**2020년 2학기**

**프로그래밍과 문제해결**

**Assignment #1**

**담당교수: 윤은영**

**학번: 20200854**

**학과: 무은재학부**

**이름: 황찬기**

**POVIS ID: hcg0827**

**명예서약(Honor code)**

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

**Problem : 숫자 야구 게임**

1. 문제 개요

이 프로그램은 숫자 야구 게임을 수행하는 프로그램으로, 자세한 규칙은 아래와 같다.

* 기본적으로 컴퓨터가 생성한 세 자리 수를 맞추는 게임이다.
* 생성된 세 자리 수와 입력한 세 자리 수를 비교하여, 위치는 다르지만 같은 숫자가 존재하면 볼, 위치와 숫자가 같으면 스트라이크라 하며 매 시도마다 스트라이크와 볼의 개수를 알려준다. 숫자와 그 위치가 모두 일치하지 않을 경우는 아웃이며, 3회 아웃될 경우 이닝이 종료된다. 또한 수 입력 시 -1을 입력하면 즉시 이닝이 종료된다.
* 게임은 여러 이닝으로 구성되며, 매 이닝이 끝날 때마다 다음 이닝을 진행할 지 여부를 결정할 수 있다. 한 이닝은 10번의 시도 횟수로 구성되어 있다.
* 매 이닝의 기본 점수는 100점이며, 시도 횟수가 1회 늘어나면 점수는 10점씩 감소한다. 3회 아웃되거나 10회 시도에도 맞히지 못하는 경우, -1을 입력하여 이닝을 종료한 경우 점수는 0점이 된다. 매 이닝이 종료될 때마다 해당 이닝의 점수를 출력하며, 게임을 종료하는 경우 전체 게임에서의 평균 점수를 출력한다.
* **절반 이상의 기회를 소모한 경우, 각 자리 숫자가 같은(111, 222 등) 수 중 랜덤으로 정해진 숨겨진 수를 입력하여 가장 앞의 숫자를 알 수 있다. (힌트 제공/추가 기능)**

이 프로그램의 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.

* 입력부 : 답, 숨겨진 숫자를 생성하고 사용자가 입력한 숫자를 저장한다.
* 처리부 : 답, 숨겨진 숫자와 사용자가 입력한 숫자를 비교하여 스트라이크, 볼 아웃의

횟수를 계산한다.

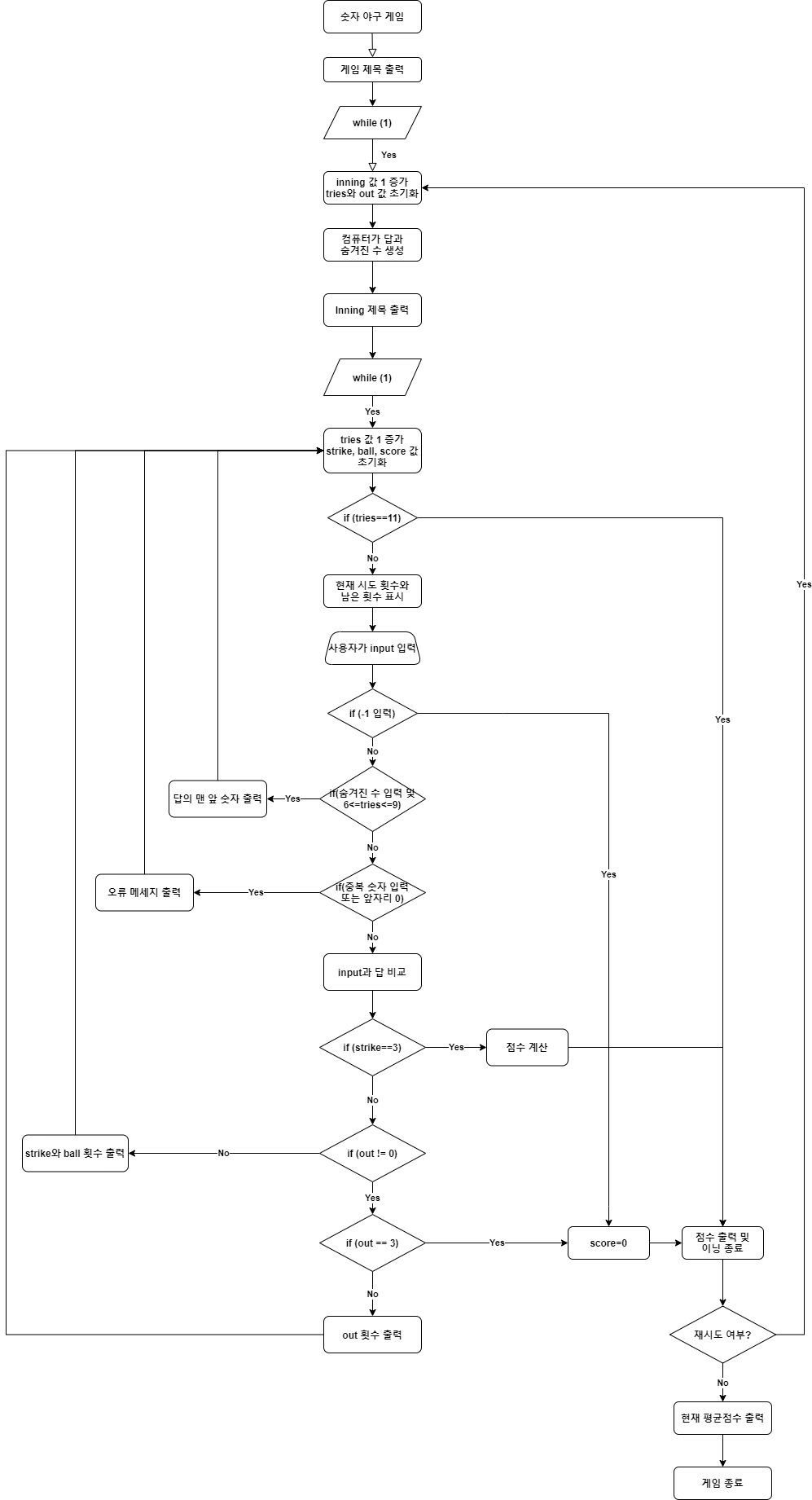
* 출력부 : 스트라이크, 볼, 아웃의 횟수를 출력하고, 게임이 끝나면 점수를 출력한다.

