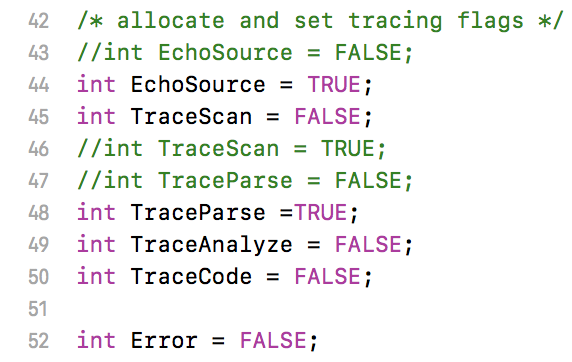
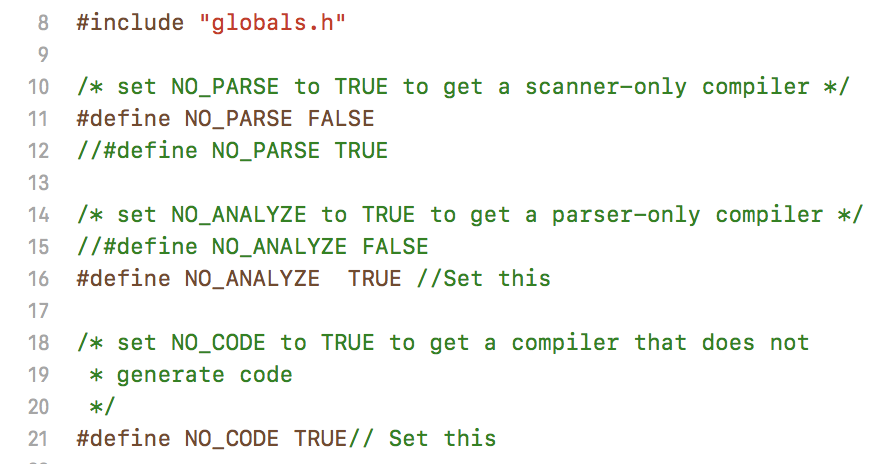
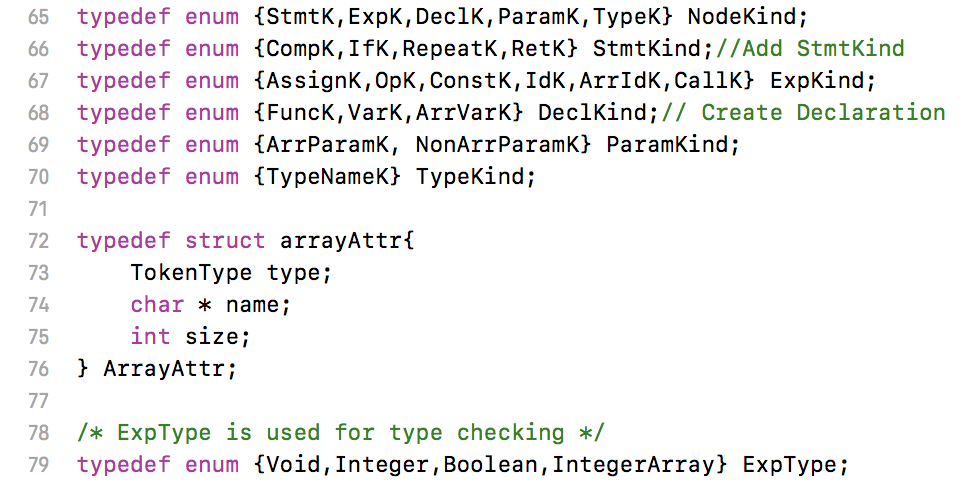
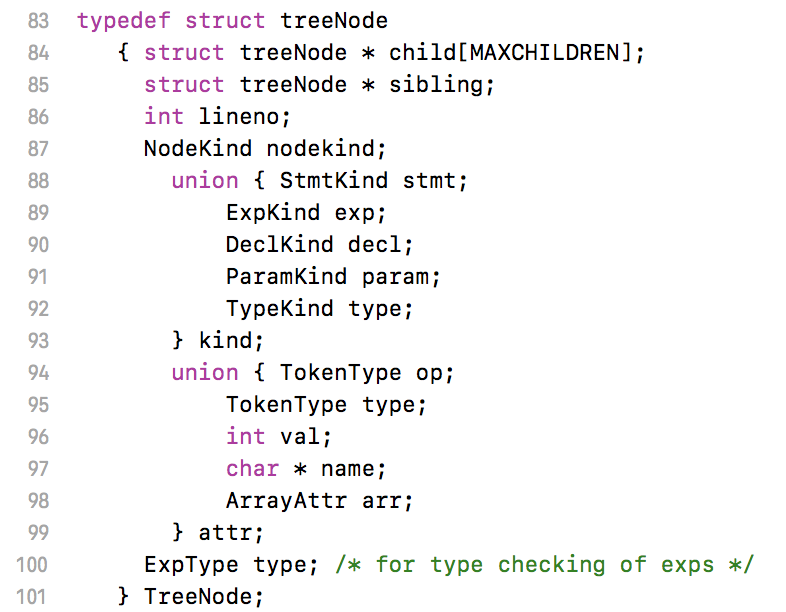
컴파일러 프로젝트#2 Parser 보고서

