

# CSED101. Programming & Problem solving

## Spring 2022

### Programming Assignment #2 (60 points)

이준영 (junyoungrhee@postech.ac.kr)

■ 제출 마감일: 2022.04.08 23:59

■ 개발 환경: Windows Visual Studio 2019

#### ■ 제출물

- C 소스 코드 (assn2.c)
  - 프로그램의 소스 코드에 채점자의 이해를 돕기 위한 주석을 반드시 붙여주세요.
- 보고서 파일 (.docx, .hwp 또는 .pdf; assn2.docx, assn2.hwp 또는 assn2.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 PLMS를 통해 제출해 주세요.

#### ■ 주의 사항

- 컴파일이나 실행이 되지 않는 과제는 0점으로 채점됩니다.
- 제출 기한보다 하루 늦게 제출된 과제는 최종 20%, 이를 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: 012구슬게임

### (문제)

컴퓨터와 대결하여 구슬을 빼앗아오는 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. 문서에 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. 프로그램에서 랜덤 시드는 프로그램 시작 시 main() 에서 srand(time(NULL)); 함수를 한번만 호출하도록 하여 한번만 초기화 한다.
6. 사용자 입력에서 숫자를 입력 받는 부분에는 숫자만 입력하는 것으로 가정합니다. 즉, 숫자 입력 받는 부분에는 문자 등의 입력에 대해서는 고려할 필요가 없습니다.
7. 명시된 예러 처리 외에는 고려하지 않아도 됩니다.
8. 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.

### (설명)

012 구슬게임은 구슬로 하는 일반 홀짝 게임에 변수를 추가하여 변형시킨 게임입니다. 모든 룰은 홀짝 게임과 비슷하지만 012 구슬게임에서는 숨겨진 구슬의 수를 홀 또는 짝이라고 추측하는 대신 구슬의 개수를 3으로 나누었을 때의 나머지를 추측해야 합니다. 본 과제에서는 유저가 컴퓨터를 상대로 게임을 진행하게 되며 먼저 구슬을 모두 잃은 플레이어가 패배하게 됩니다.

### I. ‘ 012 구슬게임 ’ 게임 규칙 상세설명

- 게임은 유저와 컴퓨터가 각자 20 개의 구슬을 갖고 시작하게 되며 게임은 한 플레이어가 구슬을 모두 잃으면 다른 플레이어가 승리하는 방식으로 진행됩니다.

- 유저와 컴퓨터는 번갈아 가며 각자의 턴을 진행합니다. 한 플레이어가 구슬을 숨기면 다른 플레이어는 숨겨진 구슬의 개수를 3으로 나누었을 때 나머지가 0 일지, 1 일지 아니면 2 일지를 맞춰야 합니다. 이때 구슬을 숨기는 플레이어는 자신이 현재 보유하고 있는 구슬의 개수 이상에 해당하는 구슬을 숨길 수 없으며 적어도 1 개 이상의 구슬을 숨겨야 합니다. 모든 게임은 컴퓨터가 먼저 구슬을 숨기면서 시작합니다.
- 만약 맞는 플레이어의 추측이 맞았다면 숨겨져 있던 구슬을 해당 플레이어가 모두 차지하게 됩니다. 반대로 추측이 틀렸다면 숨겨져 있던 구슬의 개수만큼 해당 플레이어가 상대방에게 구슬을 주어야 합니다.
- 앞서 설명했듯이 게임은 한 쪽 플레이어가 구슬을 모두 잃을 때까지 진행됩니다.

## II. ‘ 012 구슬게임 ’ 구현 요구사항

### 1. 프로그램 초기화면

- 프로그램을 시작했을 때 유저가 보는 초기 화면은 다음 요소들을 가지고 아래와 같이 출력합니다.
  - 게임의 이름: ‘ **Marble Game** ’
  - 환영의 메시지
- 프로그램 실행 화면 예시:

The screenshot shows the program's output with three distinct sections highlighted by red brackets and labels:

- 1. 프로그램 초기화면**: This section includes the title "[CSED101 Assignment 2]", a decorative border of 20 small squares, the game title "Marble Game", and a welcome message "Welcome to the Marble Game! Have fun~!".
- 2. 게임 시작화면**: This section shows the game start screen with "[Game #1]", a separator line, and the initial state: "USER: ○ (20)" and "COMP: ● (20)". Below this, there are two rows of 20 circles each, representing the initial 20 marbles for each player.
- 3. 컴퓨터가 구슬을 숨길 차례 (유저는 맞는 순서)**: This section shows the prompt "It's your turn to guess!" and the instruction "Guess 0, 1 or 2:", indicating it's the user's turn to guess the number of marbles hidden by the computer.

