하면 아웃 처리가 된다.
    - 주자에 의한 아웃은 존재하지 않으며, 스윙 후 플라이 아웃, 쓰리 스트라이크에 의한 아웃만 존재한다.
- 쓰리 아웃이 되면 한 회가 종료가 되며, 3회가 모두 끝나면 획득한 점수를 기준으로 승·패가 결정된다.
- 3회가 종료되기 전에 획득한 점수가 기준 점수를 넘을 시, 해당 이닝이 끝나면 게임 종료와 함께 승리 메시지를 출력한다.

## I. 초기 선택 메뉴

프로그램 시작 시, 그림 1과 같이 3가지 선택사항이 있는 메뉴가 출력되며, 사용자로부터 3가지 선택사항 중 한 가지를 입력 받도록 합니다.

- 초기 선택 메뉴는 main() 함수 내에서 아래의 사용자 정의 함수를 호출하여 출력하도록 하며, 반드시 switch-case문을 사용하여 메뉴 선택을 하도록 합니다.
  - **print\_menu():** 초기 메뉴 화면을 출력하고, 사용자로부터 메뉴 번호를 입력 받아 그 번호를 반환합니다.
- 메뉴 아래 위의 구분선을 출력하기 위해 반드시 아래의 함수를 정의하고 사용해야 합니다.
  - **draw\_line():** 50개의 "="을 출력합니다. (for 문을 사용하여 출력할 것)

```
=====
1. 게임 시작
2. 게임 설명
3. 게임 종료
=====
선택하고자 하는 메뉴를 입력하세요: .
```

그림 1. 메뉴 선택 화면

메뉴 입력 시, 숫자만 입력된다고 가정합니다. 숫자 1, 2, 3 외의 값을 입력할 경우, 그림 2와 같이 화면을 지우고 “메뉴를 다시 선택해주세요.”라는 메시지를 출력하고 초기 선택 메뉴(그림 1)를 다시 화면에 출력하도록 합니다. 화면을 지우는 방법은 Tips- 화면 지우기(마지막 쪽)을 참고하여 구현하도록 합니다.

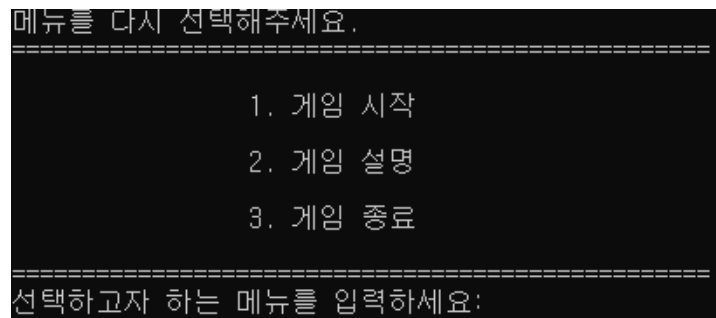


그림 2. 메뉴 선택 오류 및 재선택 화면

## II. 게임 시작

그림 1의 메뉴에서 1을 선택하면, `start_game()` 함수가 호출되며 게임이 시작된다. 모든 게임은 `start_game()` 함수 내에서 진행되어야 한다. 아래의 설명에서 반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인 후 구현하도록 한다. 설명에서 구현하라고 한 함수 외에 필요한 함수를 정의하고 사용할 수 있다.

### ➤ 사용자 정의 함수

- **start\_game():** 게임을 시작하고 게임을 시작한다는 안내 문구를 출력하며, 각각의 이닝을 진행하는 `start_inning()` 함수를 반복문을 통해 호출(총 3회)한다.

### 1. 게임 최초 화면

게임이 시작되면 제일 먼저 화면을 지운 후, 그림 3과 같이 게임을 시작한다는 알림 메시지와 함께 `start_game()` 함수 내에서 `start_inning()` 함수를 호출하여 각각의 이닝을 진행합니다. `start_inning()` 함수를 호출하여 화면에는 진행되고 있는 이닝, 점수판, 볼 · 스트라이크 · 아웃 카운트, 주자 상황 그리고 스윙을 할지 공을 거를지 선택하는 입력창을 출력합니다. 이 때, 점수판, 카운트 상황, 주자 상황은 각각 `print_score()`, `print_count()`, `print_base()` 함수를 `start_inning()` 함수내에서 호출하여 화면에 출력하며, 사용자로부터 스윙 혹은 거르기를 선택할 것인지를 입력 받습니다.

