**프로그래밍 언어론 과제 #02**

컴퓨터과학과

201710912

김지섭

1. 일반적인 recursion에 비해 tail-recursion의 장점이 무엇인지 설명하시오

일반적인 recursion의 형태는 자기 자신을 지속적으로 호출하는 형태로 실행되는데 이때 Stack에 지속적으로 같은 내용의 Data가 축적되게 되어 Overhead가 발생될 수 있습니다.

하지만 tail-recursion의 경우 반환을 하면서 함수를 실행하는 형태이기 때문에 새로운 Stack Frame을 만들지 않고 결과를 반환하여, Stack의 Overhead문제를 해결 할 수 있습니다.

2. 변수의 참조 모델(Reference model of variables)을 채택한 프 로그래밍 언어들은 가비지 콜렉션을 제공하는 경우가 많다. 왜 그런지 이유를 설명하시오.

Reference model의 경우, 모든 변수가 I-value로 사용되어, 개발자가 수동으로 전체 메모리 상태를 관리하기 어렵고, 제대로 메모리 관리를 하지 않는 경우, 메모리 누수가 발생되게 되기 때문에 가비지 콜렉션을 제공하여 binding이 제거된 메모리 공간을 해제 하기 위해서 대부분 가비지 콜렉션을 제공합니다.

3. 빈 줄(blank line)을 찾는 C언어 코드가 다음과 같다고 가정하 자. 각 함수의 이름은 대충 그 함수의 단어가 의미하는 것을 한다 고 가정한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이 코드를 while문 또는 do while문으로 다시 작성한다면 어떻게 수정할 수 있을까 코드로 작성해본다

line = read\_line();

While(line) {

if(all\_blanks(line)) break;

consum\_line(line)

line = read\_line();

}