2. 알고리즘

이 프로그램에 대한 알고리즘을 Pseudo Code로 나타내면 아래와 같다.

|  |  |
| --- | --- |
| Pseudo-algorithm for Numerical Baseball | |
|  | *// 프로그램에서 필요한 변수는 미리 선언한 것으로 가정* |
| 1 | Include essential headers <stdio.h>, <stdlib.h>, <time.h> |
| 2 | Start repetitive 1(while) which refers to innings |
| 3 | Generate random number ‘answer’ and hidden number ‘hid\_num’ |
| 4 | Start repetitive 2(while) which refers to tries |
| 5 | Quit the inning if tries are over 10. |
| 6 | Prompt user and read number ‘input’ |
| 7 | Check that the number is valid |
| 8 | If input=-1, exit the inning |
| 9 | If input=hid\_num while tries are over 5, give the number of first position. |
| 10 | If there are same numbers in input, start next try |
| 11 | In else situation, compare two numbers |
| 12 | Same Number different position : +ball |
|  | Same Number same position : +strike |
|  | No match : +out |
| 13 | If strike is 3, exit the inning and print score |
| 14 | If out is not 0, print the count of out. |
| 15 | If out is 3, exit the inning and print score |
| 16 | In else situation, print result of that try and start next try |
|  | (End of the repetitive 2) |
| 17 | Sum the score of inning at ‘sum’ |
| 18 | Read character ‘rnd\_continue’ |
| 19 | If ‘rnd\_continue’ is ‘Y’ or ‘y’, start next inning |
|  | In else siuation, quit the game |
|  | (End of the repetitive 1) |
| 20 | Print average score of innings |

이 Pseudo Code를 Flowchart로 표현하면 다음과 같다.



3. 프로그램 구조 및 설명

a) 게임 및 이닝의 제목 출력

- 프로그램을 실행하면 가장 먼저 #으로 이루어진 구분선과 제목이 출력된다. 구분선은 이닝 제목을 출력할 때도 함께 출력된다.

- 함수(draw\_line)를 정의하여 #을 31개 출력한다.

b) 새 이닝 시작 및 컴퓨터가 답과 숨겨진 수를 생성

- while문을 이용하여 이닝 반복을 위한 구조를 만들고, rand 함수를 이용하여 매 이닝마다 숫자 야구 게임의 정답과 힌트를 제공하는 숨겨진 수(모두 같은 숫자로 이루어진 세 자리 수)를 생성한다.

- 매 이닝을 시작할 때마다 inning 값을 1씩 증가시키며, 시도 횟수(tries)와 아웃(out), 점수(score) 값을 0으로 초기화하고, 몇 이닝인지를 알려주는 제목을 출력한다.

c) 시도 반복을 위한 구조 생성 및 입력 값 받기

- while문을 이용하여 이닝 내에서 시도를 반복하기 위한 구조를 생성한다. 반복할 때마다 tries 값을 1씩 증가시키고, strike, ball 값을 0으로 초기화한다.

- 시도 횟수가 증가하다 11이 될 경우 score 값을 0으로 한 후 답과 점수를 출력하고 이닝을 종료한다.