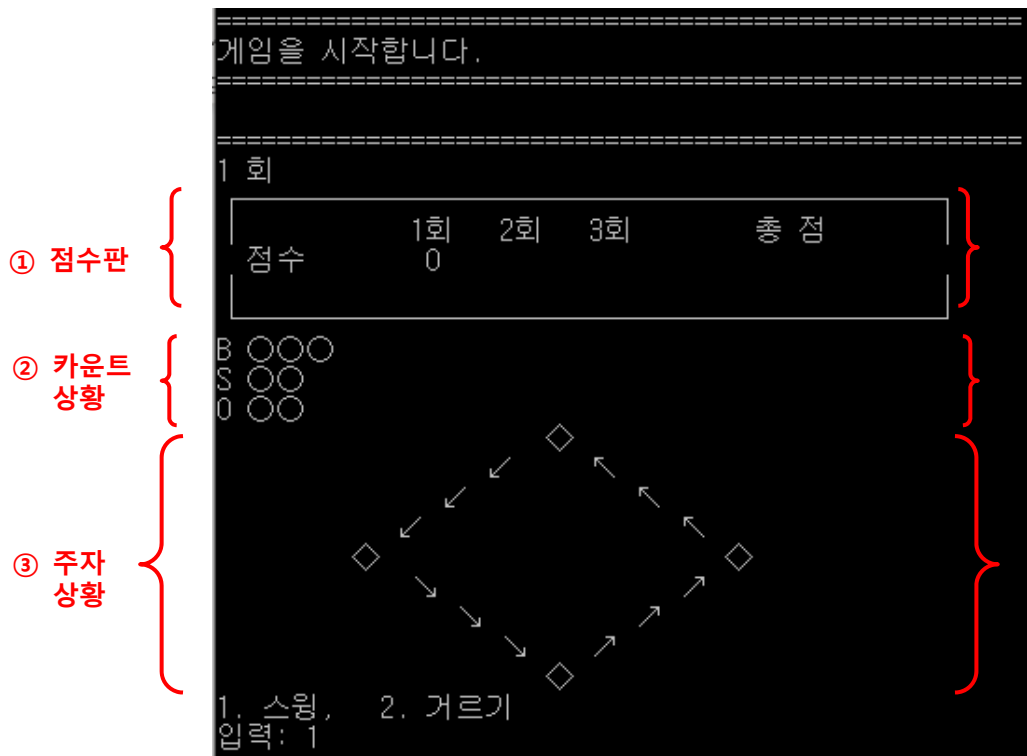


그림 3. 게임 최초 화면

#### ➤ 사용자 정의 함수

- **int start\_inning():** 하나의 이닝을 진행하는 함수로, 현재 진행되는 이닝과, 1회에서 얻은 점수, 2회에서 얻은 점수, 3회에서 얻은 점수를 매개변수로 전달받아 각 이닝에서 획득한 점수를 반환합니다.
- **void print\_score():** '①점수판'을 출력하는 함수로, 현재 진행되는 이닝과 1회에서 얻은 점수, 2회에서 얻은 점수, 3회에서 얻은 점수, 총점을 매개변수로 전달받아 각각의 이닝에서 얻은 점수를 출력해주며 3회가 모두 끝나면 총점까지 출력해줍니다. 점수판을 감싸고 있는 사각 박스는 '┌─┐└─┘' 문자 ('ㅂ' 키 입력후 한자키를 눌러서 해당 문자 선택)를 사용하여 적절히 출력합니다.
- **void print\_count():** '②카운트 상황'을 출력하는 함수로 볼 카운트, 스트라이크 카운트, 아웃 카운트를 매개변수로 전달 받아, 화면에 타자의 카운트와 해당 이닝의 아웃 카운트 상황을 출력합니다. 각각의 카운트를 나타내기 위해 ○(빈 카운트 의미), ●(카운트가 하나 있음을 의미) 문자('ㅅ' 키 입력 후 한자키를 눌러서 해당 문자 선택)를 사용하여 적절히 출력합니다.
- **void print\_base():** '③주자 상황'을 출력하는 함수로 1루, 2루, 3루의 상태를 매개변수로 전달 받아 현재 주자들이 몇 루에 진출해 있는지 출력합니다. 베이스 상태를 표현하기 위해 ◇(베이스에 주자가 없는 경우), ◆(베이스에 주자가 있는 경우) 문자와 ↗↘↙↖ 문자('ㅁ' 키 입력 후 한자키를 눌러서 해당 문자 선택)를 사용하여 적절히 출력합니다.

## 2. 플레이어의 선택

플레이어는 스윙 혹은 거르기를 입력을 통해 선택할 수 있습니다. 각각을 선택했을 때 발생하는 상황은 다음과 같습니다.

### (1) 스윙 선택

사용자가 스윙을 선택하고 나면 화면을 지운 후, 안타(1루타), 2루타, 3루타, 홈런, 플라이 아웃, 스트라이크 6가지의 경우 중 하나가 랜덤 하게 선택됩니다. 그리고 선택된 값에 따라 게임이 진행됩니다.

#### ➤ 사용자 정의 함수

- **get\_swing\_result():** 안타(1루타), 2루타, 3루타, 홈런, 플라이 아웃, 스트라이크 중에 하나가 랜덤 하게 선택 된 후, 그 결과를 반환합니다. 단, 안타가 나올 확률 14%, 2루타가 나올 확률 7%, 3루타가 나올 확률 5%, 홈런이 나올 확률 3%, 플라이 아웃이 나올 확률 33%, 스트라이크일 확률을 38%로 설정합니다. 확률 설정을 위해 rand() 함수를 활용합니다. 또한, 스윙의 결과가 프로그램이 실행될 때마다 다른 값을 반환하도록 srand() 함수를 활용합니다.

힌트)

- A. 확률 설정은 Tips - '확률 설정'(마지막 쪽)을 참고하여 작성합니다.
- B. 반환 값의 경우, 1은 1루타, 2는 2루타, 3은 3루타, ... 정도로 값을 매칭하여 반환할 수 있음

#### ➤ 안타(1루타)

안타를 치게 되면 그림 4와 같이 안타라는 안내메시지가 출력되고, 타자와 주자가 진루하게 되어 해당 베이스의 상태가 변경됩니다. 또한, 다음 타자를 위해 타자의 **볼 카운트와 스트라이크 카운트는 초기화** 됩니다.

그림4는 주자가 1루에 있는 상황에서 타자가 안타를 쳐서 타자는 1루로, 1루 주자는 2루로 이동한 상황입니다.

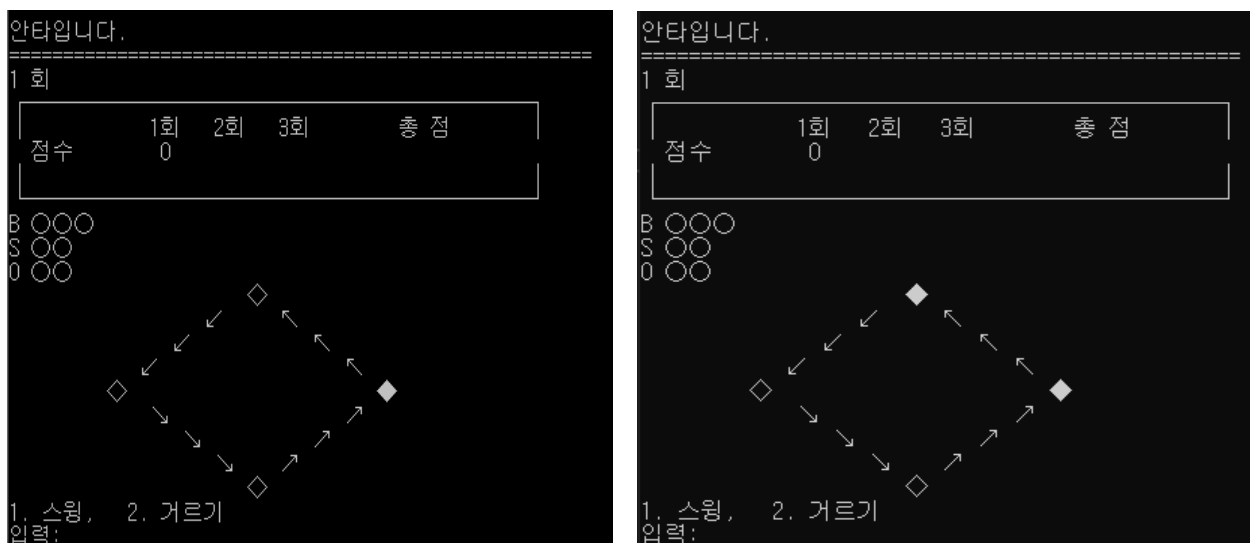


그림 4. 안타 화면 예시

그림5와 같이 3루까지 주자가 진출해 있는 경우에 안타를 치면 홈인 했다는 메시지와 함께 점수가 1점 추가되며 점수판에 기록이 됩니다.



