

1. 재귀함수 활용하기

아래는 0부터 입력받은 숫자까지 포함하여 홀수만을 더해 그 합을 출력해주는 코드를 반복문을 활용해서 작성한 예이다. 아래 코드를 활용하여 같은 기능을 하는 재귀함수를 작성하라.

```
def total_count(n):
    sum = 0
    while True:
        if (n%2==1):
            sum = sum + n
            n = n - 1
        else:
            n = n - 1

        if (n == 0):
            break

    print("The computed number is", sum)

count = int(input("Please input the number: "))
total_count(count)
```

2. 단순 무식한 곱셈 함수

곱셈연산자가 없다면 덧셈과 뺄셈 연산자만 가지고 다음과 같이 곱셈 함수를 구현할 수 있다. 여기서 m 과 n 값은 항상 자연수라고 가정한다. 즉, 음수 곱셈은 고려하지 않는다

```

1 def mult(m,n):
2     if n > 0:
3         return m + mult(m,n-1)
4     else:
5         return 0

```

이 재귀함수는 덧셈하는 횟수가 n 에 비례하고, 공간 사용량도 n 에 비례한다. `mult(3,6)` 호출을 실행추적하면 다음과 같다.

```

mult(3,6)
=> 3 + mult(3,5)
=> 3 + 3 + mult(3,4)
=> 3 + 3 + 3 + mult(3,3)
=> 3 + 3 + 3 + 3 + mult(3,2)
=> 3 + 3 + 3 + 3 + 3 + mult(3,1)
=> 3 + 3 + 3 + 3 + 3 + 3 + mult(3,0)
=> 3 + 3 + 3 + 3 + 3 + 3 + 0
=> 3 + 3 + 3 + 3 + 3 + 3
=> 3 + 3 + 3 + 3 + 6
=> 3 + 3 + 3 + 9
=> 3 + 3 + 12
=> 3 + 15
=> 18

```

위의 재귀함수를 공간 사용량이 일정한 꼬리재귀 형태로 함수를 다음 틀에 맞추어 재작성하자.

```

def mult(m,n):
    def loop(n,
        if      :
            return
        else:
            return
    loop(n,

```