문제4. 팩토리얼 구하기

재귀 호출 프로그램이 정상적으로 작동하려면

‘종료조건’ 이 필요

재귀호출 팩토리얼

```python
41  # 팩토리얼
42  def fact(n):
43      f = 1
44      for i in range(1,n+1):
45          f = f * i
46      return f
47
48  print(fact(4))
```

Python - chap_4_factorial.py:49 ✔

```
24
[Finished in 0.079s]
```

```python
43  # 팩토리얼 재귀
44  def fact(n):
45      if n<=1:
46          return 1
47      return n*fact(n-1)
48
49  print(fact(4))
```

Python - chap_4_factorial.py:46 ✔

```
24
[Finished in 0.086s]
```

연습문제 4-1 재귀호출로 n까지 합 찾기

```python
# 1부터 n까지 합 재귀
def n_hap(n):
    if n==1:
        return 1
    return n+n_hap(n-1)

print(n_hap(10))
```

Python - chap_4_factorial.py:20 ✔

```
55
[Finished in 0.083s]
```

연습문제 4-2 재귀호출로 최댓값 찾기

```
29
30  def max(x,y):                                   #큰 값 비교
31      if x>y:
32          return x
33      else:
34          return y
35
36  def find_max(list,k):                           #변수가 되는 list와 k는 리스트의 크기
37                                                  #list = [10,23,467,45,1]일 때, k=5
38      if k==1:                                    #종료조건
39          return list[0]
40      else:
41          return max(list[k-1],find_max(list,k-1))
42
43  list=[10,23,467,45,1]
44
45  print(find_max(list,len(list)))
46
```

max(list[4],**find_max(list,4)**)

max(list[3],**find_max(list,3)**)

max(list[2],**find_max(list,2)**)

max(list[1],**find_max(list,1)**)

max(list[0],find_max(list,0))

Python - chap_4_factorial.py:41 ✔

```
467
[Finished in 0.08s]
```

C:₩Users₩jundl₩workspace₩algopy₩chap_4_factorial.py* 41:49 (1, 33)

연습문제 4-2 재귀호출로 최댓값 찾기

```
29
30  def max(x,y):                                   #큰 값 비교
31      if x>y:
32          return x
33      else:
34          return y
35
36  def find_max(list,k):                           #변수가 되는 list와 k는 리스트의 크기
37                                                  #list = [10,23,467,45,1]일 때, k=5
38      if k==1:                                    #종료조건
39          return list[0]
40      else:
41          return max(list[k-1],find_max(list,k-1))
42
43  list=[10,23,467,45,1]
44
45  print(find_max(list,len(list)))
46
```

max(1,**find_max(list,4)**)

max(45,**find_max(list,3)**)

max(467,**find_max(list,2)**)

max(23,**find_max(list,1)**)

max(10,find_max(list,0))

Python - chap_4_factorial.py:41 ✔

```
467
[Finished in 0.08s]
```

C:₩Users₩jundl₩workspace₩algopy₩chap_4_factorial.py* 41:49 (1, 33)

연습문제 4-2 재귀호출로 최댓값 찾기

```
29
30  def max(x,y):                                   #큰 값 비교
31      if x>y:
32          return x
33      else:
34          return y
35
36  def find_max(list,k):                           #변수가 되는 list와 k는 리스트의 크기
37                                                  #list = [10,23,467,45,1]일 때, k=5
38      if k==1:                                    #종료조건
39          return list[0]
40      else:
41          return max(list[k-1],find_max(list,k-1))
42
43  list=[10,23,467,45,1]
44
45  print(find_max(list,len(list)))
46
```

max(1,**find_max(list,4)**)

max(45,**find_max(list,3)**)

max(467,**find_max(list,2)**)

max(23,**find_max(list,1)**)

K가 1이므로 list[0] = 10

Python - chap_4_factorial.py:41 ✔

```
467
[Finished in 0.08s]
```

C:₩Users₩jundl₩workspace₩algopy₩chap_4_factorial.py* 41:49 (1, 33)

연습문제 4-2 재귀호출로 최댓값 찾기

```
29
30  def max(x,y):                                   #큰 값 비교
31      if x>y:
32          return x
33      else:
34          return y
35
36  def find_max(list,k):                           #변수가 되는 list와 k는 리스트의 크기
37                                                  #list = [10,23,467,45,1]일 때, k=5
38      if k==1:                                    #종료조건
39          return list[0]
40      else:
41          return max(list[k-1],find_max(list,k-1))
42
43  list=[10,23,467,45,1]
44
45  print(find_max(list,len(list)))
46
```

max(1,**467**)

max(45,**467**)

max(467,**20**)

max(23,**10**)

Python - chap_4_factorial.py:41 ✔

```
467
[Finished in 0.08s]
```

C:₩Users₩jundl₩workspace₩algopy₩chap_4_factorial.py* 41:49 (1, 33)

감 사 합 니 다