

2023년 2학기

# 프로그래밍과 문제해결

## Assignment #2

담당 교수: 윤은영

학번: 20230499

학과: 무은재학부

이름: 김재환

POVIS ID: carotinoid

---

명에서약(Honor code)

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

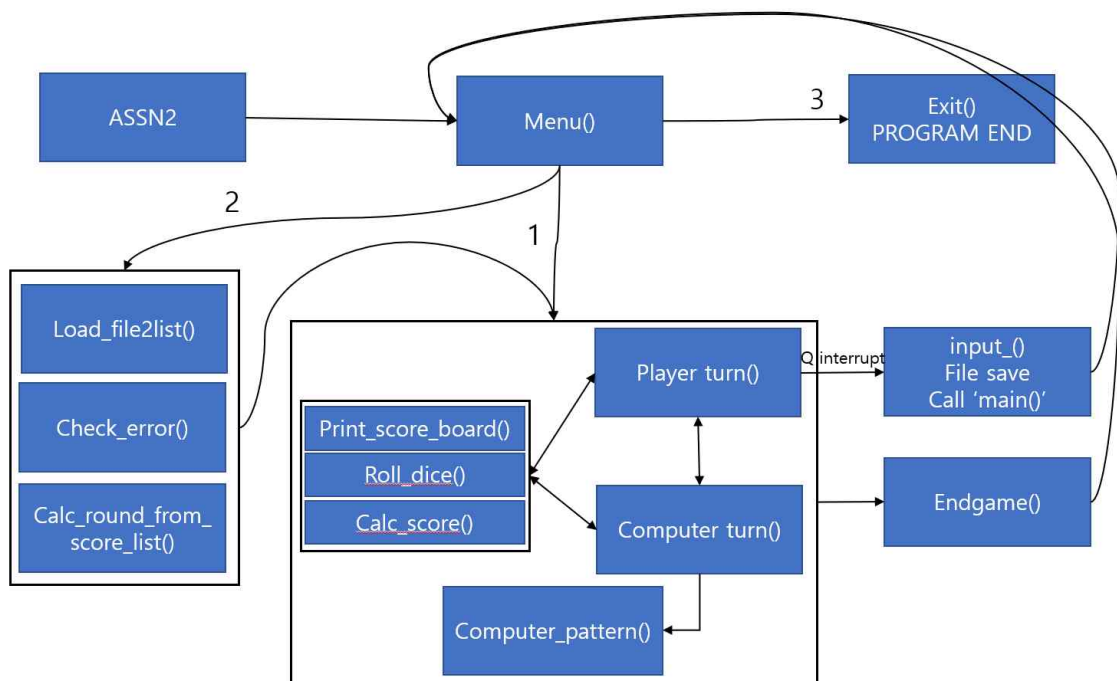
---

## 1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- 게임의 이름은 요트 다이스로, 12 라운드 동안 진행되며 한 라운드에 플레이어와 컴퓨터가 각 턴을 가진다.
- 자신의 턴에서, 총 3번 주사위를 굴릴 수 있으며, 2, 3번째 굴릴 때에는 원하는 주사위만 굴러 다시 굴릴 수 있다. 3번 주사위를 굴린 후, 12가지의 카테고리 중 원하는 카테고리를 선택하여, 그 카테고리의 기준에 맞는 점수를 매겨 적어넣는다. 이때 한번 선택한 카테고리는 선택할 수 없다. 이후 모든 라운드 뒤에는 최종 점수를 계산하여 승패를 가린다.
- 게임을 진행하면서 언제든지 'Q'를 입력하여 파일로 저장하면서 게임을 중단할 수 있다. 또한 적절한 형태의 파일로부터 게임을 불러와 이어서 진행할 수 있다.

이때 사용되는 구조 차트는 다음과 같이 나타낼 수 있다.



