

CSED101. Programming & Problem solving

Fall 2023

Programming Assignment #2 (75 points)

송영길 (songyk@postech.ac.kr)

■ 제출 마감일: 2023.11.15 23:59

■ 파이썬 버전: Python 3.x

■ 제출물

- .py 소스 코드 (assn2.py)
 - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙여 주세요.
- 보고서 파일 (.docx, .hwp 또는 .pdf; assn2.docx, assn2.hwp 또는 assn2.pdf)
 - AssnReadMe.pdf 를 참조하여 작성하시면 됩니다.
 - 프로그램 실행화면을 캡처하여 보고서에 포함시키고, 간단히 설명해 주세요.
 - 명예서약(Honor code): 표지에 다음의 내용을 포함하여 제출해 주세요. "나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다." 보고서 표지에 명예서약이 없는 경우는 과제를 제출하지 않은 것으로 처리됩니다.
- 소스코드와 보고서 파일을 PLMS를 통하여 제출해 주세요.
 - PLMS에 제출할 때는 소스파일과 보고서를 압축하지 말고 각각 업로드 해주시면 됩니다.

■ 주의사항

- 구문 오류(Syntax Error)가 발생하거나 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).
늦은 제출시 PLMS에 기록된 최종 수정일시를 기준으로 감점합니다.
- 각 문제의 제한 조건과 요구 사항을 반드시 지켜 주시기 바랍니다.
- 모든 문제의 출력 형식은 채점을 위해 아래에 제시된 예시들과 최대한 비슷하게 작성해 주세요.
- 각 문제에 명시된 예외 처리 외에는 고려하지 않아도 됩니다.
- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 ‘POSTECH 전자컴퓨터공학부 부정행위 정의’를 따릅니다. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 disciplinary.pdf를 참조할 것.)

■ Problem: 요트 다이스 (Yacht dice)

(문제)

요트 다이스는 주사위 놀이 게임으로 5개의 주사위를 사용하며 목표는 미리 정해진 주사위 조합으로 최고 점수를 얻는 것입니다.

본 과제의 요트 다이스는 두 명의 플레이어(사용자, 컴퓨터)가 게임을 하게 됩니다. 두 명의 플레이어가 번갈아 가면서 자신의 순서에, 게임 규칙에 따라 주사위를 던져 자신의 점수판에서 항목을 선택하여 점수를 기록 하게 됩니다. 모든 플레이어가 점수판을 모두 채우면 게임이 종료되며, 점수판에 기록한 점수의 총합이 높은 플레이어가 승리합니다.

[게임 규칙]

1. 각 플레이어는 자신의 순서에 5개의 주사위를 던집니다.
2. 자신의 순서에 총 3회까지 던질 수 있으며, 원하는 눈이 나온 주사위는 두고 나머지만 다시 던질 수 있습니다.
3. 3회 굴린 주사위 조합으로 자신의 점수판에 항목을 선택하여 점수를 기록합니다.
 - 이미 선택한 항목에는 다시 기록할 수 없습니다.
 - 조합과 맞는 항목이 없어도 12개의 항목 중 1개를 선택해야 합니다.
 이때 점수는 0점으로 기록됩니다.
4. 총 12라운드(점수판은 12칸)를 하면 게임이 끝나게 되며, 점수판의 총합으로 승패를 정합니다.

기본적으로 이번 과제에서 사용되는 점수판의 항목과 주사위 점수 기준은 다음과 같습니다.

[주사위 점수 기준]

