CSED101. Programming & Problem solving
Spring 2023

Programming Assignment #1
(60 points)

김장원 (jangwonkim@postech.ac.kr)

■ **제출 마감일: 2023.04.06 23:59**
■ **파이썬 버전: Python 3.x**

■ **제출물**

- .py 소스 코드 (assn1.py)
  - 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 PLMS를 통해 제출해 주세요.

■ **주의 사항**

- 구문 오류(Syntax Error)가 발생하거나 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).
  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 에러 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 (PLMS의 본 과목 공지사항에 등록된 글 중, 제목이 [document about cheating]인 글에 첨부되어 있는 disciplinary.pdf를 참조하세요).

■ Problem: 행맨 게임 (Hangman Game)

[문제]

행맨 게임은 상대방이 생각하는 단어를 주어진 기회안에 맞히는 게임입니다. 컴퓨터가 선택한 단어를 맞히는 행맨 게임을 구현해봅니다.

[목적]

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

[주의사항]

(1) 이번 과제는 함수를 정의하고 사용하는 방법을 익히는 문제이므로 함수를 정의하지 않고 기능을 구현한 경우 감점 처리됩니다. 문서에 반드시 정의해서 사용해야 할 사용자 정의 함수가 설명되어 있으니 확인 후 구현하도록 합니다.

(2) int, float, bool, str, list, tuple 자료형까지 수업시간에 다룬 내용에 한해서 사용 가능합니다.

[설명]

컴퓨터가 미리 정의된 리스트에 있는 단어 중 하나를 랜덤하게 선택합니다. 아울러 행맨의 생명(life)은 10부터 시작합니다. 단어를 맞히기 위해 사용자가 영문자를 하나 입력하는데, 선택된 단어에 입력한 영문자가 없는 경우 life의 수가 1 감소합니다. 10번의 없는 문자를 입력하여 life가 0이 되기 전에 단어를 맞혀야 하며 life가 0이 되면, 선택된 단어를 공개(출력)하고 게임은 종료됩니다.

프로그램을 실행하면 Hangman game starts! 문구와 함께 게임이 시작됩니다. 이 때 컴퓨터는 미리 단어가 저장된 리스트(word_list)에서 랜덤하게 하나의 단어를 선택하게 됩니다.

1. 단어 선택

컴퓨터는 미리 준비된 단어 리스트(list)에서 하나의 단어를 뽑게 됩니다. 이 단어 리스트는 아래와 같이 미리 선언합니다. (단어는 영문 소문자로만 구성되어 있다고 가정합니다.)

```
word_list = ['apple', 'april', 'banana', 'blue', 'coral', 'dictionary', 'flower', 'peach', 'strawberry', 'watermelon']
```

2. 프로그램 초기 화면

아래 예시는 프로그램이 실행되어, banana라는 단어가 랜덤하게 선택된 경우입니다.

```
Hangman game starts!
-----------------------------------------------
Word: _ _ _ _ _ _
Used:
Life: 10
-----------------------------------------------
Choose a character:
```

게임을 위한 화면 구성은 다음과 같습니다.

- Word란: 선택된 단어의 길이만큼 '_'를 출력
- Used란: 유저가 단어를 맞히기 위해 입력한 모든 문자를 입력한 순서대로 출력
- Life란: 남은 life를 출력

이후 Choose a character: 라는 문구와 함께 유저가 문자를 선택하기를 기다리게 됩니다.

3. 게임 진행

유저가 문자를 선택하게 되면, 컴퓨터가 선택한 단어에 입력한 문자가 있는지 없는지를 비교하여 게임을 진행하게 됩니다. 이 때 유저 입력은 영문 소문자 외에는 고려하지 않으며, 입력 시 소문자 1 개씩만 입력한다고 가정합니다.
아래 예시는 유저가 선택한 문자가 컴퓨터가 선택한 단어에 있는 경우로 해당되는 부분의 문자를 출력합니다. 또한 Used란에 입력한 문자가 순서대로 출력 됩니다. 그 후 프로그램은 다시 Choose a character 라는 말과 함께 다음 문자를 선택하기를 기다립니다. (예시의 빨간색 밑줄은 유저 입력에 해당합니다.)

```
Hangman game starts!
-----------------------------------------
Word: _ _ _ _ _ _
Used:
Life: 10
-----------------------------------------
Choose a character: a

-----------------------------------------
Word: _ a _ a _ a
Used: a
Life: 10
-----------------------------------------
Choose a character:
```