그림 5. 안타를 통한 홈인 화면

## ➤ 2루타

2루타를 치게 되면 그림 6과 같이 2루타라는 안내메시지가 출력되며, 타자와 주자가 진루하게 되어 해당 베이스의 상태가 변경됩니다. 또한, 다음 타자를 위해 타자의 볼 카운트와 스트라이크 카운트는 초기화 됩니다.

그림 6의 경우는 주자가 없는 상황에서 타자가 2루타를 친 경우로 타자가 2루로 이동한 상황입니다.

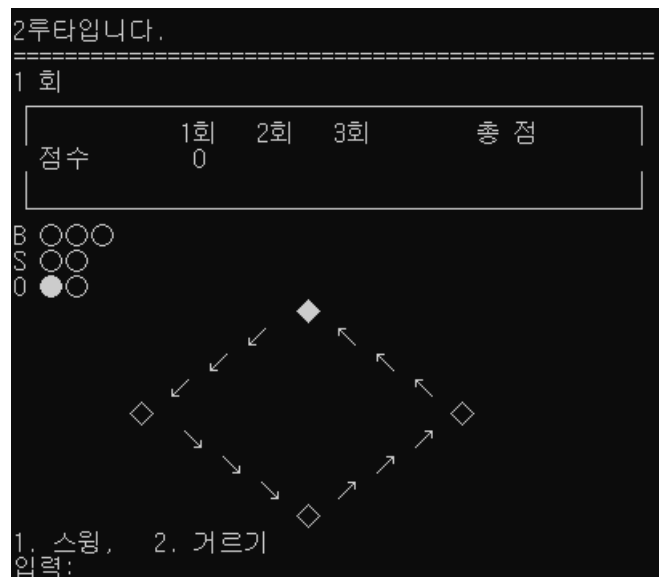


그림 6. 2루타 화면

그림7과 같이 만루의 상황에서 2루타를 치면 메시지와 함께 점수가 2점 추가되며 점수판에 기록이 됩니다.

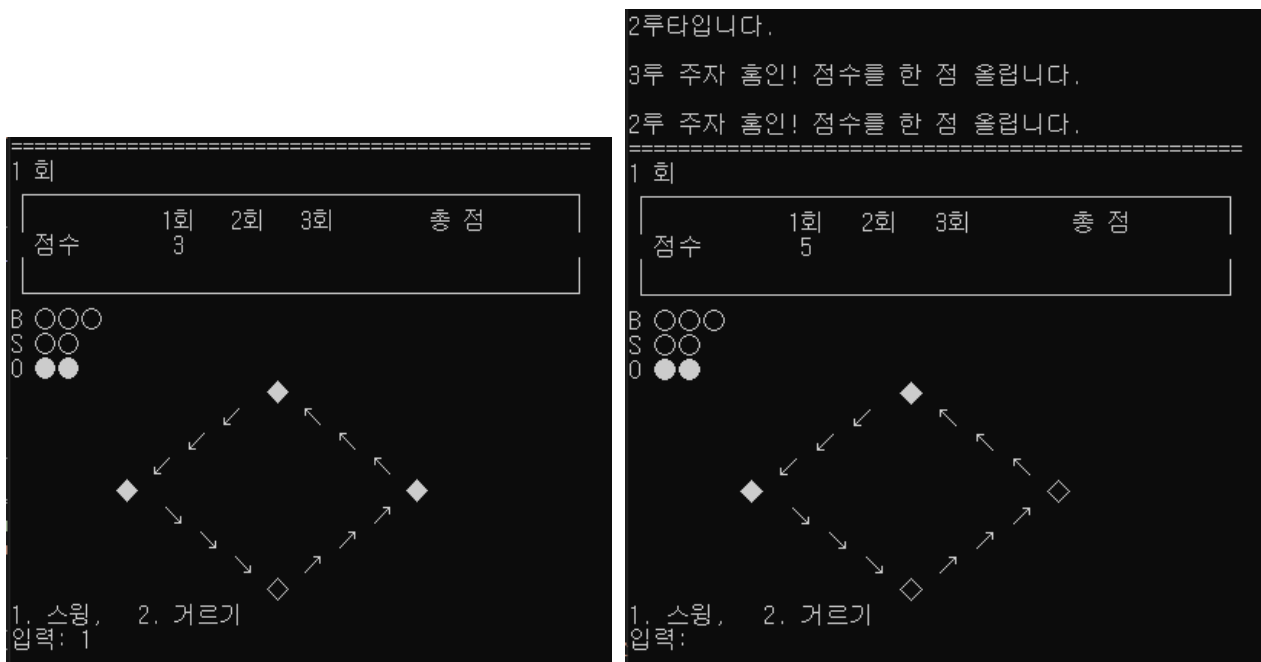


그림 7. 2 루타를 통한 홈인 화면

### ➤ 3루타

3루타를 치게 되면 그림 8과 같이 3루타라는 안내메시지가 출력되고, 타자와 주자가 진루하게 되어 해당 베이스의 상태가 변경됩니다. 또한, 다음 타자를 위해 타자의 볼 카운트와 스트라이크 카운트는 초기화 됩니다.

그림 8의 경우는 주자가 없는 상황에서 타자가 3루타를 친 경우로 타자가 3루로 이동한 상황입니다.

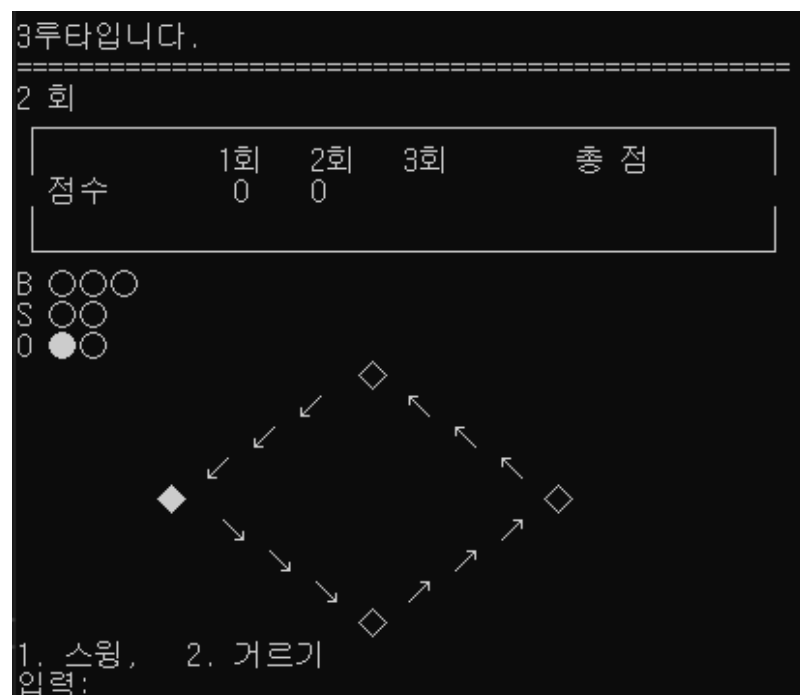


그림 8. 3 루타 화면



그림9와 같이 2루 주자만 있는 상황에서 3루타를 치면 메시지와 함께 점수가 추가되며, 타자는 3루까지 진루합니다.