2013011554이새한

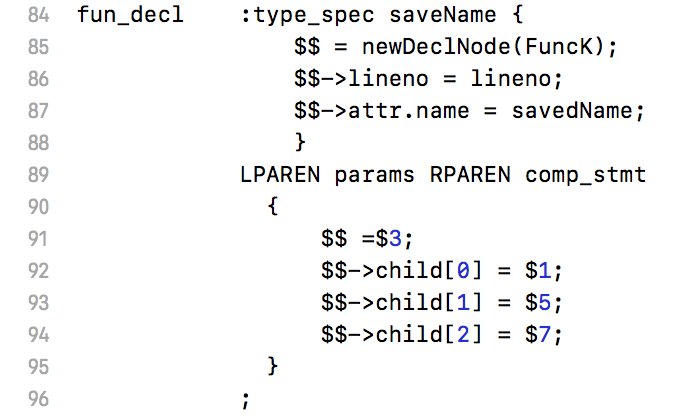
1. Compilation method and environment  
   MacBook Pro(High Sierra) Xcode & vim
2. Explanation about how to implement and how to operate
3. main.c  
   We need to modify some variables to operate parser only compiler  
   NO\_PARSE = FALSE;  
   NO\_ANALYZE = TRUE;

EchoSource = TRUE;  
TraceScan = FALSE;

TracePalse = TRUE  


1. globals.h  
   여기서는 TreeNode에 대한 디자인 개량이 필요하다. 최대한 tiny.y의 디자인을 따라갔다.우선 큰 노드 종류를 나눠야한다. tiny.y에서는 stmt와 exp는 두개다 열거되는 성격의 노드들이다. 그에 따라 Appendix에서 접미사로 -list로 되는 declaration, param을 새 종류로 추가하였다. 예외적으로 Type는 declaration과 param에서 두 노드에서도 파싱이 되기때문에 추가하였다. 그에 따라 세분화되는 nonterminal 분할은 Appendix.A.2를 따라갔다.  
     
   또한 tiny와 다르게 C-minus에서는 배열도 선언이 가능하므로 이에 따라 TreeNode의 struct또한 확장시켰다.  
   
2. cminus.l

기본적으로 문법들을 넣는 것은 상동하다. 하지만, 일부 파싱 문법같은 경우에는 String을 끌어와야 되는 곳이 2개이상을 넘기게 되어 하나가 덮어씌워져서 저장을 못할 때가 있다. 이때는 1개가 나오는 시점에서 바로 해당 동작을 시행해 주고 그 다음 nonterminal들이 savednumber, savedname의 유일성이 보장된다면 한꺼번에 시행해준다.



위의 fun\_decl을 예를 들면 saveName이 나온 직후 name을 저장해두지 않으면 뒤에 나오는 ‘param’ nonterminal 내에서 savedName을 덮어쓰기 때문에 이전에 미리 attr.name을 미리 초기화 해두는 것이다.