- 이 후 현재 시도 횟수와 남은 횟수를 □와 ■를 이용하여 출력하고, scanf를 이용하여 사용자의 입력 값을 받는다. 이때 -1을 입력하면 0점 처리 후 이닝이 즉시 종료되고, 중복되는 숫자를 입력하거나 맨 앞자리에 0이 오는 경우 오류 메시지를 출력하고 다음 시도로 넘긴다. 단, 절반 이상의 시도를 소모한 후 숨겨진 숫자를 입력하면 답의 가장 앞 숫자를 알려주고 다음 시도로 넘긴다. 마지막 시도에서는 의미가 없으므로 마지막 시도는 제외한다. (즉 6~9회째 시도 사이에서만 가능)

d) 답과 사용자의 입력 값을 비교 및 출력

- 위의 경우에 해당하지 않을 경우 답과 사용자가 입력한 값을 비교한다. 이때 위치와 숫자가 모두 같으면 같은 개수만큼 strike 값을 1씩 증가, 위치는 다르고 숫자만 같으면 같은 개수만큼 ball 값을 1씩 증가시킨다. 같은 숫자가 아예 없으면 out 값을 1 증가시킨다.

- strike 값이 3인 경우(답과 입력 값이 일치하는 경우) 현재까지의 점수를 아래 식을 통해 계산하여 출력한다. 이후 해당 이닝을 종료한다. ( tries-1인 이유는 값을 루프의 처음에 증가시키기 때문에 최종 값은 실제 시도 횟수보다 1이 더 크다.)

score = 100 - 10 \* (tries - 1)

- out 값이 0이 아닌 경우, 현재까지의 아웃 횟수를 출력한 후 다음 시도로 넘어간다. 단 out 값이 3인 경우에는 점수 값을 0으로 하고 답과 점수를 출력한 후, 해당 이닝을 종료한다.

- 이외의 경우에는 스트라이크와 볼 값이 얼마인지 출력한 후 다음 시도로 넘어간다.

e) 재시도 및 게임 종료 설정

- 매 이닝이 끝날 때마다 점수를 sum이라는 변수에 합산하고, 재시도를 하는 지의 여부를 사용자의 문자를 받아 결정한다.

- Y와 y가 입력되는 경우 재시도로 새로운 이닝을 시작하고, N과 n이 입력되는 경우 재시도를 하지 않는 것으로 판단하여 합산한 점수를 진행한 이닝 수로 나누어 평균 점수를 계산하고, 이를 %.1f를 이용하여 소수점 첫째자리까지 출력한다. 이 후 프로그램을 종료한다.