## 2. 알고리즘

본 프로그램을 작성하기 위해서 특별히 고려해야 하는 부분은 크게 3가지로 나눌 수 있다. 컴퓨터가 최선의 점수를 얻기 위한 알고리즘 `computer_pattern()`, 입력받은 파일의 오류를 검사하는 `check_error()`, Q를 눌러 파일을 저장하고 중단하는 `input_()`. 그러나 `player_turn` 과 `computer_turn` 또한 핵심 함수이지만 작동 방식은 복잡하지 않기에 생략하였다.

### Pseudo-algorithm for entire program.

// 프로그램에 필요한 변수들은 미리 선언해 놓은 것으로 가정한다.

```
1  print menu and input selected
2  if selected == 1
3      for i in range(0, 12)
4          player_turn()
5          computer_turn()
6      endgame()
7  if selected == 2
8      get filename and load_file2list()
9      file check_error()
10     calculate remain rounds
11     for i in range(rounds, 12)
12         player_turn()
13         computer_turn()
14     endgame()
15  if selected == 3
16     exit(0)
17  Repeat.
```

### Pseudo-algorithm for computer\_pattern()

// 프로그램에 필요한 변수들은 미리 선언해 놓은 것으로 가정한다.  
// get dice\_set and score\_list as parameter

```
1  computed_score = [-1 for i in range(0, 12)] # -1 : not considered
2  compute all score about each category that not filled
3  if round < 12 and sum(dice_set) < 20: score of 'C' = -1
4  Cate_index = computed_score.index(max(computed_score))
5  if 0 <= Cate_index <= 5: # 1~6
6      sel_reroll <- list: dice_set[i] s.t. dice_set[i] != int(category)
7  elif Cate_index == 6: # C
8      sel_reroll <- list: dice_set[i] s.t. dice_set[i] <= 3
9  elif Cate_index == 7, 8, 9, 10, 11, : # 4K, FH, SS, LS, Y
10     if satisfy each category's criteria: sel_reroll = []
11     if not satisfy : sel_reroll = [1, 2, 3, 4, 5]
12  return category, sel_reroll
```

---

**Pseudo-algorithm for check\_error()**

---

```
// 프로그램에 필요한 변수들은 미리 선언해 놓은 것으로 가정한다.
// get score_list as parameter, j = 0 and 1
1  err = False
2  if length of score_list[0], score_list[1]: err = True
3  check2_score_list = [[i if i != ' ' else '0' for i in j] for j in score_list]
4  if check2_score_list[j][i] is not form of str(int): error=True # use try except
5  if not err:
6      check_score_list=[[int(i) if i!=' ' else 0 for i in j] for j in score_list]
7      if player's round and computer's round is different: err = True
8      check_num <- list: each criteria of each category
9      for i in range(0, 12):
10         if not check_score_list[j][i] in check_num[i]: err = True
11  return err
```

---

---

**Pseudo-algorithm for input\_0**

---

```
// 프로그램에 필요한 변수들은 미리 선언해 놓은 것으로 가정한다.
// get score_list, text="(default) as parameter
1  i = input(text).upper()
2  if i == Q:
3      while True:
4          input fn / err = False
5          if fn == '': err = True
6          if fn is consist of only '.': err = True
7          if fn has banned_letter: err = True
8          if err:
9              print wrong input, continue
10         with open(fn, 'w') as fw:
11             for i in range(0, 12):
12                 p, c = score_list[j][i]
13                 if there is not score_list[j][i], p / c = 'x'
14                 data = "{category}: {p} {c}\n"
15                 fw.write(data)
16             print 'File saved'
17         main()
18  else: return i
```

---

### 3. 프로그램 구조 및 설명

#### a) 메인메뉴

- 게임 새로 시작 / 게임 불러오기 / 프로그램 종료 3가지 항목을 선택할 수 있으며, 입력이 잘못되면 다시 입력받는다. 1을 입력받으면 새로 게임을 시작한다. 2를 입력받으면 유효한 파일명을 입력받아 파일로부터 게임을 불러와 이어서 시작한다. 3을 입력받으면 프로그램을 종료한다.