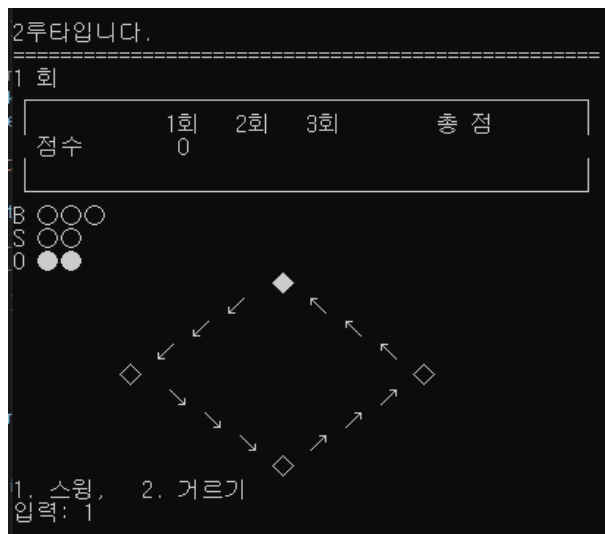


그림 9. 3 루타를 통한 홈인 화면

#### ➤ 홈런

홈런을 치게 되면 그림 10과 같이 홈런이라는 안내메시지, 출루해 있던 주자가 홈인했다는 안내메시지를 출력하며(만일 2, 3루에도 주자가 있었다면 안내메시지를 모두 출력), 타자와 주자가 모두 홈 베이스로 들어옵니다. 또한, 다음 타자를 위해 볼 카운트와 스트라이크 카운트는 초기화 됩니다.

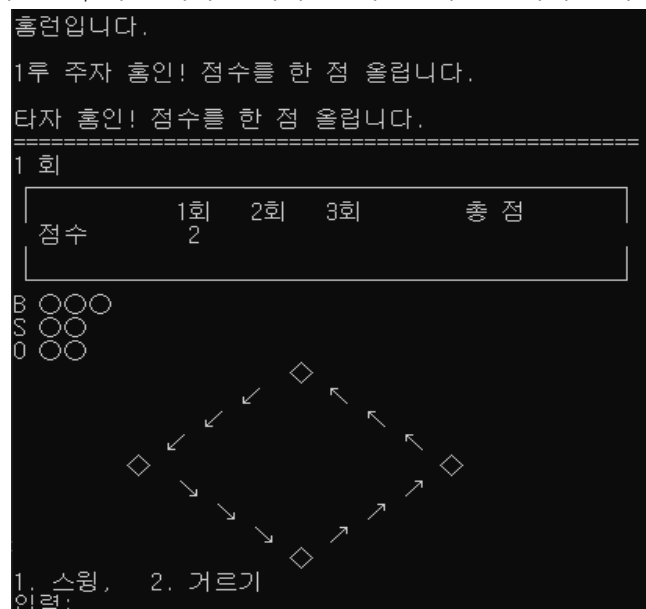


그림 10. 홈런을 통한 홈인 화면

### ➤ 플라이 아웃

플라이 아웃이란, 타자가 친 공을 9명의 수비수 중 한 명이 땅 또는 펜스에 닿지 않은 상태에서 잡을 때 타자에게 선언하는 아웃입니다.

타자가 스윙을 선택하고 플라이 아웃인 경우, 플라이 아웃이라는 메시지가 출력되며 아웃카운트가 하나 추가됩니다. (그림 11)

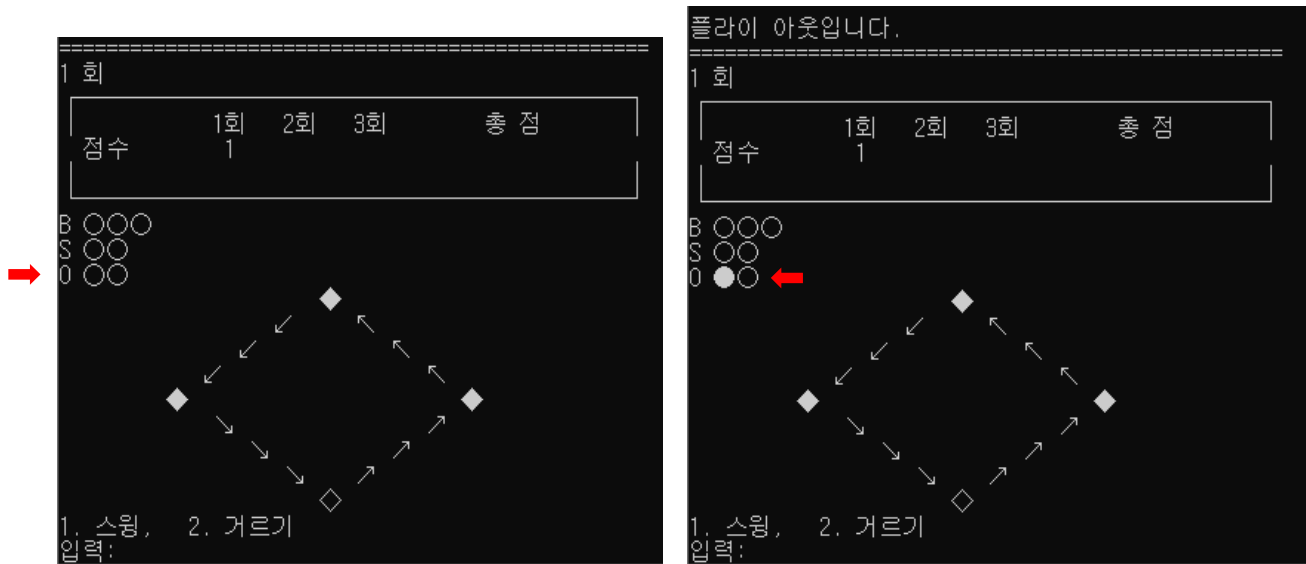


그림 11. 플라이 아웃 화면

만일, 아웃 카운트가 3이되면 이닝이 종료되었다는 메시지와 다음 이닝으로 넘어가며(그림 12는 1회 종료 후 2회 시작) 아웃카운트 또한 0으로 초기화 됩니다. 만일, 3회에서 3아웃이 되면 경기가 종료되었다는 메시지와 함께 최종적으로 얻은 점수를 출력합니다.