항목	점수 계산 방법	예시
1	Aces : 1 이 나온 주사위 눈의 총합 (최대 5 점)	[1, 1, 1, 1, 4] = 4 점
2	Deuces: 2 가 나온 주사위 눈의 총합 (최대 10 점)	[2, 2, 2, 3, 6] = 6 점
3	Threes: 3 이 나온 주사위 눈의 총합 (최대 15 점)	[1, 3, 3, 3, 5] = 9 점
4	Fours : 4 가 나온 주사위 눈의 총합 (최대 20 점)	[1, 1, 1, 4, 4] = 8 점
5	Fives : 5 가 나온 주사위 눈의 총합 (최대 25 점)	[2, 3, 4, 5, 6] = 5 점
6	Sixes : 6 이 나온 주사위 눈의 총합 (최대 30 점)	[2, 3, 6, 6, 6] = 18 점
	보너스: 위 항목의 점수 합계가 63 점 이상일 때 보너스 점수 35 점 획득	
C	Choice : 모든 주사위 눈의 합 (최대 30 점)	[1, 1, 2, 5, 5] = 14 점
4K	4 of a Kind : 동일한 주사위 눈이 4 개 이상 일 때, 모든 주사위 눈의 합 (최대 30 점)	[5, 5, 5, 5, 6] = 26 점 [1, 1, 2, 2, 2] = 0 점
FH	Full House : 동일한 주사위가 각각 3 개, 2 개가 있을 때 모든 주사위 눈의 합 (최대 30 점)	[5, 5, 6, 6, 6] = 28 점 [6, 6, 6, 6, 6] = 30 점 [1, 2, 3, 3, 3] = 0 점
SS	Small Straight: 연속된 주사위 눈이 4 개이상일 때 (15 점)	[1, 3, 4, 5, 6] = 15 점 [2, 3, 3, 5, 6] = 0 점
LS	Large Straight: 연속된 주사위 눈이 5 개일 때 (30 점)	[1, 2, 3, 4, 5] = 30 점 [1, 1, 2, 2, 2] = 0 점
Y	Yacht : 동일한 주사위 눈이 5 개일 때 (50 점)	[1, 1, 1, 1, 1] = 50 점 [1, 1, 1, 1, 4] = 0 점

[점수판]

Turn 1/12		
Categories		
■ Aces		
■ Deuces		
■ Threes		
■ Fours		
■ Fives		
■ Sixes		
Subtotal	0/63	0/63
+35 Bonus		
Bonus if ■ ■ are over 63 points		
■ Choice		
■ 4 of a Kind		
■ Full House		
■ S. Straight		
■ L. Straight		
■ Yacht		
Total	0	0

[목적]

이번 과제를 통하여

1. 2차원 리스트 사용법을 익힙니다.
2. 텍스트를 읽고 활용하는 능력을 기릅니다.
3. 조건문, 반복문, 사용자 정의 함수 및 랜덤 함수 사용법을 익힙니다.

[주의사항]

- (1) 이번 과제는 2차원 리스트 사용을 위한 과제입니다. **두 플레이어의 점수를 반드시 2차원 리스트로 생성하여 활용해야 합니다.**
- (2) 문서에 반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인 후 구현하시면 됩니다.
 - 정의해서 사용해야 할 사용자 정의 함수의 매개변수는 자유롭게 변형하여 사용할 수 있습니다.
- (3) 본 과제에서는 앞선 과제에서의 `clear_screen()` 함수의 사용을 의무화하지는 않았습니다. 하지만, 중간중간 적절히 사용해 주세요.
- (4) `int`, `float`, `bool`, `str`, `list`, `tuple`, `dictionary`, `set` 자료형까지 수업시간에 다룬 내용에 한해서 사용 가능합니다.
- (5) 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해야 합니다.

[설명 및 요구사항]

0. 프로그램 초기화면

프로그램을 실행하면 아래와 같이 사용자가 선택할 수 있는 메뉴가 출력되고, 사용자로부터 3가지 선택사항 중 하나를 입력 받을 준비를 합니다.

```
[Yacht Dice]
-----
1. New Game  2. Load Game  3. Exit
-----
Select a menu:
```

그림 1. 초기 메뉴 화면

1을 선택하면 새 게임으로, 2를 선택하면 저장된 파일로부터 게임판을 생성하여 게임을 진행하며, 3을 선택하면 그림 2와 같이 간단한 메시지 출력 후, 프로그램을 종료합니다.

예외 처리) 1, 2, 3외의 입력에 대해서는 Wrong Input! 메시지 출력 후 아래와 같이 다시 입력 받습니다.

(실행 예시의 빨간색 밑줄은 사용자 입력에 해당)

```
[Yacht Dice]
-----
1. New Game  2. Load Game  3. Exit
-----
Select a menu: 4
Wrong Input!

Select a menu: 3
Program ended. Bye!
```

그림 2. 초기 메뉴에서 범위를 벗어난 입력과 종료 예시

1. New Game(1을 선택 시)

사용자가 1을 선택하면, 사용자(Player)와 컴퓨터(Computer)가 대결하는 새 게임이 시작됩니다. 게임이 시작되면, 아래와 같이 각 플레이어의 점수판을 출력 후, Player부터 주사위를 던져 게임을 시작합니다. (새로운 게임 시작 시, 항상 Player부터 시작하도록 구현합니다.)

```
Select a menu: 1
```

```
Starting a game...
```

Player	Computer
1:	1:
2:	2:
3:	3:
4:	4:
5:	5:
6:	6:
Sub total: 0/63	Sub total: 0/63
+35 bonus:	+35 bonus:

C:	C:
4K:	4K:
FH:	FH:
SS:	SS:
LS:	LS:
Yacht:	Yacht:
Total: 0	Total: 0

[Player's Turn (1/12)]
Roll: [2, 3, 5, 3, 3]
Which dice to reroll (1~5)?