#### b) 플레이어 턴

- 라운드를 포함해 초기 정보를 출력하고, 주사위를 5개 굴려 나온 5개의 수를 출력한다. 그다음 다시 굴릴 주사위를 물어보고, 입력받은 번호의 주사위를 다시 굴리는 과정을 2회 반복한다. 최종 주사위 쌍을 정렬해 출력한 후, 카테고리를 입력받아 카테고리에 맞는 과정으로 점수를 계산하여 기록한다.
- 다시 굴릴 주사위를 입력받을 때 1에서 5 사이 정수가 아닌 정수는 입력을 무시하며, 문자열이나 소수 등 다른 자료형이 입력될 시 Wrong input을 출력하고 다시 입력받는다. 아무것도 입력하지 않고 엔터를 누르면 다시 굴릴 기회를 1회 차감한다.
- 카테고리를 입력할 때 이미 기록된 카테고리를 입력하거나 없는 카테고리를 입력하면 Wrong input을 출력하고 다시 입력받는다.

#### c) 컴퓨터 턴 & 컴퓨터 패턴

- 컴퓨터 턴 또한 플레이어 턴과 비슷한 순서로 진행된다. 대신, 플레이어가 굴릴 주사위를 선택하는 것 대신, 컴퓨터 턴에서는 컴퓨터 패턴을 호출하여 굴릴 주사위를 결정한다. 컴퓨터 패턴 함수에서는 주사위 셋과 스코어 리스트를 매개변수로 전달받은 후, 스코어 리스트에 채워지지 않았던 카테고리에 대해서 점수를 계산한 후 최대 점수를 얻을 수 있는 카테고리를 선택한다. 그리고 그 카테고리에 맞게 다시 굴릴 주사위를 선택한 후 반환한다. 이때 카테고리 C의 경우는 총합이 20 미만일 때는 무시한다.

#### d) 게임 중단 및 파일 저장

- 플레이어로부터 입력받을 때 input함수를 변형한 input\_ 함수를 사용하였다. input\_은 스코어 리스트와 문자열을 매개변수로 받는다. 받은 문자열을 출력하면서 입력받는다. 입력값이 Q가 아니면 문자열을 출력하고 리턴한다. 입력값이 Q라면 저장할 파일명을 받아 파일을 저장한 후 main()을 호출해 메뉴로 돌아간다.

#### e) 파일 입력

- 메뉴에서 2를 입력받으면 파일명을 입력받고 저장된 파일로부터 게임 상황을 불러온다. 입력된 파일이 존재하지 않거나 유효하지 않은 파일일 경우 그에 따른 오류 메시지를 출력한다. 유효하지 않은 파일은 카테고리가 모두 작성되지 않거나, 잘못된 카테고리가 적

혀있거나, 해당 카테고리에서 가질 수 없는 점수가 적혀있을 경우 등이 포함된다.

f) 스코어보드 출력

- 점수판을 출력할 때 몇 가지 유의할 점이 있다.
- 작성되지 않았을 경우 아무것도 출력되지 않아야 한다. 이는 "을 저장해 출력하는 것으로 구현했으며, 점수판이 어그러지지 않게 하려고 %2s를 사용하였다. 전체 점수는 100점을 넘을 수 있으므로 %3s를 사용하였다.
- 보너스 점수같은 경우 처음에는 빈칸으로 있다가, 1~6의 점수의 합이 63점이 넘을 경우 35를 출력한다. 63점이 넘지 않았더라도 1~6이 모두 채워진 경우 0을 출력해야한다.

## 4. 프로그래밍 실행 방법 및 예시

(1) 처음에 프로그램을 실행하면 메뉴(menu)가 출력되며, 메뉴 중 하나의 항목을 입력받는다. 1, 2, 3이 아닌 다른 값을 입력하면 Wrong Input! 문구를 출력하고 다시 입력받는다. 3을 입력하면 프로그램 종료 문구가 출력되며, 종료된다.