- 사용자 정의 함수
  - **void draw\_line():** 20 개의 ‘ ■ ’ 를 출력하는 함수로, for 문을 사용하여 출력합니다.  
프로그램 초기화면 출력 부분 등에서 적절히 호출하여 사용합니다.
  - ※ "■" 문자 입력은 ‘ □ ’ 키 입력 후, 한자키를 눌러 해당 문자 선택

## 2. 게임 시작화면

- 새로운 게임이 시작되었을 때 나오는 화면으로 다음 요소들이 반드시 포함되어 있어야 합니다:
  - 몇 번째 게임인지 (예. ‘ Game #1 ’ )
  - 유저와 컴퓨터의 구슬 보유 개수
- 게임은 게임 시작화면이 출력된 후 곧바로 시작됩니다 (아래 ‘ 3. 컴퓨터가 구슬을 숨길 차례 ’ 로 바로 넘어감).
- 게임 시작화면의 예시:

```
[Game #1]
-----
USER: ○ (20)
COMP: ● (20)
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
-----
```

- 사용자 정의 함수

- **void print\_marble\_status(int n1, int n2)**

- 유저와 컴퓨터가 각각 보유하고 있는 구슬의 개수를 매개변수로 전달받아 각 플레이어의 구슬 보유 개수와 상태를 출력해 주는 함수로, 구슬 보유 상태를 유저는 ○로, 컴퓨터는 ●을 이용하여 출력합니다 (기호 복사 붙여 넣기).
    - 구슬 출력 시 for 문을 이용하여 구현합니다.
    - 구슬은 먼저 유저가 보유하고 있는 수만큼 출력을 합니다. 유저 구슬은 첫 번째 줄의 왼쪽에서부터 오른쪽 방향으로 채워 나가며 첫 줄이 다 채워진 경우(20 개)에는 두 번째 줄의 왼쪽에서부터 오른쪽 방향으로 남은 구슬의 수만큼 채워서 출력합니다. 유저 구슬 출력이 끝나면 나머지를 컴퓨터의 구슬(●)로 출력합니다.
    - 아래는 유저의 구슬이 37 개, 컴퓨터의 구슬이 3 개인 예시입니다.

```
-----
USER: ○ (37)
COMP: ● ( 3)
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○○●●●
-----
```

## 3. 컴퓨터가 구슬을 숨길 차례

- 컴퓨터의 구슬 숨기기

모든 홀수 턴에는 컴퓨터가 구슬을 숨기게 됩니다 (첫 턴 포함). 이때 컴퓨터는 숨길 구슬의 개수를 자신의 보유하고 있는 범위 내에서 랜덤하게 정합니다.

➤ **int get\_rand\_num(int n):**

컴퓨터가 현재 보유하고 있는 구슬의 개수를 전달받아 1 과 해당 입력 숫자 사이의 임의의 정수 1 개를 반환

• 유저의 추측 값 입력

유저는 컴퓨터가 숨긴 구슬의 숫자를 3 으로 나누었을 때 나머지를 맞추기 위하여 추측하여 숫자를 입력합니다.

➤ **int get\_user\_guess(void):**

유저로부터 0, 1, 또는 2 를 입력 받는 함수입니다 (**scanf()** 사용). 우선 유저가 추측을 할 차례라고 설명을 한 후 추측을 입력하라고 지시합니다. 유저가 해당 범위 내의 숫자를 입력하게 되면, 해당 입력 값을 출력 후 반환합니다.

**예외처리)** 유저가 0, 1, 또는 2 이외의 다른 숫자를 입력하면 아래의 예시처럼 에러 메시지 출력 후, 올바른 범위 내의 입력을 받을 때까지 반복하여 새로운 입력을 받습니다.

➤ **int is\_valid\_num(int a, int b, int num):**

전달받은 숫자(num)가 a 이상 b 이하이면 사이의 범위( $a \leq num \leq b$ )에 해당하면 1 을, 아니면 0 을 반환하는 함수로, **get\_user\_guess()**에서 이 함수를 호출하여 사용하도록 합니다.

