**컴파일러 과제1-2**

유재우교수님

소프트웨어학부 20163231 신승은

※※과제1-1를 완벽하게 못 끝내서 결과가 달라지진 않았습니다. 완성은 못했지만 해본데까지 작성하여 제출했습니다. 죄송합니다. ※※

개발환경 : Visual Studio 2019

**<코드>**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#include <math.h>

#define MAX\_SIZE 20

int num;

char\* ptr;

enum token {

NUMBER, PLUS, STAR, LPAREN, RPAREN, END }token;

enum t {

INTTYPE,FLOATTYPE

};

static struct s {

enum t type;

// enum token token;

union {

int i;

float f;

}value;

}s1;

void\* get\_token(char\* ptr);

int term(char\* ptr);

int factor(char\* ptr);

int expression(char\* ptr);

void cError(int i);

void\* get\_token(char\* ptr) {

struct s s1;

char tmp = 0;

//static char ch=' ';

int j = 1;

//ptr = ch;

//ptr = get\_token(ptr);

if ('0'<=ptr&&'9'>=ptr) {

do {

num = ptr - '0';

tmp = ptr;

// printf("%d\n", tmp);

\*ptr= getchar();

//정수 두자릿수까지만 받을수 있음

if ('0'<=ptr&&'9'>=ptr) {

int k = 10;

j+=1;

tmp \*= pow(k, j);

ptr += tmp;

s1.value.i = ptr - '0';

num = s1.value.i;

ptr = getchar();

s1.type = INTTYPE;

return (NUMBER);

}

} while ('0'<=ptr&&'9'>=ptr);

if (ptr == '.') {

//실수는 첫째자리까지만 받음

ptr = getchar();

s1.value.f = atof(ptr);

s1.value.f /= 10;

(float)num += s1.value.f;

ptr = getchar();

s1.type = FLOATTYPE;

return (NUMBER);

}

}

else if (ptr == '+') {

//token = PLUS;

return (PLUS);

}

else if (ptr == '\*') {

return (STAR);

}

else if (ptr == '(') {

return LPAREN;

}

else if (ptr == ')') {

return RPAREN;

}

else if (ptr == EOF) {

return END;

}

else

cError(1);

\*(ptr++);

return ptr;

}

int expression(char\* ptr) {

int result;

result = term(ptr);

while (token == PLUS) {

ptr=get\_token(ptr);

result = result + term(ptr);

}

return (result);

}

int term(char\* ptr) {

int result;

result = factor(ptr);

while (token == STAR) {

ptr=get\_token(ptr);

result = result \* factor(ptr);

}

return (result);

}

int factor(char\* ptr) {

int result;

if (token == NUMBER) {

result = num;

ptr=get\_token(ptr);

}

else if (token == LPAREN) {

ptr=get\_token(ptr);

result = expression(ptr);

if (token == RPAREN)

ptr=get\_token(ptr);

else

cError(2);

}

else

cError(1);

return (result);

}

void main() {

char\* ptr = (char\*)malloc(sizeof(char) \* MAX\_SIZE);

printf("input :");

char ch = getchar();

ptr=get\_token(ch);

int result;

result = expression(ptr);

if (token != END)

cError(3);

else

printf("%d\n", result);

free(ptr);

}

void cError(int i) {

switch (i) {

case 1: printf("error: number or '(' expected\n"); break;

case 2: printf("error: ')' expected\n"); break;

case 3: printf("error: EOF expected\n"); break;

}

exit(1);

}

온라인강의를 들으면서 이해도는 높아졌지만 코드작성까지는 시간이 많이 부족했습니다. 강의 자료에 있던 토큰 자르는 함수를 참고하여 get\_token을 다시 작성해봤지만 token값을 읽어오지 못했고, getchar을 이용하여 다음 문자를 가져오는데 어려움을 겪었습니다.

문자열에서 정수와 실수를 구분하기 위해서 ‘.’을 읽으면 atof함수를 사용해 자료형을 float형으로 바꿔준 뒤, 10으로 나눠 소수점 첫째 자리 까지만 받을 수 있게 작성했습니다. 나눈 값을 num에 합하여 float형의 num으로 만듭니다.

현재 코드 상태에서 디버깅을 해본 결과 48번 line의 \*ptr= getchar();에서 예외가 발생하여 멈추게 됩니다.

**실행화면**