(2) 메뉴에서 1을 입력받으면 Starting a game... 문구와 함께 스코어 보드가 출력되며, 플레이어 턴이 진행된다. 플레이어 턴에는 현재 라운드와 처음 굴린 주사위 결과를 출력하며, 어느 주사위를 굴릴지 질문한다.

(3) 이때 1, 2, 3, 4, 5 이외에 정수는 입력을 무시하며, 문자열 등이 입력될 시 Wrong input! 문구를 출력하며 다시 입력받는다. 2번 다시 굴리고 난 후, 주사위를 정렬하여 Sorted Roll: [] 형태로 출력한 후, 카테고리를 입력받는다. 정해진 카테고리가 아닌 다른 문구(FY)를 입력될 시 Wrong input! 문구를 출력한다. 알맞은 카테고리를 입력받았다면 점수를 계산하여 스코어 보드를 출력한다.

(4) 이후 자동으로 컴퓨터의 턴이 진행된다. 그림에서는 처음에 2, 2, 6, 5, 5가 나와서 최대 득점으로 C를 선택했고, 이후 2, 2가 3, 3이 나왔다. 다시 C를 선택해서 1, 2번째 주사위를 다시 굴려 최종 점수로 5, 6, 6, 5, 5를 얻었다. 이때 C와 FH일 때 얻는 점수가 가장 높으며, 그중 하나인 C가 선택되었다. 이후 다시 플레이어 턴으로 넘어가며, 플레이어와 컴퓨터가 턴을 반복한다.

(5) 모든 턴이 진행되고 나면 <Final Score Board>를 출력하며, 승자를 결정하여 출력한다. 이때 플레이어와 컴퓨터 모두 1~6의 합이 63을 넘지 못했기 때문에 보너스 칸에 0이 출력된 것을 볼 수 있다. 이후 Press Enter to continue...문구를 출력하고 입력을 받으며, 엔터를 누르면 다시 (1)의 메인 화면으로 돌아간다.

