

웹파이선프로그래밍 2020년 1학기 중간점검과제

[문제1] N을 입력 받으면, $N \times (N-1) \times \dots \times 1$ 을 계산하는 함수 CalcFactorial()를 작성합니다.

- 예를 들어, CalcFactorial(3)을 실행하면 이 함수는 $3 \times 2 \times 1$ 을 계산하여 6을 return 함
- 함수의 이름은 반드시 CalcFactorial()로 함
- CalcFactorial() 함수는 입력 parameter를 하나만 받음
- CalcFactorial() 함수는 입력 parameter가 정수가 아니면 -1을 return 함
- CalcFactorial() 함수는 입력 parameter가 0보다 작거나 같으면 -2를 return 함

[문제2] 가위/바위/보 게임을 만들어 봅시다.

- 답안 화일의 “Start of Answer”와 “End of Answer” 사이에 답안을 작성함
- 답안 내의 caseTypes 리스트처럼, “가위/바위/보”는 각각 "scissor", "rock", "paper"로 사용함
- GenerateRandomCaseForComputer() 함수는 Python의 built-in 난수 생성 함수를 사용하여, "scissor", "rock", "paper" 중 한 단어를 난수로 생성하여 return 함
- GenerateRandomCaseForComputer() 함수는 입력 parameter가 없음
- MakeDecision() 함수는 사용자의 선택과 컴퓨터가 난수로 생성한 선택을 입력 parameter로 받은 후, 사용자와 컴퓨터 중 누가 이겼는 지를 알려줌
- MakeDecision() 함수는 첫째 입력 parameter로 사용자의 선택에 해당하는 문자열을 전달 받고, 둘째 입력 parameter로 컴퓨터가 생성한 난수 기반 선택에 해당하는 문자열을 전달 받음
- MakeDecision() 함수는 입력 parameter로 전달 받은 사용자의 선택과 컴퓨터의 선택이 "scissor", "rock", "paper"이 아닌 경우에 -1을 return함
- MakeDecision() 함수는 사용자와 컴퓨터의 선택을 토대로 승부를 결정하고, 사용자가 이긴 경우는 문자열 “User”를 return하고, 컴퓨터가 이긴 경우는 문자열 “Computer”를 return함. 무승부의 경우는 문자열 “Tie”를 return함

[문제3] 1 과 자신만을 약수(나누어 떨어지는 수)로 가지는 수를 소수(prime number)라 한다. 예를 들어, 5는 소수이다. 8은 소수가 아니며 [1,2,4,8]을 약수로 가진다. 소수인지 아닌지 판별하는 isPrime() 함수를 구현 하시오.

- 이 함수의 입력 파라미터는 num(정수)과 L(리스트)이다. 입력 파라미터 num은 1을 초과하는 정수라고 가정 하고, L은 empty list라고 가정한다. 이를 확인하는 절차는 불필요하다.
- 이 함수는 num이 소수이면 True, 아니면 False를 반환한다 (Boolean type을 반환).
- 만약, num이 소수가 아니면, num의 약수의 리스트를 L에 지정하고 출력(print)한다. 이 리스트는 1과 num을 포함한다. num이 소수일 때 L의 값은 어떤 값 이어도 상관없으며 출력하지 않는다.

[문제4] 다음 세 함수를 구현하여 영어단어를 맞히는 퀴즈 프로그램을 구현 하시오. 이 프로그램은 정답인 단어가 몇 글자인지 사용자에게 보여주고 글자(알파벳)는 모두 숨긴다. 사용자는 이 단어에 어떤 알파벳이 포함되었는지 여러 기회에 걸쳐서 맞춰 나간다. 기회의 횟수는 단어의 길이와 동일하도록 제한한다. 해당 알파벳이 단어에 한번 이상 포함되면 그 알파벳의 위치를 모두 보여주어 조금씩 단어가 드러난다. 주어진 단어와 입력 알파벳 모두 대문자로 가정한다.