다음 예시는 유저가 선택한 문자가 컴퓨터가 선택한 단어에 존재하지 않는 경우입니다. 예시에서 확인할 수 있듯이 없는 문자를 입력하면 life 의 수가 감소하게 됩니다. 아울러 used 란에 입력한 문자가 추가된 것을 확인할 수 있습니다.

```
Choose a character: c

-----------------------------------------
Word: _ a _ a _ a
Used: a c
Life: 9
-----------------------------------------
Choose a character:
```

만약 유저가 문자를 선택 시, 이전에 선택했던 문자를 다시 선택하면 아래 예시와 같은 에러와 함께 다시 문자를 선택하게 됩니다. (이 때 life 는 감소하지 않습니다.)

```
-----------------------------------------
Word: _ a _ a _ a
Used: a c
Life: 9
-----------------------------------------
Choose a character: a
You have already checked this character. Try another one.

-----------------------------------------
Word: _ a _ a _ a
Used: a c
Life: 9
-----------------------------------------
Choose a character:
```

4. 게임 종료

유저가 선택된 단어를 life 가 0 이 되기 전에 맞히거나, life 가 0 이 되면 게임이 종료됩니다. 게임이 종료되면, Do you want to play another game? 이라는 문구가 출력되며 유저가 no 를 입력하면 프로그램이 종료됩니다. 유저가 yes 를 입력한 경우, 컴퓨터가 단어를 랜덤하게 선택한 후 게임이 다시 시작됩니다. yes, no 외 입력은 Wrong input! 이라는 문구와 함께 다시 입력 받습니다.

(1) 게임 성공 화면

아래는 life 가 0 이 되기전에 답을 맞힌 경우로 Hangman Survived! 라는 문구가 출력됩니다.

```
-------------------------------------------
Word: _ a _ a _ a
Used: a c
Life: 9
-------------------------------------------
Choose a character: b

-------------------------------------------
Word: b a _ a _ a
Used: a c b
Life: 9
-------------------------------------------
Choose a character: n

-------------------------------------------
Word: b a n a n a
Used: a c b n
Life: 9
-------------------------------------------

Hangman Survived!
Do you want to play another game?:
```

(2) 게임 실패 화면

다음 예시는 주어진 10 개의 life 안에 답을 맞히지 못한 경우로 Hangman Die! 문구와 함께 정답 단어가 출력됩니다.

```
-------------------------------------------
Word: b a _ a _ a
Used: a c b z x v s q y w e
Life: 1
-------------------------------------------
Choose a character: u

Hangman Die!
The answer was banana.
Do you want to play another game?:
```

(3) 기타

하나의 게임이 끝난 후, Do you want to play another game? 문구에 대한 유저입력에 따른 예시는 다음과 같습니다.

가. yes 입력 시, 다시 게임을 시작하는 예시

```
The answer was banana.
Do you want to play another game?: yes


-----------------------------------------
Word: _ _ _ _
Used:
Life: 10
-----------------------------------------
Choose a character:
```

나. no 입력 시, 프로그램을 종료한 예시

```
Hangman Survived!
Do you want to play another game?: no


Quit the Hangman Game.
```

[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 가능 합니다. 변경 시 무엇을 어떻게 변경했는지 보고서에 기록하도록 합니다. 이 외에 필요한 함수는 정의해서 사용할 수 있습니다.

- print_status(comp_word, used, life)
  - 컴퓨터가 선택한 단어(comp_word), 유저가 사용한 문자 목록(used), 남은 생명(life)을 매개변수로 전달 받아, 아래와 같이 현재 어떤 문자가 드러났는지(Word 란). 어떤 문자를 사용했는지(Used 란), life 가 얼마나 남아있는지(Life 란)를 출력하는 함수입니다.
  - 게임 시작화면과 더불어 게임 진행 과정 중에 계속 호출하여 사용하도록 합니다.

```
-----------------------------------------
Word: _ a _ a _ a
Used: a c
Life: 9
-----------------------------------------
```

- reveal_word(comp_word, used)
  - 컴퓨터가 선택한 단어(comp_word), 유저가 사용한 문자 목록(used)을 매개변수로 전달 받아, 아래와 같이 현재 유저가 맞혀야 할 단어에 드러난 문자를 출력하는 함수입니다.
  - print_status 함수 호출 시 같이 사용하도록 합니다.

```
_ a _ a _ a
```

- is_word_guessed(comp_word, used)
  - 컴퓨터가 선택한 단어(comp_word), 유저가 사용한 문자 목록(used)을 매개변수로 전달 받아, 유저가 단어를 맞혔는지 알려주는 함수입니다. 유저가 답을 맞힌 경우 True, 그렇지 않은 경우 False 를 반환합니다.