1. util.c

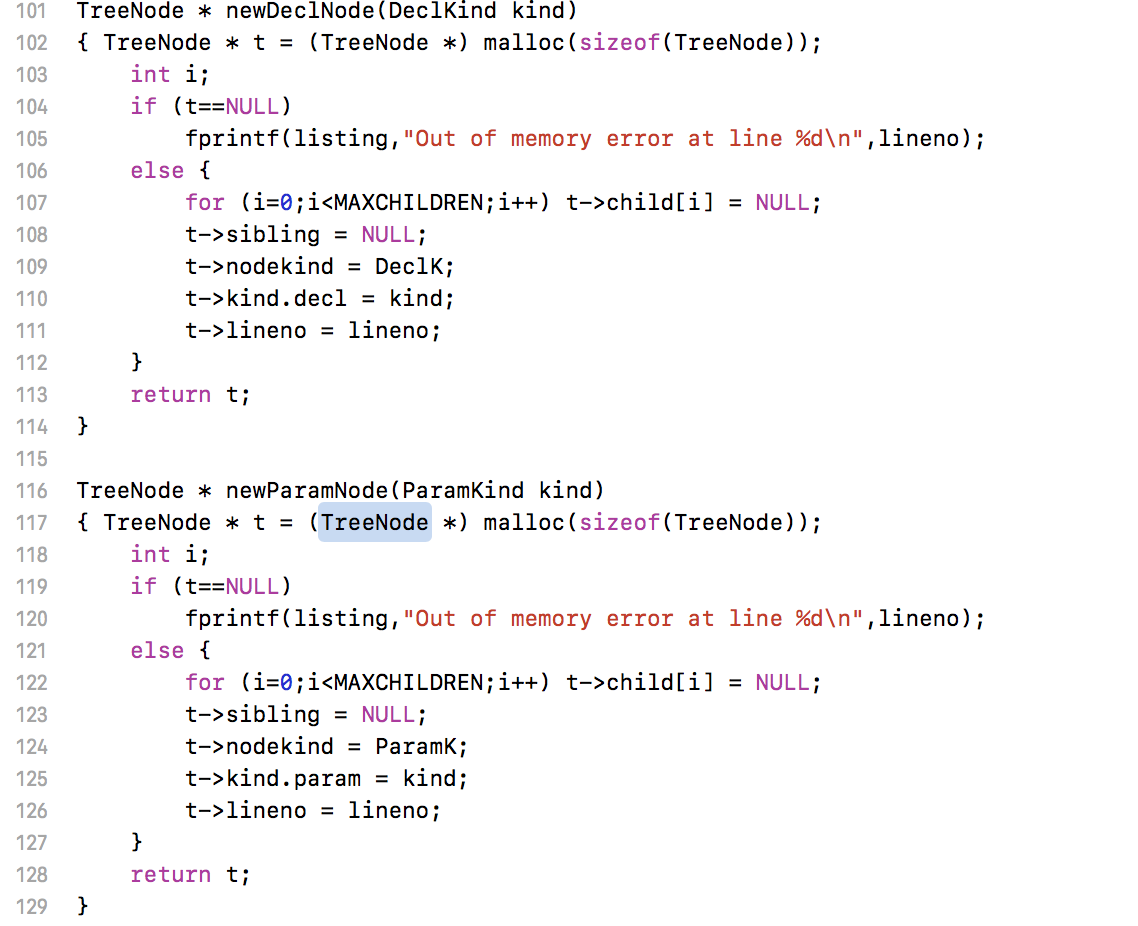
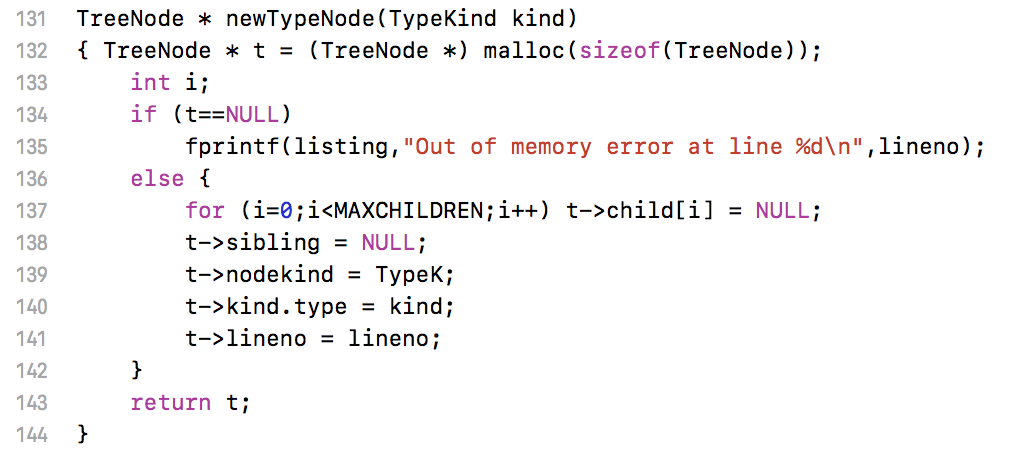
TreeNode의 디자인들이 좀더 구체화 된 만큼 그에 따라 Tree  
Node를 생성시키는 함수를 추가하였다. ( NewdeclNode, new ParamNode, newTypeNode함수). 또한 printTree함수에서 새로운 nonterminal이 생겼으므로 과제 pdf 예시출력에 맞게 기존 구문을 이용하여 변형하였다.   
  


