

CSED101. Programming & Problem solving

Fall, 2020

Programming Assignment #1

정백진 (bjjung@postech.ac.kr)

■ 제출 마감일: 2020.11.02. 23:59

■ 개발 환경: Windows Visual Studio 2019

■ 제출물

- C 소스 코드 (assn1.c)
 - 프로그램의 소스 코드에 채점자의 이해를 돕기 위한 주석을 반드시 붙여주세요.
- 보고서 파일 (.docx, .hwp 또는 .pdf; assn1.docx, assn1.hwp 또는 assn1.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 LMS를 통해 제출해 주세요.

■ 주의 사항

- 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 (LMS의 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: 숫자 야구 게임

(문제)

컴퓨터가 임의로 생성한 3자리 숫자를 사용자가 맞히는 프로그램을 작성하세요.

(목적)

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

(주의사항)

1. 이번 과제는 함수를 정의하고 사용하는 방법을 익히는 문제이므로 main() 함수에 모든 기능을 구현한 경우 감점 처리 합니다.
2. 과제 보충 설명에서 반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인 후 구현하도록 합니다. 이 때, 설명에서 지정한 사용자 정의 함수의 매개변수의 개수와 자료형, 함수 이름, 반환 자료형 등은 자유롭게 변경이 가능합니다. 그러나 동일한 기능을 하는 함수는 반드시 있어야 하며, 변경 시 무엇을 어떻게 변경해서 구현했는지 보고서에 기록하도록 합니다. 이외에 필요한 함수는 정의해서 사용할 수 있습니다.
3. 프로그램 구현 시, main() 함수를 호출을 직접 하지 않습니다. 즉, 소스 코드 내에 main(); 이라고 호출하지 않습니다.
4. 전역 변수, 배열 및 goto 문은 사용할 수 없으며, 포인터의 경우 수업시간에 다룬 내용에 한해서 사용이 가능합니다.
5. 사용자 추측 입력은 숫자만 입력하는 것으로 가정합니다. 즉, 문자 등의 입력에 대해서는 고려할 필요가 없습니다.
6. 명시된 예러 처리 외에는 고려하지 않아도 됩니다.
7. 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.

(설명)

이 과제에서 작성할 ‘숫자 야구 게임’은 2인용 게임으로 컴퓨터가 생성한 숫자를 사용자가 추측하여 10번 시도 내에 맞히고, 그 과정과 결과를 출력하는 프로그램을 작성합니다.

1. 컴퓨터는 임의의 3자리 양수를 생성합니다. 단, 각 자리 숫자는 중복될 수 없습니다.
2. 사용자는 컴퓨터가 생성한 숫자를 맞히기 위하여 3자리 숫자를 추측하여 입력합니다. 중복된 숫자가 포함된 입력 또는 3자리 숫자가 아닌 입력을 하게 되면, ‘Wrong input!!’이라는 메시지가 출력되며 시도 횟수가 1 차감됩니다.
3. 프로그램은 사용자가 입력한 숫자와 컴퓨터의 숫자를 비교한 결과를 알려줍니다. 결과의 종류는 다음과 같습니다.
 - 1) **Strike**: 사용자가 추측한 3자리 수를 구성하는 숫자 3개 중에서 컴퓨터가 생성한 수에도 포함되어 있으며 그 자릿수(위치)까지 같은 것들의 개수
 - 2) **Ball**: 사용자가 추측한 3자리 수를 구성하는 숫자 3개 중에서 컴퓨터가 생성한 수에도 포함되어 있으나 자릿수(위치)가 틀린 것들의 개수
 - 3) **Out**: 추측한 숫자들 중 컴퓨터가 생성한 수에 들어있는 숫자가 하나도 없는 경우
4. 게임의 종료
 - 3 Strike 0 Ball 인 경우 (컴퓨터의 숫자를 맞힌 경우)

- 3 Out 인 경우 (완전히 틀린 추측을 3번 했을 경우)
- 최대 시도 횟수(10회)가 끝난 경우
- 사용자 추측 입력 시, -1을 입력한 경우

5. 점수 계산

- 기본 점수는 100점입니다.
- 사용자 입력 횟수가 증가함에 따라 10점씩 감점됩니다. 즉, 첫번째 시도에서 맞히면 100점, 두번째 시도에서 맞히게 되면 90점이 됩니다. 단, 최대 시도 횟수는 10회입니다.
- 3 Out 인 경우와 점수가 0점이되면 더 이상의 기회를 주지 않고 컴퓨터가 생성한 숫자를 출력하고 게임을 끝냅니다. 3 Out인 경우 획득 점수는 0점입니다.
- 사용자 추측 입력 시, -1을 입력한 경우 획득 점수는 0점입니다.