그림 12. 플라이 아웃으로 인한 이닝 전환

## ➤ 스트라이크

- 스윙을 선택하고 스트라이크인 경우, 스트라이크라는 메시지가 출력되며 스트라이크 카운트가 하나 추가된다. (그림 13은 1 스트라이크에서 2 스트라이크로 증가)

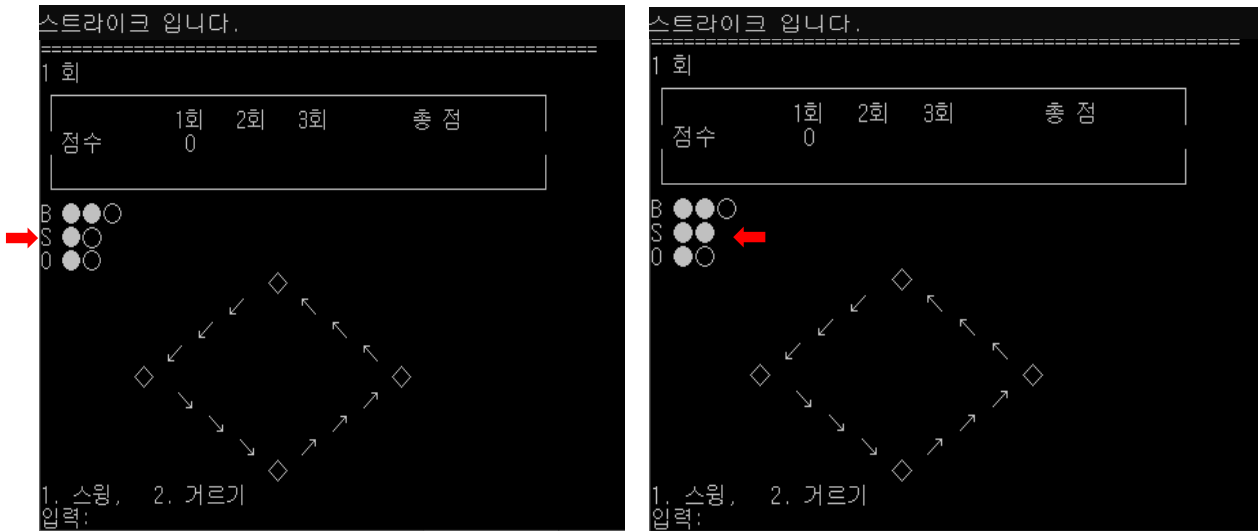


그림 13. 스트라이크 화면

- 스트라이크 카운트가 3이되면 그림 14의 오른쪽 예시와 같이 "스트라이크 아웃!!"이라는 메시지가 출력되고, 다음 타자로 넘어가며 아웃 카운트가 하나 추가된다. (그림 14는 1 아웃에서 2 아웃으로 증가) 또한, 볼 카운트와 스트라이크 카운트는 0으로 초기화 된다.

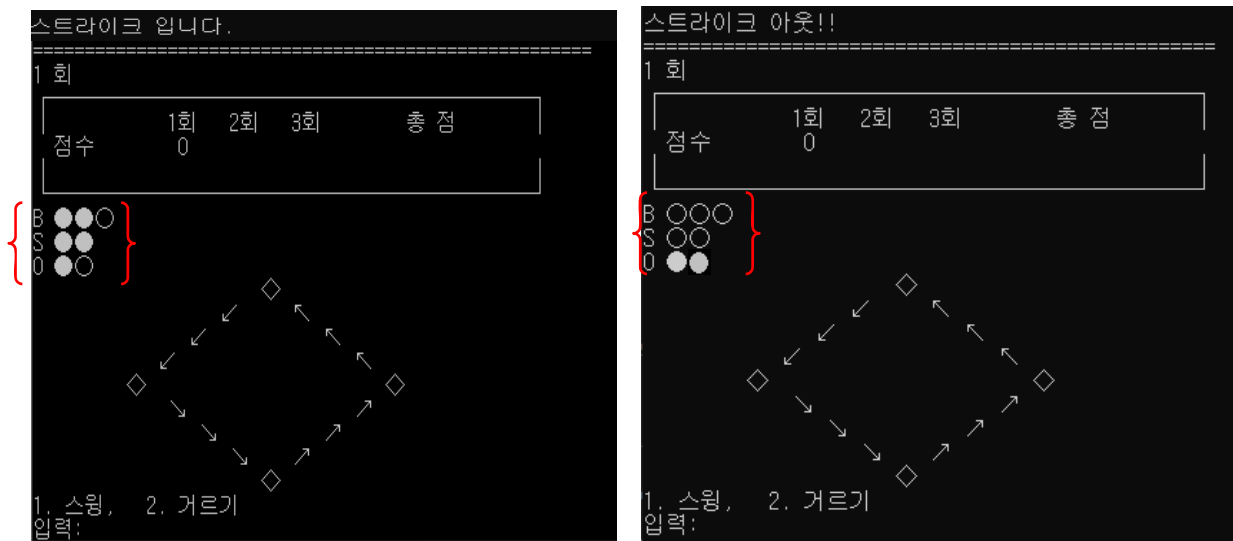


그림 14. 스트라이크 아웃 화면

- 2 아웃, 2 스트라이크 상황에서 스트라이크이면, 이닝이 종료되어 다음 이닝으로 넘어가며, 3회에서 이와 같은 상황이면 경기가 종료되었다는 메시지와 함께 최종 점수가 출력됩니다.

## (2) 거르기 선택

사용자가 거르기를 선택하고 나면 먼저 화면을 지우고, 볼과 스트라이크 2가지의 경우 중 하나가 랜덤하게 선택됩니다. 그리고 선택된 값에 따라 게임이 진행되어 그 결과가 출력됩니다.

## ➤ 사용자 정의 함수

- **get\_pass\_result()**: 볼 또는 스트라이크가 랜덤 하게 선택된 후, 그 결과를 반환합니다. 단, 볼이 나올 확률을 65%, 스트라이크가 나올 확률을 35%로 설정한다. 확률 설정을 위해 rand() 함수를 활용합니다.