<p>(1)</p> <pre> [Yacht Dice] ----- 1. New Game 2. Load Game 3. Exit ----- Select a menu: 3 Program ended. Bye!  [Yacht Dice] ----- 1. New Game 2. Load Game 3. Exit ----- Select a menu: asd Wrong Input!  Select a menu: █ </pre>	<p>(2)</p> <pre> [Yacht Dice] ----- 1. New Game 2. Load Game 3. Exit ----- Select a menu: 1  Starting a game... </pre> <table border="1"> <thead> <tr> <th>Player</th><th>Computer</th></tr> </thead> <tbody> <tr><td>1:</td><td>1:</td></tr> <tr><td>2:</td><td>2:</td></tr> <tr><td>3:</td><td>3:</td></tr> <tr><td>4:</td><td>4:</td></tr> <tr><td>5:</td><td>5:</td></tr> <tr><td>6:</td><td>6:</td></tr> <tr> <td>Sub total: 0/63 +35 bonus:</td><td>Sub total: 0/63 +35 bonus:</td></tr> <tr> <td>C:</td><td>C:</td></tr> <tr> <td>4K: FH: SS: LS: Yacht:</td><td>4K: FH: SS: LS: Yacht:</td></tr> <tr> <td>Total: 0</td><td>Total: 0</td></tr> </tbody> </table> <pre> [Player's Turn (1/12)] Roll: [4, 5, 2, 5, 6] Which dice to reroll (1~5)? n </pre>	Player	Computer	1:	1:	2:	2:	3:	3:	4:	4:	5:	5:	6:	6:	Sub total: 0/63 +35 bonus:	Sub total: 0/63 +35 bonus:	C:	C:	4K: FH: SS: LS: Yacht:	4K: FH: SS: LS: Yacht:	Total: 0	Total: 0																						
Player	Computer																																												
1:	1:																																												
2:	2:																																												
3:	3:																																												
4:	4:																																												
5:	5:																																												
6:	6:																																												
Sub total: 0/63 +35 bonus:	Sub total: 0/63 +35 bonus:																																												
C:	C:																																												
4K: FH: SS: LS: Yacht:	4K: FH: SS: LS: Yacht:																																												
Total: 0	Total: 0																																												
<p>(3)</p> <pre> [Player's Turn (1/12)] Roll: [4, 5, 2, 5, 6] Which dice to reroll (1~5)? n Wrong input! Which dice to reroll (1~5)? 1 6 Roll: [1, 5, 2, 5, 6] Which dice to reroll (1~5)? Roll: [1, 5, 2, 5, 6]  Sorted Roll: [1, 2, 5, 5, 6] Choose a category: FY Wrong Input! Choose a category: 5 </pre> <table border="1"> <thead> <tr> <th>Player</th><th>Computer</th></tr> </thead> <tbody> <tr><td>1:</td><td>1:</td></tr> <tr><td>2:</td><td>2:</td></tr> <tr><td>3:</td><td>3:</td></tr> <tr><td>4:</td><td>4:</td></tr> <tr><td>5: 10</td><td>5:</td></tr> <tr><td>6:</td><td>6:</td></tr> <tr> <td>Sub total: 10/63 +35 bonus:</td><td>Sub total: 0/63 +35 bonus:</td></tr> <tr> <td>C:</td><td>C:</td></tr> <tr> <td>4K: FH: SS: LS: Yacht:</td><td>4K: FH: SS: LS: Yacht:</td></tr> <tr> <td>Total: 10</td><td>Total: 0</td></tr> </tbody> </table>	Player	Computer	1:	1:	2:	2:	3:	3:	4:	4:	5: 10	5:	6:	6:	Sub total: 10/63 +35 bonus:	Sub total: 0/63 +35 bonus:	C:	C:	4K: FH: SS: LS: Yacht:	4K: FH: SS: LS: Yacht:	Total: 10	Total: 0	<p>(4)</p> <pre> [Computer's Turn (1/12)] Roll: [2, 2, 6, 5, 5] Which dice to reroll (1~5)? 1 2 Roll: [3, 3, 6, 5, 5] Which dice to reroll (1~5)? 1 2 Roll: [5, 6, 6, 5, 5]  Sorted Roll: [5, 5, 5, 6, 6] Choose a category: C </pre> <table border="1"> <thead> <tr> <th>Player</th><th>Computer</th></tr> </thead> <tbody> <tr><td>1:</td><td>1:</td></tr> <tr><td>2:</td><td>2:</td></tr> <tr><td>3:</td><td>3:</td></tr> <tr><td>4:</td><td>4:</td></tr> <tr><td>5: 10</td><td>5:</td></tr> <tr><td>6:</td><td>6:</td></tr> <tr> <td>Sub total: 10/63 +35 bonus:</td><td>Sub total: 0/63 +35 bonus:</td></tr> <tr> <td>C:</td><td>C: 27</td></tr> <tr> <td>4K: FH: SS: LS: Yacht:</td><td>4K: FH: SS: LS: Yacht:</td></tr> <tr> <td>Total: 10</td><td>Total: 27</td></tr> </tbody> </table>	Player	Computer	1:	1:	2:	2:	3:	3:	4:	4:	5: 10	5:	6:	6:	Sub total: 10/63 +35 bonus:	Sub total: 0/63 +35 bonus:	C:	C: 27	4K: FH: SS: LS: Yacht:	4K: FH: SS: LS: Yacht:	Total: 10	Total: 27
Player	Computer																																												
1:	1:																																												
2:	2:																																												
3:	3:																																												
4:	4:																																												
5: 10	5:																																												
6:	6:																																												
Sub total: 10/63 +35 bonus:	Sub total: 0/63 +35 bonus:																																												
C:	C:																																												
4K: FH: SS: LS: Yacht:	4K: FH: SS: LS: Yacht:																																												
Total: 10	Total: 0																																												
Player	Computer																																												
1:	1:																																												
2:	2:																																												
3:	3:																																												
4:	4:																																												
5: 10	5:																																												
6:	6:																																												
Sub total: 10/63 +35 bonus:	Sub total: 0/63 +35 bonus:																																												
C:	C: 27																																												
4K: FH: SS: LS: Yacht:	4K: FH: SS: LS: Yacht:																																												
Total: 10	Total: 27																																												