6. 한 게임이 끝나고 나면, 다시 게임을 할 것인지를 물어 보고, 만약 게임의 실행을 완전히 끝내고자 하는 경우에는 지금까지의 게임 점수의 평균을 출력하고 프로그램의 실행을 종료합니다.

- 게임 재 시작 여부에 대한 사용자 입력은 y, Y, n, N 의 4가지(대소문자)가 가능하며, 이외의 입력에 대해서는 고려하지 않습니다.
- 게임 점수의 평균은 소수점 이하 첫째자리까지 출력합니다.

(실행 예시 1)

컴퓨터가 첫번째로 생성한 숫자는 **873**, 두번째로 생성한 숫자는 **261**, 세번째로 생성한 숫자가 **943**에 해당하는 예시입니다. 예시의 **빨간색 밑줄**은 사용자의 입력입니다.

```
#####
Baseball Game
#####

<<< Inning #1 >>>
#####
■ ■ ■ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): 456
[0 Strike 0 Ball] 1 Out!!

□ ■ ■ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): 987
[0 Strike 2 Ball]

□ □ ■ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): 978
[1 Strike 1 Ball]

□ □ □ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): 879
[2 Strike 0 Ball]

□ □ □ □ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): 873
[3 Strike 0 Ball]
```

Right!!
Your score is 60 points.
Do you want to play again? (y/n) y

<<< Inning #2 >>>

#####

■ ■ ■ ■ ■ ■ ■ ■ ■ ■

Enter your guess(-1 to quit): 122
[0 Strike 0 Ball] Wrong input!!

□ ■ ■ ■ ■ ■ ■ ■ ■ ■

Enter your guess(-1 to quit): 378
[0 Strike 0 Ball] 1 Out!!

□ □ ■ ■ ■ ■ ■ ■ ■ ■

Enter your guess(-1 to quit): 453
[0 Strike 0 Ball] 2 Out!!

□ □ □ ■ ■ ■ ■ ■ ■ ■

Enter your guess(-1 to quit): 790
[0 Strike 0 Ball] 3 Out!!

The answer is 261.
Your score is 0 points.
Do you want to play again? (y/n) y

<<< Inning #3 >>>

#####

■ ■ ■ ■ ■ ■ ■ ■ ■ ■

Enter your guess(-1 to quit): 123
[1 Strike 0 Ball]

□ ■ ■ ■ ■ ■ ■ ■ ■ ■

Enter your guess(-1 to quit): 543
[2 Strike 0 Ball]

...

□ □ □ □ □ □ ■ ■ ■ ■

Enter your guess(-1 to quit): 943
[3 Strike 0 Ball]

Right!!
Your score is 30 points.
Do you want to play again? (y/n) n

Your average score is 30.0 points.
Thanks for playing!

그림 1. 실행 예시 1

(실행 예시 2)

컴퓨터가 첫번째로 생성한 숫자는 123, 두번째로 생성한 숫자는 460에 해당하는 예시

```
#####
      Baseball Game
#####

<<< Inning #1 >>>
#####
■ ■ ■ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): 123
[3 Strike 0 Ball]

Right!!
Your score is 100 points.
Do you want to play again? (y/n) Y

<<< Inning #2 >>>
#####
■ ■ ■ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): -1
[0 Strike 0 Ball]

The answer is 460.
Your score is 0 points.
Do you want to play again? (y/n) N

Your average score is 50.0 points.
Thanks for playing!
```

그림 2. 실행 예시 2

(보충 설명)

다음은 게임 진행에 대한 화면 구성 등에 대한 보충 설명 및 반드시 구현해야 하는 사용자 정의 함수를 설명합니다.

1. 프로그램을 실행하면, 컴퓨터는 임의의 3자리 양수를 생성한 후, 아래와 같이 사용자 입력을 기다립니다.

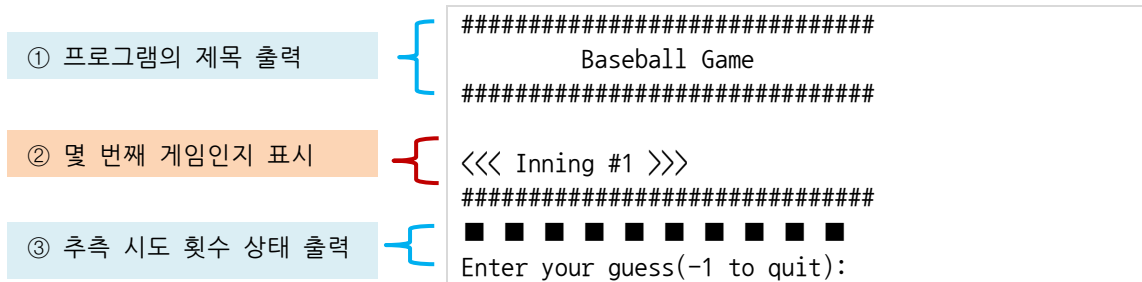


그림 3. 프로그램 초기 실행 화면 및 구성 설명

먼저, 게임(Inning)을 위해 컴퓨터는 임의의 3자리 양수를 생성합니다.