4-1) 게임의 최종 결과를 반환하는 getResult() 함수를 구현

- 입력 parameter는 status이다. status는 맞춰 나가는 단어의 현재 상태이며 str type이다. 아직 못 맞힌 글자는 언더바('_')로 표시되고, 맞힌 글자는 알파벳으로 표시된다 (예: 'HIPP_P_TA_US').
- 이 함수는 string type을 반환한다.
- 단어의 현재 상태(status)에 못 맞힌 글자가 하나라도 있으면 함수는 “You lose”를 반환한다.
- status에 못 맞힌 글자가 하나도 없이 알파벳이 모두 드러난 상태라면 함수는 “You win”을 반환한다.

4-2) status를 업데이트 하는 updateStatus() 함수를 구현

- 입력 parameter는 word (정답 단어), x (사용자가 입력한 하나의 알파벳), status (맞혀 나가는 단어의 현재 상태)이고 입력 parameter는 모두 str type 이다. 이 함수는 status를 업데이트한 new_status (str type)를 반환한다.
- 이 함수는 word 안에 x가 존재하는지 확인하여, 존재하지 않으면 기존 status값을 그대로 new_status에 지정하여 반환한다.
- word 안에 x가 존재하면 기존의 status에서 해당 알파벳 위치의 언더바('_')를 x의 알파벳으로 바꾼 new_status를 반환한다. 이 때, word 안에 x가 2개 이상 존재하면 이를 모두 찾아서 바꿔주어야 한다. 예를 들어, word = 'HIPPOPOTAMUS', x='P', status = 'HI____TA_US'이면, updateStatus() 함수의 반환값은 'HIP_P_TA_US'이다.
- Hint: string method 중 str.count()와 str.find()를 활용하시오. 필수는 아님.

4-3) getResult()와 updateStatus()를 반드시 활용하여 startGame() 함수를 구현

- 입력 parameter는 사용자가 맞혀야 하는 단어 word이다.
- 사용자가 주어진 기회 안에 모든 알파벳을 맞히면 “You win”을 반환하고, 주어진 기회가 모두 끝났는데 포함된 알파벳을 모두 알아내지 못하면 “You lose”를 반환한다.
- 지역변수 count (int type)는 사용자에게 주어지는 기회이며 word의 길이와 같은 값으로 초기화한다. 예를 들어, word='HIPPOPOTAMUS'라면, 기회는 12번이다.
- 지역변수 status (str type)는 맞혀 나가는 단어의 현재 상태이며 word의 같은 길이의 언더바로 초기화한다. 예를 들어, word='HIPPOPOTAMUS'라면, status의 초기값은 '_____'이다.
- 사용자가 입력하기 직전에 매번 남은 기회의 횟수(count)와 단어의 현재 상태(status)를 출력(print)해주고, 함수에서 최종 결과를 반환하기 직전에 마지막으로 한 번 더 단어의 현재 상태(status)를 출력한다.

```
-----
12 chances. Enter a letter: A
_____A____
11 chances. Enter a letter: E
_____A____
10 chances. Enter a letter: I
_I_____A____
9 chances. Enter a letter: O
_I__O_O_A____
8 chances. Enter a letter: U
_I__O_O_A_U__
7 chances. Enter a letter: S
_I__O_O_A_US_
6 chances. Enter a letter: H
HI__O_O_A_US_
5 chances. Enter a letter: P
HIPPOPO_A_US_
4 chances. Enter a letter: T
HIPPOPOTA_US_
3 chances. Enter a letter: M
HIPPOPOTAMUS_
You win
```

(a) 실행 예시 1

```
-----
12 chances. Enter a letter: A
_____A____
11 chances. Enter a letter: E
_____A____
10 chances. Enter a letter: I
_I_____A____
9 chances. Enter a letter: O
_I__O_O_A____
8 chances. Enter a letter: U
_I__O_O_A_U__
7 chances. Enter a letter: P
_IPPOPO_A_U__
6 chances. Enter a letter: B
_IPPOPO_A_U__
5 chances. Enter a letter: V
_IPPOPO_A_U__
4 chances. Enter a letter: D
_IPPOPO_A_U__
3 chances. Enter a letter: L
_IPPOPO_A_U__
2 chances. Enter a letter: M
_IPPOPO_AMU__
1 chances. Enter a letter: S
_IPPOPO_AMUS_
You lose
```

(b) 실행 예시 2

[그림] 문제 4의 프로그램 실행 예시