그림 3. 초기 메뉴에서 1. New Game 선택 화면

A. Player 순서

(주사위 던지기)

각 플레이어는 5개의 주사위로, 자신의 순서에 최대 3회까지 주사위를 던질 수 있습니다. 원하는 주사위 조합을 맞추기 위해, 주사위를 던지는 각 회마다 다시 던질 주사위를 선택하여 최대 2회까지 다시 던질 수 있습니다. (각 주사위는 random 모듈의 함수를 사용하여 랜덤하게 선택되도록 합니다.)

그림 3에서 Player는 자신의 순서에 5개의 주사위를 던져 그 결과를 출력한 후, 다시 던질 주사위 선택을 위하여 입력을 기다리고 있습니다. 이때 Player는 다시 던질 주사위의 위치를 그림 4의 예시처럼 스페이스를 이용하여 구분하여 입력하도록 합니다.

```
[Player's Turn (1/12)]
Roll: [2, 3, 5, 3, 3]
Which dice to reroll (1~5)? 1 0 4 # 0 입력으로 인한 에러 메세지 출력
Wrong input!
Which dice to reroll (1~5)? 1 3
Roll: [3, 3, 2, 3, 3]
Which dice to reroll (1~5)? 3 6 # 6 위치 무시 후 3 번 주사위만 reroll
Roll: [3, 3, 4, 3, 3]

Sorted Roll: [3, 3, 3, 3, 4]
Choose a category:
```

그림 4. Player의 주사위 던지기

그림 4에서 Player는 [2, 3, 5, 3, 3]의 주사위에서 첫 번째와 세 번째 주사위를 선택하여 다시 던졌고, 그 결과로 첫 번째 주사위는 3, 세 번째 주사위는 2로 바뀐 것을 확인할 수 있습니다. 이때 다시 한번 세 번째 주사위를 선택하여 다시 던졌고, 세 번째 주사위가 4로 바뀌었습니다.

예외 처리)

- 입력에 숫자 혹은 스페이스가 아닌 다른 입력 값이 들어오면 “Wrong input!” 문구 출력과 함께 다시 주사위 위치를 입력 받습니다. (Hint: try, except 활용)
- 다시 던질 주사위 위치에 대한 범위(1~5)를 벗어난 입력에 대해서는 해당 주사위 위치를 무시합니다.

- 다시 던질 주사위가 없다면 엔터를 입력하도록 합니다.
- 중복 주사위가 입력 (ex) 3 3 1)으로 들어올 경우 정상적으로 처리될 수 있도록 합니다.
(ex) 3 1로 처리)

(점수판 선택)

3회의 주사위 던지기가 끝나면, Player는 자신의 점수판에서 원하는 항목을 선택하여 점수를 기록해야 합니다. 이를 위해 최종 주사위 결과를 정렬하여 출력 후, 점수판에서 채워지지 않은 항목 중 하나를 선택하여 입력 받습니다. Player가 입력할 수 있는 점수판의 항목 이름은 "1, 2, 3, 4, 5, 6, C, 4K, FH, SS, LS, Y" 중 하나입니다. (2쪽 ' [주사위 점수 기준] '의 항목을 참고) 아래 예시에서 Player는 최종 주사위 결과 [3, 3, 3, 3, 4]을 가지고 점수판의 항목 3을 선택하여 점수판 3의 항목에 12점의 점수가 기록되었습니다.

[Sorted Roll] [3, 3, 3, 3, 4]

Choose a category: 3

Player	Computer
1:	1:
2:	2:
3: 12	3:
4:	4:
5:	5:
6:	6:
Sub total: 12/63	Sub total: 0/63
+35 bonus:	+35 bonus:
C:	C:
4K:	4K:
FH:	FH:
SS:	SS:
LS:	LS:
Yacht:	Yacht:
Total: 12	Total: 0

[Computer's Turn (1/12)]

Roll: [3, 6, 3, 1, 3]

그림 5. Player의 점수판 선택

예외처리)

- 이미 채워진 항목 또는 없는 항목을 입력한 경우, Wrong Input! 메시지 출력 후 다시 입력 받습니다.
- 항목 선택 시, 소문자로 입력을 해도 동일한 항목이 선택되도록 합니다.

