

< 과제 4 : Yacc Programming >

1. 개요

Yacc Program 은 Lex Program 에서 특정부분을 분석하며 읽는 내용을 넘겨받아서 특정 이벤트를 발생시킬 수 있다. 이 기능을 이용하여 읽어 들인 부분이 어떤 기능을 하는지를 분석하고 각 기능들에 대한 카운트를 하여 출력하는 프로그램을 구현하도록 한다.

2. 순서

- A. Yacc Program 을 실행 시 C 코드를 읽어오게 된다.
- B. 코드의 내용을 분석한다. (test.c 파일의 내용을 읽어 들임)
- C. 각 코드에 나오는 특성에 따라서 변수에 카운트를 한다.
- D. 카운트 된 값을 다음과 같이 화면에 출력한다.

```
function = 3
operator = 15
int = 5
char = 2
pointer = 1
array = 2
selection = 2
loop = 2
return = 1
```

3. 요구사항

함수 선언, 함수 정의, 일반 변수 선언문, 포인터 변수 선언문, 배열 변수 선언문, 수식, 함수 호출, 선택문, 조건문, 반복문, 리턴문 카운팅 확인

A. Function

함수 선언, 함수 사용, printf 와 같은 내장 함수 사용 횟수를 카운팅.
sizeof 는 함수가 아닙니다.

B. Operator

++, +=, =, -> 등 연산에 관련된 operator 들을 카운팅 하는
것입니다. 단, a++; 이 있을 경우 ++이 하나의 의미이므로 1 개로
카운팅.

Unary_operator 는 수식이 아니라 부호입니다. Unary_operator 는
카운트 하지 않습니다.

? : 연산자는 카운트 하지 않습니다.

포인터 연산자 중 .는 애매한 경우가 있어 빼고 -> 만 count 하시면
됩니다.

C. Int

Int 로 선언된 변수 개수. 파라미터로 선언된 Int 개수도 count 해야
합니다. Int a,b 는 int 개수 2 증가입니다.

D. Char

Char 로 선언된 변수 개수. 파라미터로 선언된 char 개수도 count
해야 합니다.

E. Pointer

Pointer 로 선언된 변수 개수.

Int * a; 는 int 개수, pointer 개수 모두 증가.

int **a 인 경우 포인터 개수 하나만 증가하면 됩니다.

파라미터로 선언된 pointer 개수도 count 해야 합니다.

포인터 함수 int (*p)(int, int) 는 포인터 개수, 함수 개수 모두

증가합니다. 이때 int 로 선언된 변수가 없으므로 Int 값은 증가하지 않습니다.

F. Array

배열로 선언된 변수 개수.

Int a[3]은 int 개수, 배열 개수 모두 증가.

또한 int a[3] = {0}과 같은 것도 처리가 가능해야 함.

2 차원 배열도 Array 1 증가

파라미터로 선언된 Array 개수도 count 해야 합니다.

G. selection

선택문 개수.

선택문이란, If() {}, switch() {}, 단 else 문, else if 문 제외.

또한 return a>b?a:b;도 선택문이 아님.

H. 반복문

반복문 개수.

반복문이란 While() {}, do{} while(), for() {}.

I. 리턴문

return 의 개수.

단, printf("return ;");을 했을 때 카운팅 하면 안 됨.

문법상 return 역할을 할 때만 카운팅.

J. 변수의 선언 위치 자유

제공된 ANSI C 문법을 수정하여 기존의 변수의 전방 선언만 허용하는 문법을 변수의 선언이 자유롭도록 문법 수정.

K. 주의사항

1) 이 과제는 문맥상의 오류를 찾는 과제.

int a=4; int a=10; 과 같은 문장은 실제 컴파일 때는 에러가 뜨지만 여기에서는 메모리 상의 문제가 아닌 문맥상의 오류만 찾으므로 에러가 뜨지 않아도 됩니다.

- 2) Overflow 나 `int a = 2.4;` 과 같은 예러는 잡지 않습니다.
- 3) 테스트 파일이 `#include<stdio.h>` 같은 헤더 파일로 시작할 수 있습니다.
- 4) 테스트 파일이 `#define a 5` 와 같이 `define` 처리가 가능해야 합니다. 이때 마지막 파라미터가 상수인 경우만 구현하시면 됩니다.
- 5) `auto`, `break`, `case`, `const`, `continue`, `default`, `double`, `enum`, `extern`, `float`, `goto`, `long`, `register`, `short`, `signed`, `sizeof`, `static`, `struct`, `typedef`, `union`, `unsigned`, `volatile` 처리가 가능해야 합니다.
- 6) 변수에 값을 대입할 때, 10 진수 뿐만 아니라 16 진수, 8 진수 또한 가능해야 합니다
- 7) type casting ex) `(float)a` 처리가 가능해야 합니다.
- 8) 테스트 파일에서 `for`, `while`, `do while`, `if`, `switch` 등 에는 `{}`가 있을 예정입니다.
- 9) 비트연산이 가능해야 합니다.
- 10) 구조체로 변수를 선언 할 수 있어야 합니다.
- 11) 테스트 파일에서 함수 선언은 기본 자료형인 `void`, `int`, `float` ... 으로 할 예정입니다. `struct` 와 같은 임의의 자료형으로 반환하는 함수는 없을 예정입니다. 단 파라미터에는 `struct` 로 선언된 자료형이 들어 갈 수 있습니다.
- 12) 테스트케이스는 예러없이 실행 되는 C 코드입니다.

4. 문서화

구현한 부분에 대한 **코드 분석** 및 **사용 이유** 등의 설명을 **자세히**

문서화하도록 한다. 문서 자체의 비중이 코드의 비중보다 더욱 큼을 유념.

5. 제출 결과물

A. 제출 파일

- 1) hw4.l hw4.y hw4.pdf hw4.tex

B. 보고서 내용

- 1) yacc 에 대한 설명 (동작 방식)
- 2) Lex, yacc 코드 옮겨 적은 후 주석 작성

C. 제출 마감 시간 및 장소

- 1) 제출 시작 시간 : 5 월 15 일
- 2) 제출 마감 시간 : 5 월 29 일 밤 11 시 59 분
- 3) 제출 방법 : submit pem_ta hw4_
(_ == 3 분반 : c, 4 분반 : d, 5 분반 : e)

6. 유의 사항

A. 채점 기준

- 1) 어느 정도 이해하였는가? (보고서를 통해 파악)
- 2) 프로그램의 구동은 잘 되는가?

B. 감점 사항

- 1) 부정행위 발견 시 관련 학생 모두 0 점 처리하며, Latex 파일에서 과제에 대한 이해도 부족으로 부정행위로 간주될 경우 큰 감점
- 2) 프로그램의 출력이 제시된 출력과 다르면 0 점

C. 질문 사항

- 1) 제목 첫 글자에는 pl 추가
- 2) 밤 11 시 이후에 받은 메일은 다음날 답장
- 3) 질문 : hyochangkevin@gmail.com

- 4) 수업시간에 여러번 알려 주었거나 강의록에 명시되어 있는 부분은
답변하지 않음.
- 5) 간단한 구글링을 통해 해답을 얻을 수 있는 질문은 답변하지 않음.
- 6) 5.28 일 23:00 이후의 질문은 답변하지 않음.