확률 설정은 Tips - '확률 설정'(마지막 쪽)을 참고하여 작성합니다.

## ➤ 스트라이크

- 공을 타격을 안 하고 걸렸을 때, 스트라이크인 경우 스트라이크 카운트가 하나 추가됩니다. (그림 15는 0 스트라이크에서 1 스트라이크로 증가)

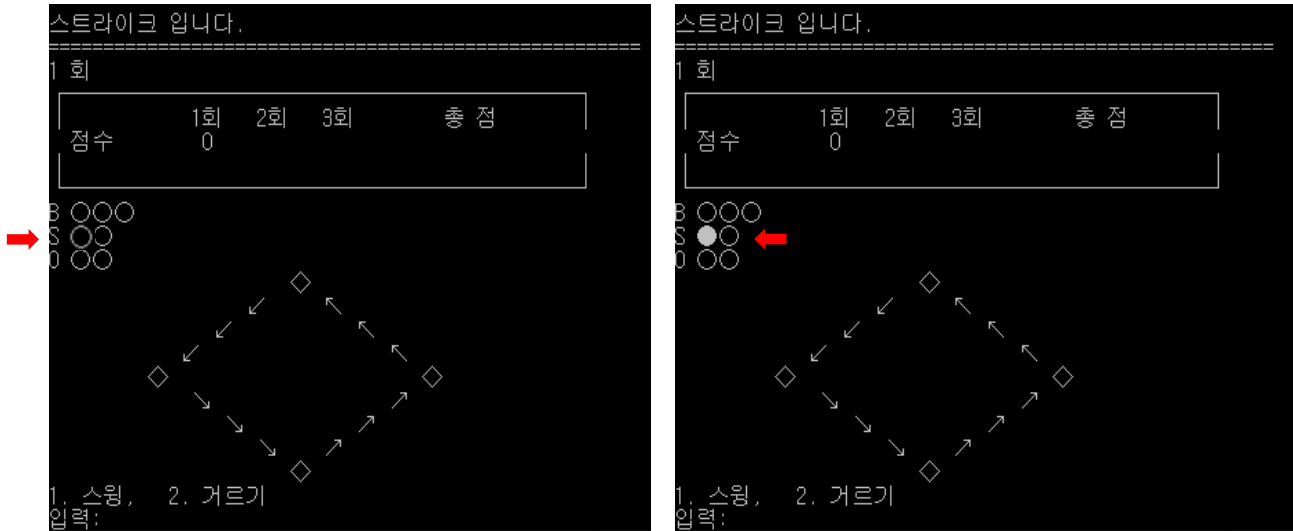


그림 15. 스트라이크 화면

- 공을 거르고 3 스트라이크가 되면, 그림 16의 오른쪽과 같이 "스트라이크 아웃!!"이라는 메시지가 출력되고 아웃카운트가 추가됩니다. (그림 16은 1 아웃에서 2 아웃으로 증가) 또한, 볼 카운트와 스트라이크 카운트는 0으로 초기화 됩니다.

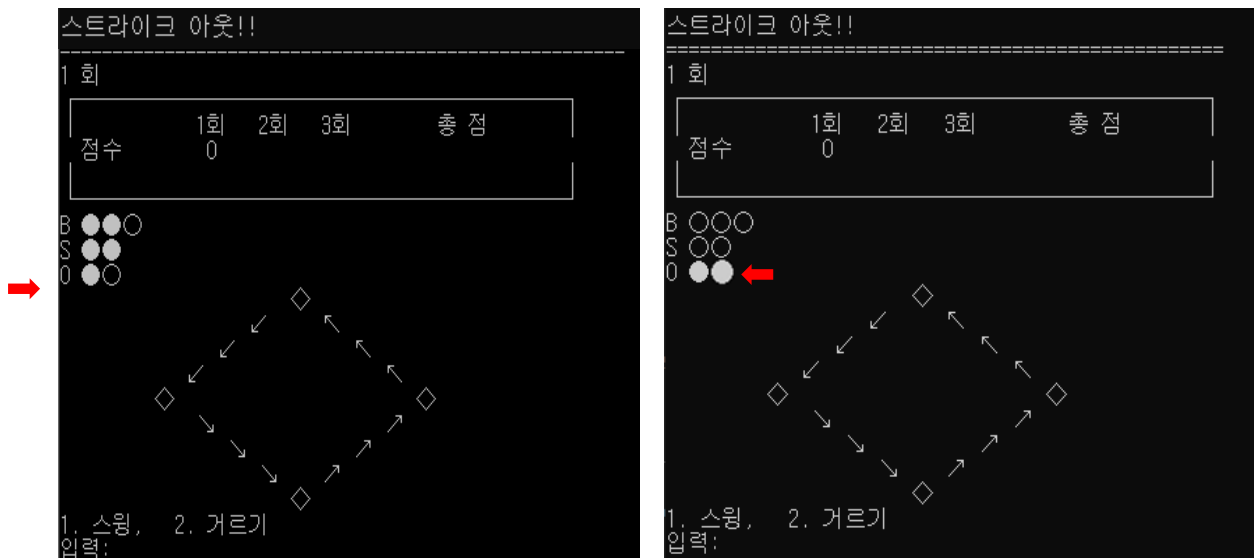


그림 16. 스트라이크 아웃 화면

- 2 아웃, 2 스트라이크 상황에서 다시 한 번 스트라이크이면 이닝이 종료됩니다. 3회에서 이와 같은 상황이면 경기가 종료되었다는 메시지와 함께 최종 점수가 출력됩니다.

➤ 볼

- 공을 타격 안 하고 걸렸을 때, 볼인 경우 볼 카운트가 하나 증가합니다. (그림 17은 0 볼에서 1 볼로 증가)

