

26일 (토)

1. 프로그래머스 문제 풀이

1) 다트 게임 (못 푼 문제)

[1차] 다트 게임

카카오 게임별의 하반기 신규 서비스로 다트 게임을 출시하기로 했다. 다트 게임은 다트판에 다트를 세 차례 던져 그 점수의 합계로 실력을 겨루는 게임으로, 모두가 간단히 즐길 수 있다.

갓 입사한 무지는 코딩 실력을 인정받아 게임의 핵심 부분인 점수 계산 로직을 맡게 되었다. 다트 게임의 점수 계산 로직은 아래와 같다.

1. 다트 게임은 총 3번의 기회로 구성된다.
2. 각 기회마다 얻을 수 있는 점수는 0점에서 10점까지이다.
3. 점수와 함께 Single(S), Double(D), Triple(T) 영역이 존재하고 각 영역 당첨 시 점수에서 1제곱, 2제곱, 3제곱 (점수¹, 점수², 점수³)으로 계산된다.
4. 옵션으로 스타상(*), 아차상(#)이 존재하며 스타상(*) 당첨 시 해당 점수와 바로 전에 얻은 점수를 각 2배로 만든다. 아차상(#) 당첨 시 해당 점수는 마이너스된다.
5. 스타상(*)은 첫 번째 기회에서도 나올 수 있다. 이 경우 첫 번째 스타상(*)의 점수만 2배가 된다. (예제 4번 참고)
6. 스타상(*)의 효과는 다른 스타상(*)의 효과와 중첩될 수 있다. 이 경우 중첩된 스타상(*) 점수는 4배가 된다. (예제 4번 참고)
7. 스타상(*)의 효과는 아차상(#)의 효과와 중첩될 수 있다. 이 경우 중첩된 아차상(#)의 점수는 -2배가 된다. (예제 5번 참고)
8. Single(S), Double(D), Triple(T)은 점수마다 하나씩 존재한다.
9. 스타상(*), 아차상(#)은 점수마다 둘 중 하나만 존재할 수 있으며, 존재하지 않을 수도 있다.

0~10의 정수와 문자 S, D, T, *, #로 구성된 문자열이 입력될 시 총점수를 반환하는 함수를 작성하라.

입력 형식

"점수|보너스|[옵션]"으로 이루어진 문자열 3세트.

예) 1S2D*3T

- 점수는 0에서 10 사이의 정수이다.
- 보너스는 S, D, T 중 하나이다.
- 옵션은 *이나 # 중 하나이며, 없을 수도 있다.

입력 형식

"점수|보너스|[옵션]"으로 이루어진 문자열 3세트.

예) 1S2D*3T

- 점수는 0에서 10 사이의 정수이다.
- 보너스는 S, D, T 중 하나이다.
- 옵션은 *이나 # 중 하나이며, 없을 수도 있다.

출력 형식

3번의 기회에서 얻은 점수 합계에 해당하는 정수값을 출력한다.

예) 37

입출력 예제

예제	dartResult	answer	설명
1	1S2D*3T	37	$1^1 * 2 + 2^2 * 2 + 3^3$
2	1D2S#10S	9	$1^2 + 2^1 * (-1) + 10^1$
3	1D2S0T	3	$1^2 + 2^1 + 0^3$
4	1S*2T*3S	23	$1^1 * 2 * 2 + 2^3 * 2 + 3^1$
5	1D#2S*3S	5	$1^2 * (-1) * 2 + 2^1 * 2 + 3^1$
6	1T2D3D#	-4	$1^3 + 2^2 + 3^2 * (-1)$
7	1D2S3T*	59	$1^2 + 2^1 * 2 + 3^3 * 2$

[해설 보러가기](#)

#구글 검색 풀이

```
def solution(dartResult):
    answer = []
    dartResult = dartResult.replace('10','k') # 내가 해매던 부분 10을 인식하는 법 치환!!
    point = ['10' if i == 'k' else i for i in dartResult] # k 를 다시 10으로 변경하는 과정
    print(point)

    i = -1 # answer 리스트를 0 부터 탐색하기 위한 설정
    sdt = ['S', 'D', 'T']
    for j in point:
        if j in sdt :
            answer[i] = answer[i] ** (sdt.index(j)+1) #리스트 인덱스 숫자를 이용해서 숫자 만들기.. 배워야 할점
        elif j == '*':
            answer[i] = answer[i] * 2
```

```

        if i != 0 :
            answer[i - 1] = answer[i - 1] * 2
        elif j == '#':
            answer[i] = answer[i] * (-1)
        else:
            answer.append(int(j))
            i += 1 # 숫자가 들어갈 때만 인덱스를 넘겨주고 나머지 알파벳, # * 은 인덱스 안넘기고 계산
    return sum(answer)

```

#문제풀이 1위

import re # TIL ??

```

def solution(dartResult):
    bonus = {'S' : 1, 'D' : 2, 'T' : 3}
    option = {'' : 1, '*' : 2, '#' : -1}
    p = re.compile('(\d+)([SDT])([*#]?)') # TIL re 라이브러리 compile 메소드 와 정규표현식
    dart = p.findall(dartResult) # TIL findall( ) 메서드
    for i in range(len(dart)):
        if dart[i][2] == '*' and i > 0:
            dart[i-1] *= 2
        dart[i] = int(dart[i][0]) ** bonus[dart[i][1]] * option[dart[i][2]]

    answer = sum(dart)
    return answer

```

— 배운점 —