+35 bonus는 1 (Aces) ~ 6 (Sixes)까지의 점수 합이 63이상인 경우 자동적으로 채워지도록 만듭니다. 1 ~ 6까지의 항목이 모두 채워진 경우 63을 넘지 않으면 0을 채우도록 합니다.

예시) | +35 bonus: +35 |

B. Computer 순서

Player의 점수판 기록이 끝나면, 컴퓨터의 순서가 됩니다. 이때 컴퓨터는 일정한 룰에 따라 행동하도록 만듭니다. 만들 수 있는 룰은 다양하지만, 본 과제에서는 최대한 간단하게 다음 룰만 적용하도록 합니다.

1. 점수판에서 채워지지 않은 칸의 5개의 주사위로 만들 수 있는 점수를 모두 계산합니다. C칸은 총 점수가 20점 이상인 경우만 고려합니다.
2. 그 중, 가장 높은 점수를 만들 수 있는 칸과 다시 던질 주사위를 선택합니다.
(가장 높은 점수가 2개 이상이면 가장 높은 점수 칸 중 어떤 칸을 선택해도 무방함)
 - 1~6: 칸에 해당하는 주사위를 제외하고 나머지 다시 던지기.
(ex) [6 6 4 4 6]에서 가장 높은 점수로 “6”이 선택된 경우 [6 6 4 4 6]에서 유지할 조합으로 주사위 1, 2, 5번째 선택, 다시 던질 주사위로 3, 4번째 주사위 선택.
 - C: 주사위 눈이 3 이하인 주사위는 다시 던지기.
 - 4K, FH, SS, LS, Y: 해당 칸을 완성한 조합이면 다시 던질 주사위 없음, 나머지는 모든 주사위 다시 던지기.
(ex) [1 2 3 4 6]으로 SS 선택된 경우, 다시 던질 주사위 없음.
3. 선택된 다시 던질 주사위를 다시 던져 주사위 조합을 업데이트 합니다. 이 과정을 2회 반복합니다.

아래 예시에서 컴퓨터는 자신의 순서에, 주사위를 3회 던지고 최종 조합 [5, 5, 5, 5, 5]으로 Y(Yacht)를 선택하여 컴퓨터 점수판의 Y 항목에 50점이 기록되었습니다.

```
[Computer's Turn (1/12)]
Roll: [5, 3, 5, 1, 5]
Which dice to reroll (1~5)? 2 4    # '5' 선택, 다시 던질 주사위 2, 4 번째 선택
Roll: [5, 2, 5, 5, 5]
Which dice to reroll (1~5)? 2      # 'C' 선택, 다시 던질 주사위 2 선택
Roll: [5, 5, 5, 5, 5]
```

Sorted Roll: [5, 5, 5, 5, 5]

Choose a category: Y

Player	Computer
1:	1:
2:	2:
3: 12	3:
4:	4:
5:	5:
6:	6:
Sub total: 12/63	Sub total: 0/63
+35 bonus:	+35 bonus:
C:	C:
4K:	4K:

FH:		FH:	
SS:		SS:	
LS:		LS:	
Yacht:		Yacht:	50
Total:	12	Total:	50

[Player's Turn (2/12)]
Roll: [1, 1, 4, 2, 1]

그림 6. 컴퓨터의 순서 (주사위 던지기, 점수판 기록)

C. 게임 종료

모든 플레이어가 모든 점수판(총 12칸)을 채우면 최종 점수판이 출력되고, 최종 점수를 비교하여 승패를 결정합니다. Player가 이긴 경우 "You win!", 진 경우 "You lose!", 비긴 경우 "Draw"로 간단하게 아래의 예시와 같이 출력합니다.

<Final Score Board>			
Player		Computer	
1:	2	1:	2
2:	6	2:	8
3:	12	3:	9
4:	8	4:	12
5:	15	5:	10
6:	12	6:	6
Sub total: 55/63 +35 bonus: 0		Sub total: 47/63 +35 bonus: 0	
C:	16	C:	21
4K:	14	4K:	6
FH:	18	FH:	14
SS:	15	SS:	15
LS:	30	LS:	0
Yacht:	50	Yacht:	50
Total:	198	Total:	153

You win!

Press Enter to continue...

