

[소프트웨어 프로젝트 II]

[week4_assignment_수정]

20171661_이다운_class04

이시윤 교수님

1. factorial_recursive

1)while 문을 통한 실행문 추가

```

6      -N = int(input("Enter your number: "))
7      -
8      6
9      7      def factorial(N):
10     8          if N <= 1:
11     9              return 1
12    10          else:
13    11              return N*factorial(N-1)
14    12
15      -print(N,"! = ",factorial(N))
13      +run = True
14      +
15      +while run:
16      +      N = int(input("Enter your number: "))
17      +      if N >= 0:
18      +          print(N,"! = ",factorial(N))
19      +      elif N == -1:
20      +          run = False
21      +          print("end")
22      +      else:
23      +          N = int(input("Enter positive number: "))

```

Red-line 15 에서는 입력 값 N 을 조건에 맞는지 검사하지 않고 바로 출력했지만 green-line 13-23 과 같이 while 문을 통해 N 이 양수일 때 원하는 실행을 하고 N 이 -1 일 때는 종료, N 이 음수일 때는 입력 값을 다시 받도록 수정하였습니다.

2. combination

1) 공식적 계산 함수에 조건 추가

```

13 +def combination1(n, m):
14 +     if n == m or m == 0:
15     return 1
16     elif n > m:
17 -         return factorial(n)/(factorial(m)*factorial(n-m))
17 +         return factorial(n) / (factorial(m) * factorial(n - m))
18 +

```

Green-line14 에 nCr 에서 r 이 0 일 때 조합의 결과는 1 이라는 항목을 추가했습니다.

2) 형식적 부분 수정

```

10 -         return N*factorial(N-1)
10 +         return N * factorial(N - 1)

13 -def combination(n,m):
14 -     if n == m:
13 +def combination1(n, m):

17 -         return factorial(n)/(factorial(m)*factorial(n-m))
17 +         return factorial(n) / (factorial(m) * factorial(n - m))

```

Line10, 17 에서 연산자 앞뒤 간격, line13 에서 ‘,’ 뒤 간격을 수정했습니다.

3)combination recursive 함수 추가

```

20 +def combination2(n, m):
21 +     if n == m or m == 0:
22 +         return 1
23 +     else:
24 +         return combination2(n - 1, m - 1) + combination2(n - 1, m)
25 +

```

받은 입력 값 n, m 의 값이 같거나 m 이 0 일 때 1 인 결과를 갖습니다. 이 밖의 조건일 땐, 본인의 함수를 사용하여 함수를 계산합니다.

4) combination recursive 함수(combination2)추가에 따른 실행문 수정

```

41 +         elif n >= m:
42 +             result1 = combination1(n, m)
43 +             result2 = combination2(n, m)
44 +             print("공식: C(", n, ",", m, ") = ", int(result1))
45 +             print("재귀적 함수: C(", n, ",", m, ") = ", result2)

```

두 함수를 모두 사용해서 결과값을 비교해보도록 출력합니다.

5) n, m 범위

```

19 28 run = True
20 29 while run:
21 30     n = int(input("Enter n: "))
22 31     if n < 0 and n != -1:
23 -         print("양수인 정수를 입력해 주세요.")
24 +         print("양수인 정수 n을 입력해 주세요.")
25 32
26 33     continue
27 34     elif n == -1:
28 35         print("종료")
29 36         run = False
30 37
31 38     else:
32 39         m = int(input("Enter m: "))
33 40         if m < 0:
34 -             print("양수인 정수를 입력해 주세요.")
35 -             elif n > m:
36 -                 result = combination(n,m)
37 -                 print("C(",n,",",m,") = ", result)
38 +             print("양수인 정수 m을 입력해 주세요.")
39 +             elif n >= m:
40 +                 result1 = combination1(n, m)
41 +                 result2 = combination2(n, m)
42 +                 print("공식: C(", n, ",", m, ") = ", int(result1))
43 +                 print("재귀적 함수: C(", n, ",", m, ") = ", result2)
44 +             else:
45 +                 print("n보다 작은 정수 m을 입력해 주세요")

```

위와 같이 실행문에서 n, m의 범위 조건을 지정해주었기 때문에 함수에서 n, m의 범위를 지정해주지 않아도 된다고 생각합니다.