• 결과 확인

유저의 추측을 받은 이후에는 해당 추측이 맞았는지를 확인하고 다음과 같이 유저에게 결과를 알려줍니다:

■ 유저가 맞은 경우: 유저가 맞했다는 메시지와 함께 컴퓨터가 숨겼던 구슬의 개수를 공개한 후 그 개수만큼 유저의 구슬을 증가하는 동시에 컴퓨터의 구슬을 차감합니다.

■ 유저가 틀린 경우: 유저가 틀렸다는 메시지와 함께 컴퓨터가 숨겼던 구슬의 개수 및 정답을 공개한 후 그 개수만큼 유저의 구슬을 차감하는 동시에 컴퓨터의 구슬을 증가합니다.

※ 보유 구슬의 수가 상대방에게 줘야하는 구슬의 수보다 적은 경우에는 보유 구슬의 수만큼만 줄 수 있습니다. 예를 들면, 보유 구슬의 수는 5 개인데 줘야하는 구슬의 수가 10 개라면 보유한 5 개의 수만큼 줄 수 있습니다.

➤ **void change\_marble\_status(int \*winner, int \*loser, int num) :**

승자의 구슬 보유수와 패자의 구슬 보유수를 저장하고 있는 변수의 주소를 각각 전달받아, 승자는 구슬의 수(num)만큼 증가시키고 패자인 경우에는 구슬의 수만큼 감소시키는 함수

- 구슬 보유 상태 출력

턴이 끝나면 `print_marble_status()`를 사용해 현재 각 플레이어가 보유하고 있는 구슬의 개수를 보여줍니다.

- 구슬의 개수가 0 개 이하인 플레이어가 있으면 게임을 끝냅니다.
- 컴퓨터가 구슬을 숨길 차례일 때 게임화면 예시:

예시 1) 유저가 맞은 경우 (예시의 **빨간색 밑줄**은 사용자 입력에 해당)

```
It's your turn to guess!

Guess 0, 1 or 2: 2
You guessed 2.

You guessed correctly!
The computer hid 17 marbles.

-----
USER: ○ (22)
COMP: ● (18)
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
○○●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
-----
```

예시 2) 유저가 틀린 경우

```
It's your turn to guess!

Guess 0, 1 or 2: 4
That is a wrong input!
Please guess 0, 1 or 2: 1
You guessed 2.

You guessed incorrectly...
The computer hid 18 marbles.
The correct guess would have been 0.

-----
USER: ○ ( 2)
COMP: ● (38)
○○●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
-----
```

#### 4. 유저가 구슬을 숨길 차례

- 유저의 구슬 숨기기:

모든 짝수 턴에는 유저가 구슬을 숨기게 됩니다. 이때 사용자로부터 숨길 구슬의 수는 아래 설명하는 `get_user_choice()`를 사용하여 받도록 합니다.

➤ `int get_user_choice(int n):`

유저로부터 숨길 구슬의 개수를 입력 받아 반환하는 하는 함수입니다 (`scanf()` 사용). 우선 이번에는 컴퓨터가 추측을 할 차례라고 설명을 한 후 구슬을 몇 개

숨기고 싶은 지를 입력하라고 지시합니다. 이때 사용자가 숨길 수 있는 범위 내의 숫자를 입력하게 되면, 해당 입력 값을 출력 후 반환합니다.

**예외처리)** 유저 입력에서 유저가 0 이하 또는 보유하고 있는 구슬의 개수를 초과하는 숫자를 입력하면 아래의 예시처럼 에러 메시지를 출력 후, 올바른 범위 내의 입력을 받을 때까지 반복하여 입력을 받습니다.

이 함수 구현 시, `is_valid_num()` 함수를 호출하여 사용하도록 합니다.

- 컴퓨터의 추측 값 생성:

컴퓨터는 사용자가 숨긴 구슬의 숫자를 3으로 나누었을 때의 나머지를 추측하여 맞춰야 합니다. 이때 컴퓨터의 추측 값은 0, 1, 또는 2 중 임의로 선택이 되도록 구현합니다

- 결과 확인:

컴퓨터의 추측 값이 정해진 이후에는 컴퓨터의 추측이 무엇인지를 출력한 후 해당 추측이 맞았는지를 확인하고 다음과 같이 유저에게 결과를 알려줍니다:

- 컴퓨터가 맞은 경우: 컴퓨터가 맞았다는 메시지를 출력한 후 유저가 숨겼던 구슬의 개수만큼 유저의 구슬을 차감하는 동시에 컴퓨터의 구슬을 증가합니다.
- 컴퓨터가 틀린 경우: 컴퓨터가 틀렸다는 메시지와 함께 정답을 공개한 후 유저가 숨겼던 구슬의 개수만큼 유저의 구슬을 증가하는 동시에 컴퓨터의 구슬을 차감합니다.