그림 7. 게임 종료 화면

그림 7의 화면에서 엔터키를 입력하면 아래 실행 예시처럼 초기 메뉴가 출력됩니다. 이때 엔터키 외의 입력은 고려하지 않습니다.

Press Enter to continue...

[Yacht Dice]

1. New Game 2. Load Game 3. Exit

Select a menu:

그림 8. 게임 종료 후 초기 메뉴로 이동 화면

D. 게임 중단 및 점수판 저장

게임 중간에 사용자 입력이 있을 때는 언제든지, 'Q'를 입력하여 현재 진행중인 게임을 중단할 수 있습니다. 아래 예시와 같이 게임 중간에 'Q'를 입력하면 현재까지 기록된 점수판을 저장할 파일이름을 입력 받아, 입력 받은 파일명으로 점수판을 저장합니다. 저장 후, 초기 메뉴로 돌아갑니다.

Player	Computer
1:	1:
2: 2	2:
3: 9	3: 9
4:	4: 8
5:	5:
6:	6:
Sub total: 11/63 +35 bonus:	Sub total: 17/63 +35 bonus:
C:	C: 24
4K:	4K:
FH: 18	FH:
SS:	SS:
LS:	LS:
Yacht:	Yacht:
Total: 29	Total: 41

[Player's Turn (4/12)]

Roll: [1, 3, 2, 5, 3]

Which dice to reroll (1~5)? 4 5

Roll: [1, 3, 2, 2, 3]

Which dice to reroll (1~5)? Q

Game paused. Enter the filename to save:

game1.txt

File saved.

[Yacht Dice]

1. New Game 2. Load Game 3. Exit

Select a menu:

game1.txt

1: x x

2: 2 x

3: 9 9

4: x 8

5: x x

6: x x

C: x 24

4K: x x

FH: 18 x

SS: x x

LS: x x

Y: x x

그림 9. 게임 중단 및 점수판 저장

점수판을 파일에 저장 시,

- 총 12줄이며, 각 줄은 "[항목이름:] [Player 점수] [Computer 점수]" 형식으로 저장합니다.
- 아직 채워지지 않은 점수는 'x'로 표기합니다.

2. Load Game(2를 선택 시)

초기 메뉴에서 사용자가 2를 선택하면 점수판이 저장된 파일을 읽어와 점수판을 채운 후, 게임이 진행됩니다. 중단된 게임(9쪽)을 저장한 파일을 불러와 게임을 계속 진행할 수 있는 기능입니다.

```
[Yacht Dice]
-----
1. New Game  2. Load Game  3. Exit
-----
Select a menu: 2
Enter filename to load: test1.txt
```

그림 10. 초기 메뉴에서 2. Load Game 선택 예시

불러오게 될 txt 파일의 예시는 다음과 같습니다.

test1.txt

```
1: x x
2: x x
3: 6 x
4: x 8
5: x 10
6: 18 24
C: x 21
4K: x 26
FH: x x
SS: 15 x
LS: 30 x
Y: 50 x
```

Starting a game...

Player		Computer	
1:		1:	
2:		2:	
3:	6	3:	
4:		4:	8
5:		5:	10
6:	18	6:	24
Sub total: 24/63		Sub total: 42/63	
+35 bonus:		+35 bonus:	
C:		C:	21
4K:		4K:	26
FH:		FH:	
SS:	15	SS:	
LS:	30	LS:	
Yacht:	50	Yacht:	
Total:	119	Total:	89

[Player's Turn (6/12)]

Roll: [1, 4, 3, 5, 6]

Which dice to reroll (1~5)?

그림 11. txt 파일 불러오기 예시 - 유효한 파일

이때 파일은 텍스트(.txt) 파일이며, 파일의 구성은 다음과 같습니다.

- 총 12줄이며, 각 줄에는 player와 computer의 점수가 적혀 있습니다.
- 아직 채워지지 않은 점수는 'x'로 표기합니다.
- 각 줄의 점수가 숫자와 'x' 외의 다른 문자가 들어가는 것은 고려하지 않습니다.
- 두 플레이어의 남은 턴 수가 다른 경우는 없다고 가정합니다.

만일, 사용자로부터 파일명을 입력 받았을 때,

- 존재하지 않은 파일명이라면 "File does not exist." 라고 출력합니다.
- 유효하지 않은 내용이라면 "Invalid file content."문구 출력 후, 다시 파일을 불러올 것인지를 입력 받습니다.