그림 1 새 TreeNode초기화 예시

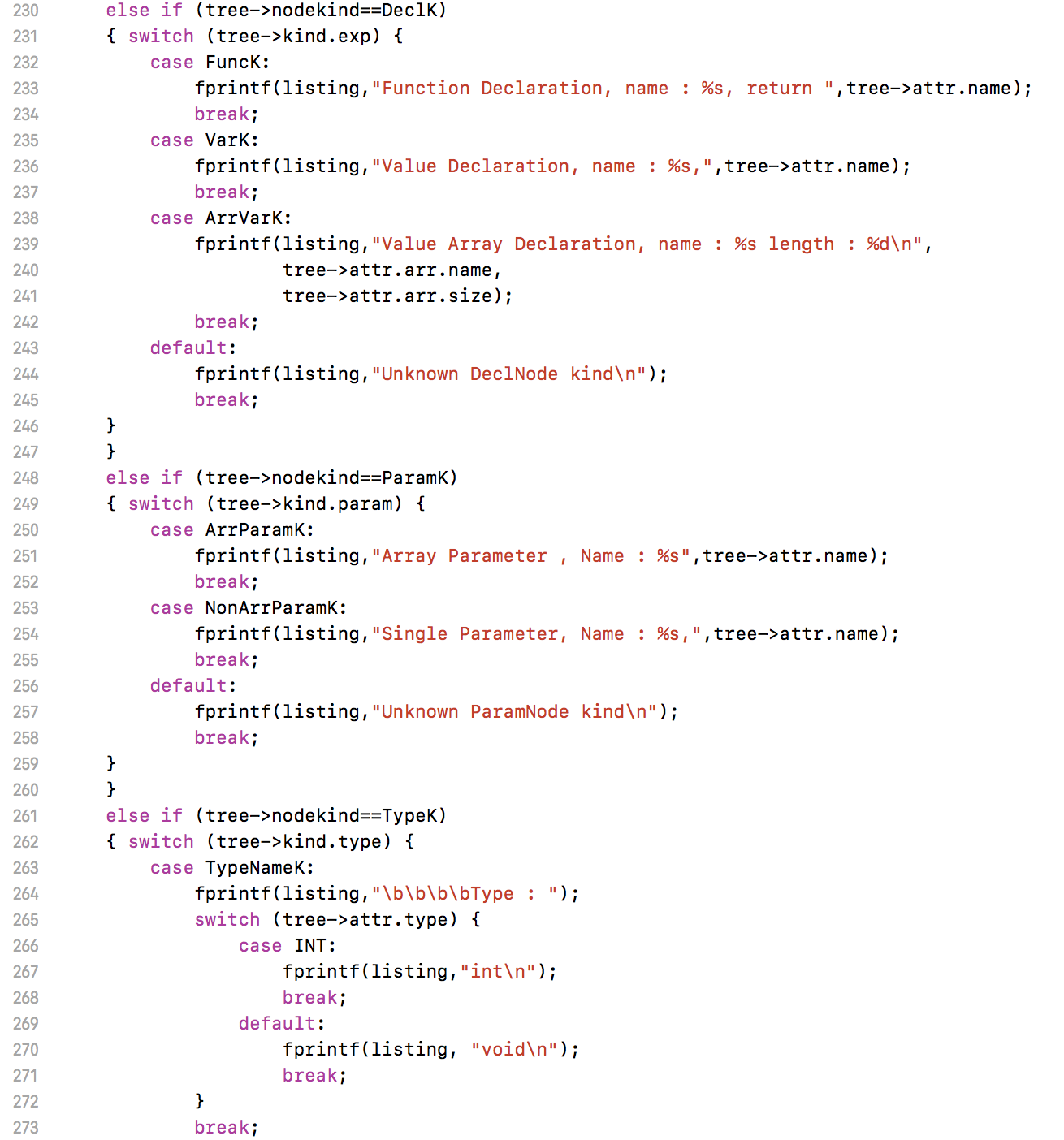
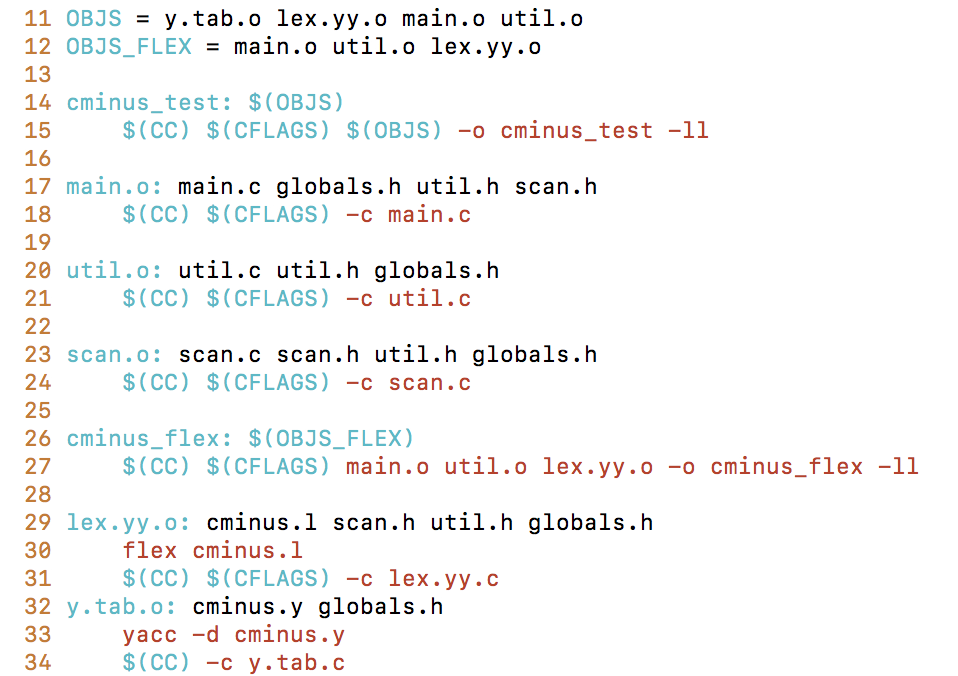


그림 printTree 일부분

**5) Makefile**

**  
 yacc가 추가됬으므로 이에 따라 의존성도 추가한다.**

1. **Example and Result Screenshot**
2. **test.cm**/\* A program to perform Euclid's

Algorithm to computer gcd \*/

int gcd (int u, int v)

{

if (v == 0) return u;

else return gcd(v,u-u/v\*v);

/\* u-u/v\*v == u mod v \*/

}

void main(void)

{

int x; int y;

x = input(); y = input();

output(gcd(x,y));

}

1. **Result Screenshot**  
   