- 숫자는 프로그램을 실행할 때마다 다른 숫자가 생성되도록 작성을 합니다. 즉, rand(),

srand() 함수를 사용하여 임의의 수를 생성하면 됩니다.

- 각 자리 숫자는 중복될 수 없습니다.

- 사용자 정의 함수

- **get_random_num()**: 새로운 이닝 시작 시, 임의의 3자리 숫자를 생성해서 반환
- **is_valid_num(int num)**: 전달 받은 숫자(num)의 각 자리의 숫자를 비교하여 서로 중복되는 숫자가 없으면 1을 반환하고, 중복되는 숫자가 있으면 0을 반환하는 함수로, get_random_num()함수에서 이 함수를 호출하여 사용하도록 합니다. 사용자의 추측 입력 시에도 이 함수를 사용하도록 합니다.

게임 화면은 그림 3과 같은 구성을 가집니다.

① 프로그램의 제목 출력

프로그램 시작 시에 한 번 출력이 되는 부분으로, 제목 아래 위의 구분선(‘#’ 31개로 구성)을 출력하기 위해 반드시 아래 함수를 정의하고 사용합니다.

- **draw_line()**: 31개의 ‘#’을 출력하는 함수로, for문을 사용하여 출력합니다.

② 몇 번째 게임인지 표시

게임(Inning)은 한 회에 해당하는 게임을 말하는 것으로, 프로그램 처음 시작 시 “Inning #1”로 시작하여 게임 종료 후, 게임을 새로 시작할 때마다 1씩 증가합니다.

참고로, 프로그램 새로 실행 시 “Inning #1”로 시작됩니다.

③ 추측 시도 횟수 상태 출력

한 이닝에서 사용자의 현재 추측 시도 횟수 상태를 출력하는 부분으로, 아래의 함수를 반드시 정의하고 사용합니다.

- **void print_tries_status(int tries)**: 현재까지 시도한 추측 횟수를 전달 받아, 최대 추측 시도 횟수 10회 중 사용한 시도 횟수는 □, 남은 횟수는 ■을 이용하여 출력합니다. (그림 1, 그림 2의 실행 예시 참고)

2. 사용자는 컴퓨터가 생성한 숫자를 맞추기 위해 3자리 숫자를 추측해서 입력합니다.

```
#####
Baseball Game
#####

<<< Inning #1 >>>
#####
■ ■ ■ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): 456
[0 Strike 0 Ball] 1 Out!!

□ ■ ■ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): 137
[1 Strike 1 Ball]

□ □ ■ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit): 173
[2 Strike 0 Ball]

□ □ □ ■ ■ ■ ■ ■ ■ ■
Enter your guess(-1 to quit):
```

그림 4. 컴퓨터가 생성한 숫자(123)에 대한 사용자 입력 예시

그림 4는 컴퓨터가 생성한 숫자가 123인 경우로, 이 숫자와 사용자가 입력한 숫자를 비교하여 그 결과를 아래에 출력합니다.

이해를 돕기 위하여 각 사용자 입력에 대한 결과를 아래의 표로 나타냈으며, 주황색 셀은 스트라이크, 하늘색 셀은 볼, 회색 셀은 생성 숫자에 포함되지 않는 경우를 말합니다.

컴퓨터 생성 숫자	1	2	3	
첫번째 시도	4	5	6	Out
두번째 시도	1	3	7	1 Strike 1 Ball
세번째 시도	1	7	3	2 Strike 0 Ball

첫번째 시도는 입력한 각 자리 숫자가 컴퓨터가 생성한 숫자에 하나도 없는 경우로 아웃, 두번째 시도는 1이 있고 위치가 맞으므로 1스트라이크, 3은 있지만 위치는 다르므로 1볼, 세번째 시도는 1, 3이 위치까지 맞는 경우로 2스트라이크의 결과를 가집니다.

- 사용자 정의 함수

- **void compare(int comp, int user, int *strike, int *ball):** 컴퓨터가 생성한 숫자(comp)와 사용자가 입력한 숫자(user)를 비교하여 스트라이크와 볼의 수를 계산한 결과를 strike와 ball 변수를 통해 전달
- **print_comparison_results():** 스트라이크와 볼의 수 등을 전달받아 그 결과를 출력