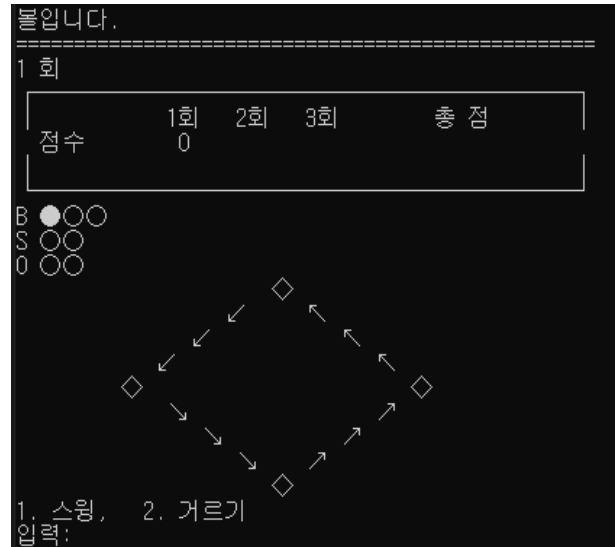


그림 17. 볼 화면

- 공을 거르고 볼넷이 되면 타자가 1루로 진루합니다. 이미 진출한 주자가 있다면, 주자들 또한 한 베이스씩 진루하게 됩니다. 단, 볼넷의 경우에는 진루하고자 하는 베이스에 주자가 없는 경우라면, 그 베이스의 다음 주자들은 이동하지 않습니다. (ex. 주자가 2, 3루에 있는 경우, 현재 타자가 볼넷으로 출루하게 되면 2, 3루의 주자는 다음 베이스로 이동하지 않습니다.)
- 볼넷으로 3루 주자가 홈베이스로 들어오는 경우, 홈인 했다는 추가 메시지와 함께 점수가 1점 추가됩니다. 그림 18은 볼넷으로 3루의 주자가 홈인하여 점수를 획득하는 화면이며, 다음 타자를 위해 볼 카운트와 스트라이크 카운트가 초기화 된다.

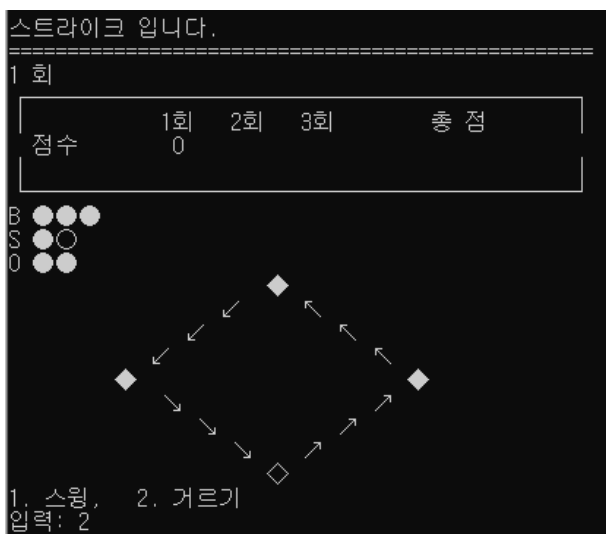


그림 18. 볼 넷 화면

### (3) 1~2외의 숫자 입력

만일, 사용자가 스윙(1), 거르기(2) 외의 숫자를 입력하면, 그림 19와 같이 에러 메시지를 출력하고, 이전 상황에 이어 게임을 진행하도록 합니다.



그림 19. 선택 오류 화면

### 3. 경기 종료 및 승패 판정

3회의 게임이 끝난 경우, "경기가 종료되었습니다."라는 메시지와 함께 점수판에 총점이 함께 출력됩니다. 이 때, 플레이어가 획득한 총점으로 승패를 결정하며, 플레이어가 10점 이상을 획득한 경우 플레이어의 승리가 됩니다. 아래의 예제를 참고하여 게임의 결과를 출력하도록 합니다. 또한, -1을 입력하면 화면 지우기 후 그림 1의 초기 선택 메뉴로 돌아가도록 합니다. (단, 메인 메뉴로 돌아가기 위한 입력으로 -1 외에는 고려할 필요가 없습니다.)

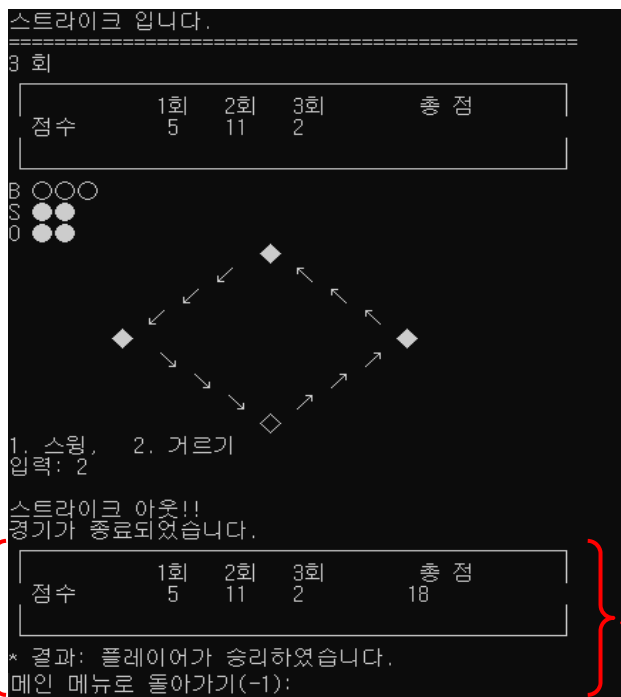


그림 20. 게임 결과 화면(승리)

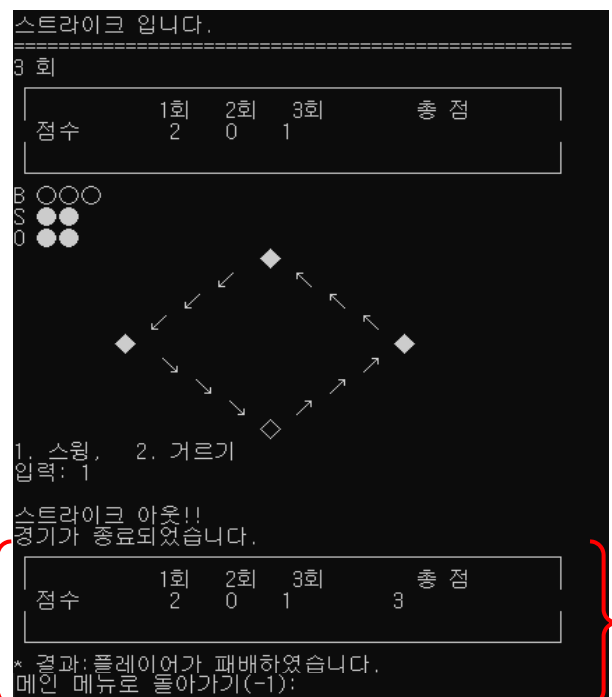


그림 21. 게임 결과 화면(패배)

➤ 3회 종료 이전에 10점 획득 시

- 3회 이전에 10점을 획득 하면 더 이상 게임을 진행하지 않아도 승리가 결정되므로, 게임 도중 10점을 획득하면 해당 이닝 까지만 경기를 진행하도록 합니다. 해당 이닝이 3아웃으로 이닝이 종료 가 되면 경기가 종료되었다는 메시지, 총점 그리고 플레이어가 승리하였다는 결과를 출력합니다. (그림 22는 1회에서 총 13점을 획득하여 1회가 종료 된 후 경기 종료 화면 임)