4. 프로그램 실행 예제

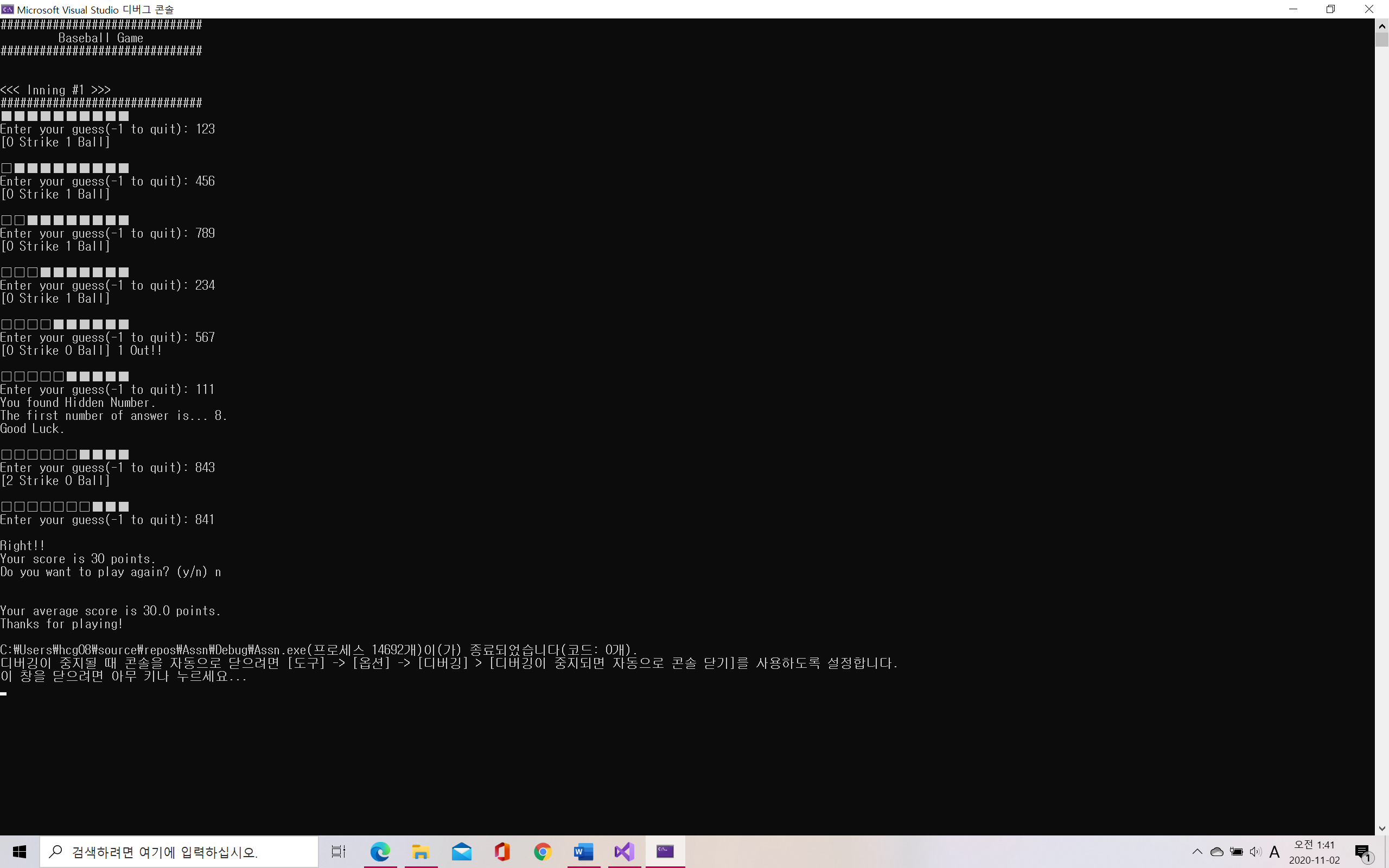
a) 답을 맞춘 경우

|  |  |
| --- | --- |
|  |  |
| 한 이닝만 진행하고 종료한 경우 | 여러 이닝 진행 후 종료한 경우 |

b) 점수가 0점인 경우

|  |  |
| --- | --- |
|  |  |
| -1을 입력하여 게임을 종료한 경우 | 3회 아웃된 경우 |
|  | |
| 시도 횟수(10회)를 모두 소모한 경우 | |

c) 숨겨진 수를 입력한 경우



5. 토론

- 재시도 여부를 scanf를 이용하여 문자를 받는 코드를 작성할 때, scanf가 무시되어 문자를 입력하지 못하는 문제가 발생했다. 그 원인은 첫번째 scanf에서 문자를 입력한 후 사용한 개행 문자 ‘엔터’가 남아있어 발생하는 문제이다.

- 이에 대한 해결 방법은 getchar();이나 fflush(stdin);을 추가하는 등 다양한 방법이 있는데, 이번 코드에서는 아래와 같이 %c 앞에 공백을 추가하여 이를 구분자로 인식하게 해 이러한 문제를 해결했다.

scanf(“ %c”, &rnd\_continue);

- 코드를 완성하고 직접 실행해보며 오류가 있는지 확인하던 중, strike의 값이 3임에도 게임이 끝나지 않고 [3 Strike 0 Ball]이라는 메시지만 출력되는 경우가 발생했다. 이 때문에 코드를 확인해 본 결과, strike의 값이 3인 경우에 수행되는 코드와 out의 값이 존재하는 경우 수행되는 코드의 우선순서가 바뀌어 strike의 값이 3임에도 이전에 out이 발생한 경우 넘어가지 않는 상황이였다.

6. 결론

- 이번 과제를 통해 while문과 for문 등 반복문을 적극 활용하여 이들의 다양한 활용 방법을 연습할 수 있었고, if문을 이용해 경우를 분류하며 프로그램의 수행 과정을 머릿속으로 구현하여 이를 작성하는 능력을 기를 수 있었다. 또한 프로그램 실행 과정에서 발생하는 여러가지 오류를 해결하며 어떤 상황에는 코드를 어떻게 수정해야 하는지 확인할 수 있었다.

7. 개선사항

- 이번 과제에서는 기존에 주어졌던 사항 이외에 ‘숨겨진 수’를 도입하여 답을 얻기 위한 힌트를 얻는 사항을 추가했다. 이외에도 한 두가지 사항을 더 추가하려 했으나 오류를 고치다 코드가 꼬여 추가하지 못했는데, 나중에 이를 추가하여 더 재미있게 만들어 보고 싶다.

- 컴퓨터가 생성한 답과 사용자가 입력한 수를 분석할 때, 이 프로그램에서는 각 자리의 숫자를 따로 생성하여 세 자리 수로 합친 후, 이를 비교할 때 다시 각 자리를 분리하여 사용했다. 다음에 배열에 대해 학습한다면, 각 자리의 숫자를 배열에 넣어 조금 더 간결하게 코드를 작성할 수 있을 것 같다.