해당 과제에서 고려하고자 하는 유효하지 않은 내용은 두가지입니다.

1. 총 줄의 개수가 12가 아닌 경우
2. 주사위 조합으로 불가능한 점수가 적혀 있는 경우

test_w1.txt	test_w2.txt	
1: x x	1: x x	Select a menu: <u>2</u>
2: x x	2: x x	Enter filename to load: <u>test11.txt</u>
3: 6 x	3: 6 x	File does not exist.
4: x 8	4: x 8	
5: x 10	5: x 11	Enter filename to load: <u>test_w1.txt</u>
6: 18 24	6: 18 24	Invalid file content.
C: x 21	C: x 21	
4K: 17 x	4K: 17 x	Enter filename to load: <u>test_w2.txt</u>
FH: 18 x	FH: 18 x	Invalid file content.
SS: 15 x	SS: 15 x	
LS: 30 x	LS: 30 x	Enter filename to load:
Y: 50 x	Y: 50 x	
W: 12 12		

그림 12. Txt 파일 불러오기 예시 - 유효하지 않은 파일

위 예시에서 test11.txt는 존재하지 않는 txt file이며, test_w1.txt는 1번 오류, test_w2.txt는 2번 오류에 해당합니다.

[사용자 정의 함수]

다음 함수들을 반드시 작성하여 프로그램을 구현해야 합니다. 아래 명시된 함수 이름은 변경하지 말아주세요. 각 함수의 매개변수의 개수 및 리턴 값 등은 자유롭게 변경 가능합니다. 그러나 동일한 기능을 하는 함수는 반드시 있어야 하며, 변경 시 무엇을 어떻게 변경했는지 보고서에 기록 하도록 합니다. 이 외에 필요한 함수는 자유롭게 정의해서 사용할 수 있습니다.

- **print_score_board(score_list)**

- 현재까지 얻은 점수 리스트(score_list)를 매개변수로 전달받아, 아래 예시와 같이 점수 판의 현재 상태를 출력하는 함수입니다.
- 게임 시작화면과 더불어 게임 진행 과정 중에 호출하여 사용하도록 합니다.
- 달라지는 점수 리스트에 따라 점수 표 형태가 어그러지지 않도록 주의합니다.

Player	Computer
1:	1:
2:	2:
3: 6	3:
4:	4: 8
5:	5: 10
6: 18	6: 24
Sub total: 24/63	Sub total: 42/63
+35 bonus:	+35 bonus:
C:	C: 21
4K:	4K: 26
FH:	FH:
SS: 15	SS:
LS: 30	LS:
Yacht: 50	Yacht:
Total: 119	Total: 89

그림 13. 점수판 출력 예시

- **load_file2list(filename)**

- 파일명을 매개변수로 전달받아 해당 파일의 스코어를 리스트로 생성하여 반환하는 함수입니다.

- **check_error(score_list)**

- 텍스트로부터 생성된 점수 리스트(score_list)가 유효한지 알려주는 함수입니다. 11쪽에 명시되어 있는 유효하지 않은 파일 조건을 만족하면 True, 유효한 파일이라면 False를 반환하도록 합니다.

- **roll_dice(dice_set=[], reroll_indices=[])**

- 현재 주사위 조합(dice_set)과 다시 던질 주사위의 인덱스 리스트(reroll_indices)를 매개변수로 전달받아 업데이트 된 주사위 목록을 반환하도록 합니다.
- 각 플레이어가 자신의 순서에 주사위를 3회 던질 때, 해당 함수를 적절하게 호출하여 사

용하도록 합니다.

- **calc_score(dice_set, sel)**

- 주사위 조합(dice_set)과 알고자 하는 점수칸(sel)을 매개변수로 전달받아 알맞은 점수를 반환하는 함수입니다. (Hint, 주사위 조합(dice_set)을 sorting한 뒤 점수를 계산하면 수월합니다.)

- **computer_pattern(dice_set, score_list)**

- 현재 주사위 조합과 현재 점수 리스트(score_list)를 매개변수로 전달받아 7쪽에 명시되어 있는 컴퓨터 룰에 따라 선택된 점수 칸(sel), 다시 던질 주사위의 인덱스 리스트를 생성해 반환하는 함수입니다. 현재 점수 리스트는 아직 채워지지 않은 칸을 판별하기 위해 사용되며, 현재 주사위 조합은 해당 주사위 조합으로 얻을 수 있는 점수들을 알기 위해 사용됩니다.