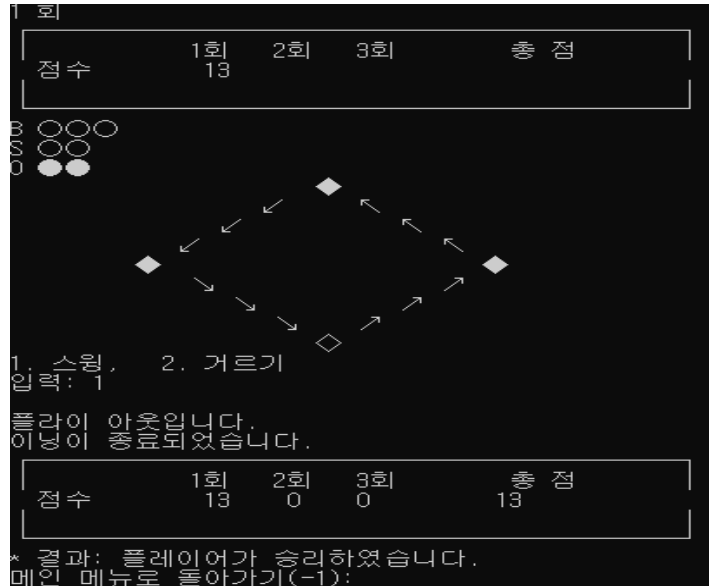


그림 22. 게임 결과 화면(승리)

### Ⅲ. 게임 설명

그림 1에서 “2. 게임 설명” 메뉴를 선택하면 화면이 지워진 후, 그림 23과 같이 게임 설명이 출력된다. 아래에 “메인 메뉴로 돌아가기(-1)”라는 메시지가 출력되며 사용자 입력을 기다린다. 사용자가 -1을 입력하면 화면 지우기 후 그림 1의 메뉴가 출력된다.

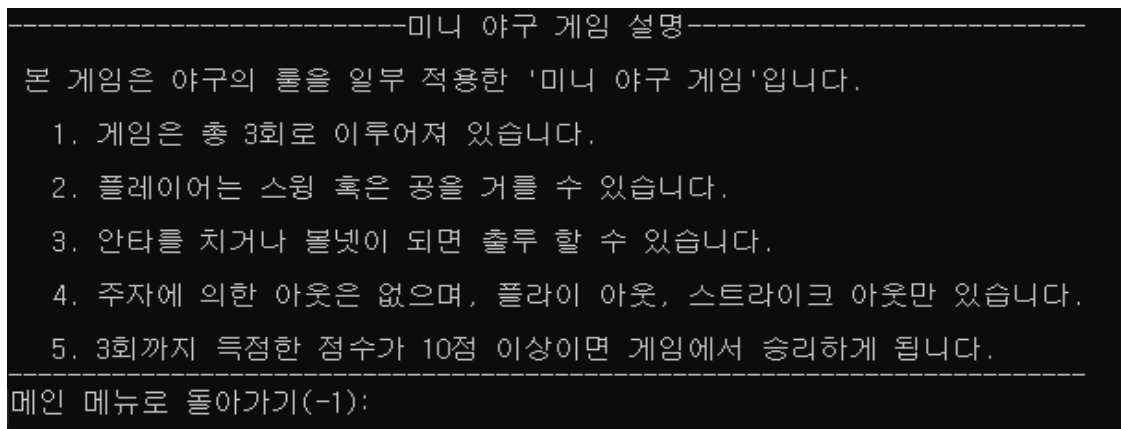


그림 23. 게임 설명 화면

- 사용자 정의 함수

- `print_info()`: 게임에 대한 설명을 출력하는 함수

#### IV. 게임 종료

그림 1에서 “3. 게임 종료” 메뉴를 선택하면, 그림 24와 같이 프로그램 종료메시지를 출력 후, 프로그램을 종료한다.

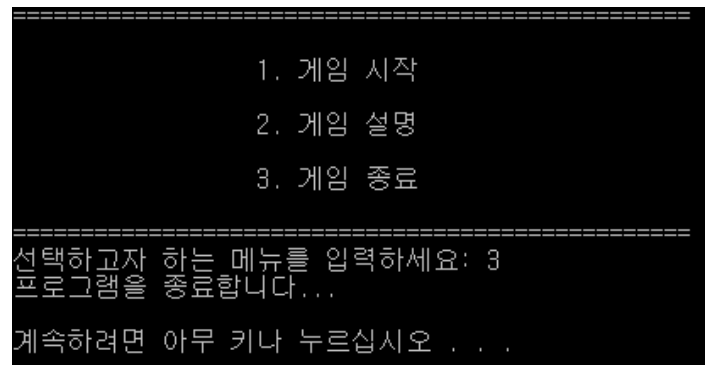


그림 24. 게임 종료 화면 예시



## Tips.

### ● 화면 지우기

화면을 지우고 싶을 경우, `stdlib` 헤더 파일을 포함 시킨 후 `system()` 함수를 사용하도록 한다. `system()` 함수의 매개변수로 "cls"를 넘겨주면 된다. 아래의 소스 코드를 컴파일하여 실행하면 콘솔 화면에 "Erase me!" 문자열이 출력된다. Enter키를 입력하면 화면이 지워진 후에 "Erased" 문자열이 출력되는 것을 볼 수 있다.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char c;

    printf("Erase me!\n");
    scanf("%c", &c);
    system("cls");
    printf("Erased\n");

    return 0;
}
```

※ 리눅스에서 테스트 해보길 원하는 경우, `system("cls");` 대신에 `system("clear");`로 변경하면 된다.

### ● 확률 설정

특정한 확률로 발생하는 사건을 나타내기 위해서 난수를 통해 확률을 표현할 수 있다. 만일 볼과 스트라이크가 각각 60%, 40% 확률로 발생한다면, 아래의 예제와 같이 발생하는 난수의 범위를 0~99(100개의 정수)로 지정하여 발생하는 난수를 통해 각각의 사건이 발생했음을 표현할 수 있다.

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

int main()
{
    int ball_random = rand() % 100;

    if (0 <= ball_random && ball_random < 60) {
        printf("볼입니다.\n");
        printf("생성된 난수는: %d\n", ball_random);
    }
    else {
        printf("스트라이크입니다.\n");
        printf("생성된 난수는: %d\n", ball_random);
    }

    return 0;
}
```