(6) 플레이어 턴의 입력 받는 상황에서 Q를 입력시(소문자도 가능함), 게임을 멈추고 저장할 파일 이름을 입력받는다. 이때 입력값이 ""(입력 없이 엔터를 친 경우), 마침표로만 이루어진 경우, ₩, /, :, \*, ?, ", <, >, |와 같이 OS나 경로에 사용되는 문자열이 포함된 경우에 Wrong input! 문구를 출력하며, 정상적인 파일 이름이 입력될시 File saved 문구를 출력하고 다시 메뉴로 돌아간다.

(7) 메뉴에서 2를 입력할 시, 불러올 파일의 이름을 입력받는다. 이때 입력받은 파일이 존재하지 않으면 File does not exist. 문구와 함께 다시 입력받는다. 입력받은 파일이 유효하지 않은 파일이라면 Invalid file content. 문구를 출력하며 다시 입력받는다. 정상적인 파일 이름을 입력받으면 파일에 저장된 정보에 따라 스코어보드를 출력하며, 남은 라운드를 이어서 진행한다.

(8) 플레이어 턴에서 카테고리를 입력받을 때, 이미 채운 카테고리를 입력하면 Wrong Input!을 출력한다.



<p>(5)</p> <pre> &lt;Final Score Board&gt;    Player      Computer   1:          3      1:          2   2:          2      2:          4   3:          9      3:          3   4:          0      4:          4   5:         10      5:         10   6:         18      6:         12  Sub total: 42/63   Sub total: 35/63 +35 bonus:  0     +35 bonus:  0  C:           21     C:           23  4K:           0     4K:          26 FH:          24     FH:          28 SS:          15     SS:          15 LS:          30     LS:           0 Yacht:        0     Yacht:        0  Total:       132    Total:       127  You win!  Press Enter to continue... </pre>	<p>(6)</p> <pre> [Player's Turn (1/12)] Roll: [2, 5, 5, 1, 6] Which dice to reroll (1~5)? Q  Game paused. Enter the filename to save: /as Wrong input!  Game paused. Enter the filename to save: Wrong input!  Game paused. Enter the filename to save: testing.txt File saved  [Yacht Dice] ----- 1. New Game 2. Load Game 3. Exit ----- Select a menu: </pre>
<p>(7)</p> <pre> [Yacht Dice] ----- 1. New Game 2. Load Game 3. Exit ----- Select a menu: 2 Enter filename to load: asdf File does not exist.  Enter filename to load: test_w1.txt Invalid file content.  Enter filename to load: good.txt    Player      Computer   1:          3      1:          6   2:          2      2:          12   3:          6      3:          10   4:          12     4:          12   5:         15     5:          10   6:         12     6:          12  Sub total: 30/63   Sub total: 40/63 +35 bonus:        +35 bonus:  C:           23     C:           24  4K:           0     4K:          22 FH:          18     FH:          15 SS:           0     SS:           0 LS:          30     LS:           0 Yacht:        50     Yacht:        50  Total:       151    Total:       151  [Player's Turn (9/12)] Roll: [5, 2, 5, 5, 2] Which dice to reroll (1~5)? </pre>	<p>(8)</p> <pre>   Player      Computer   1:          3      1:          4   2:          2      2:          3   3:          9      3:          4   4:          4      4:          4   5:         10      5:         10   6:         18      6:         12  Sub total: 40/63   Sub total: 33/63 +35 bonus:        +35 bonus:  C:           21     C:           23  4K:           0     4K:          26 FH:          24     FH:          28 SS:          15     SS:          15 LS:          30     LS:           0 Yacht:        0     Yacht:        0  Total:       130    Total:       125  [Player's Turn (10/12)] Roll: [6, 1, 2, 4, 6] Which dice to reroll (1~5)? 2 3 4 Roll: [6, 2, 1, 1, 6] Which dice to reroll (1~5)? 2 3 4 Roll: [6, 1, 4, 2, 6]  Sorted Roll: [1, 2, 4, 6, 6] Choose a category: 3 Wrong Input! Choose a category: a Wrong Input! Choose a category: </pre>