- 구슬 보유 상태 출력:

턴이 끝나면 `print_marble_status()` 를 사용해 현재 각 플레이어가 보유하고 있는 구슬의 개수를 보여줍니다.

- 구슬의 개수가 0 개 이하인 플레이어가 있으면 게임을 끝냅니다.

- 유저가 구슬을 숨길 차례일 때 게임화면 예시:

예시 1) 컴퓨터가 맞은 경우

```
It's the computer's turn to guess!

How many marbles would you like to hide? 13
You hid 13 marbles.

The computer guessed 1.
The computer guessed correctly...

-----
USER: ○ (24)
COMP: ● (16)
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
○○○○●●●●●●●●●●●●●●●●●●●●●●●●●●●●
-----
```

## 예시 2) 컴퓨터가 틀린 경우

```
It's the computer's turn to guess!

How many marbles would you like to hide? 0
That is a wrong input!
Please select a number between 1 and 16: 14
You hid 14 marbles.

The computer guessed 1.
The computer guessed incorrectly!
The correct guess would have been 2.

-----
USER: ○ (30)
COMP: ● (10)
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○●●●●●●●●●●●●●●●●●●●●●●
-----
```

## 5. 게임 종료화면

- 게임이 끝나게 되면 게임이 끝났다는 메시지와 함께 누가 이겼는지를 알립니다.
- 이후 유저에게 게임을 다시하고 싶은지를 물어봅니다. 이때 유저가 ‘y’를 입력하면 게임은 다시 시작되고 프로그램은 다시 위 ‘2. 게임 시작화면’으로 돌아가게 됩니다.
- 만약 유저가 ‘y’가 아닌 다른 입력을 한다면 프로그램 종료화면 (아래 ‘6. 종료화면’)으로 넘어갑니다.
- 만약 마지막으로 끝난 게임이 9 번째 게임이라면 유저에게 게임의 재개 여부를 묻지 않고 곧장 프로그램 종료화면(아래 ‘6. 종료화면’)으로 넘어갑니다.
- 게임 종료화면의 예시:

### 예시 1) 유저가 이긴 경우

```
GAME OVER!
You have won! :)

Would you like to play again?
Type 'y' to play again:
```

### 예시 2) 유저가 진 경우

```
GAME OVER!
You have lost to the computer :(

Would you like to play again?
Type 'y' to play again:
```

## 6. 프로그램 종료화면

- 프로그램이 끝나기 직전에 유저가 보는 마지막 화면은 아래와 같습니다.  
다음 요소들이 반드시 포함되어 있어야 합니다:



- 프로그램 종료화면의 예시:

You have won 2 games out of 3!  
Thank you for playing! Bye~!

계속하려면 아무 키나 누르십시오. . .

### III. 프로그램 실행예시

다음은 012 구슬게임 프로그램의 실행 예시입니다. (빨간색 밑줄은 사용자 입력에 해당)

```
[CSED101 Assignment 2]
#####
#####
#####
#####      Marble Game      #####
#####
#####
#####

Welcome to the Marble Game! Have fun~!
#####

[Game #1]
-----
USER: ○ (20)
COMP: ● (20)
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
-----

It's your turn to guess!

Guess 0, 1 or 2: 3
That is a wrong input!
Please guess 0, 1 or 2: 0
You guessed 0.

You guessed incorrectly...
The computer hid 16 marbles.
The correct guess would have been 1.

-----
USER: ○ ( 4)
COMP: ● (36)
○○○○●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
-----

It's the computer's turn to guess!

How many marbles would you like to hide? 0
That is a wrong input!
```

```
The computer guessed 0.  
The computer guessed incorrectly!  
The correct guess would have been 1.
```

It's your turn to guess!

```
You guessed correctly!  
The computer hid 4 marbles.
```

It's the computer's turn to guess!

```
The computer guessed 0.  
The computer guessed correctly...
```

```
GAME OVER!  
You have lost to the computer :(
```

10