## 5. 토론

- 입력할 때 Q가 들어올 시 언제든지 프로그램을 중단하고 파일을 저장하는 기능을 구현하기 위하여, input 함수를 변형하여 input\_ 함수를 만들어 사용하였다.
- 파일이 유효한지 검사하는 과정에서 많은 어려움을 겪었다. 카테고리의 순서가 달라도 잘 입력받기 위해서 미리 12줄의 리스트를 선언했었는데, 그러면 빠진 카테고리가 있더라도 len()에서 12가 나오게 되는 문제가 있었다. 이를 해결하기 위해서 체크할 에러 순서를 적절히 선택해야 했다.
- 파일을 저장할 때, 문제 파일에서 제시되어 있진 않았지만, 파일로 존재할 수 없는 형식인지 검사하였다. 윈도우 운영체제 기준으로 빈칸, 마침표로만 이루어진 파일, 몇 특수 문자가 포함된 문자는 사용될 수 없다.
- 일반적으로 주사위 5개의 눈 수의 합은 5에서 30까지 가능하지만, 모든 수를 더하는 경우 중 하나인 폴하우스인 경우는 6과 29가 될 수 없다.
- 'FH'와 같은 카테고리의 문자열과, 카테고리의 인덱스(FH의 경우 8)를 서로 변환해야 할 때가 많았다. 이를 위해서 각 경우를 딕셔너리로 전역변수에 저장하여 사용하였다.

## 6. 결론

- 본 과제에서는 프로그램을 작성하면서 사용되는 튜플, 딕셔너리와 같은 여러 자료형을 사용하는 방법을 익혔고, 파일을 입력 및 저장하면서 적절한 형태로 가공하는 방법을 학습할 수 있었다. 함수의 디폴트 매개변수를 사용함으로써 하나의 함수가 할 수 있는 일을 확장할 수 있었으며, 적절한 형태의 리턴 값을 사용해 코드를 단순화 하였다. 또한 복잡한 코드에서 오류를 찾아내기 위하여 디버깅 기능을 공부하였다.

## 7. 개선 방향

- 컴퓨터 차례일 때 컴퓨터가 주어진 상황(주사위)에서 더 높은 점수와 그에 따른 가능성을 바탕으로 최종적으로 높은 점수를 얻을 수 있게 하는 알고리즘(computer\_pattern)을 개선하면 더욱 난이도 있는 게임을 만들 수 있을 것이다.
- 주어진 조건, 함수의 매개변수까지 정확하게 따르면서 효과적으로 함수 간 데이터를 주고받는 방법에 대하여 생각해 볼 필요성이 있다고 느꼈다. 개인적으로 전역변수와 global 키워드를 사용하는 것은 변수명을 매우 헷갈리게 해 오류를 만들어 내기 때문에 지양하는 편이나, input()을 변형하여 만든 함수인 input\_() 같은 경우 매개변수로 score\_list를 전달받지 않으면서도 그에 대한 데이터를 접근할 수 있어야 가시성이 뛰어났다. 이러한 점에서 더 쉬우면서도 편리한 코드를 작성하는 방법에 대하여 고민하게 되었다.