/\*

240109 1번 - 중위표기식을 후위표기식으로 변환하는 알고리즘

스택, 큐, 전위,중위,후위표기법

피연산자 : 항상 출력

연산자 : 아래 과정을 스택에 저장될 때까지 반복

if (새 연산자의 우선순위 > 스택의 top 연산자 우선순위)

|| (스택이 비었음) then

스택에 저장

else

top의 연산자를 pop하여 출력

식을 다 읽었으면 스택이 빌 때까지 연산자를 pop하여 출력한다.

후위표기식의 장점 : 괄호가 없으므로 고려할 사항이 적음

연산 방법

1) 피연산자는 스택에 push

2) 연산자는 필요한 만큼의 피연산자를 스택에서 pop하여 연산, 연산결과를 다시 스택에 push.

마지막에 스택에 남은 값이 결과

왜 컴파일러는 중위표기식에서 후위표기식으로 변경해야 할까? 즉, 컴파일러에선 후위표기식만 사용

A) 중위표기식은 연산자의 우선순위 때문에 왼쪽에서 오른쪽으로 연산이 되지 않기 때문이다.

중위표기식은 괄호를 가지고 있기 때문이기도 하다.

입력으로 주어진 중위표기식을 읽어가며 처리

스택을 활용하여 연산자의 우선순위 반영, 결과로서 후위표기식 출력.

수식 표기방식의 변경 규칙

1) 피연산자는 그대로 출력

2) 스택이 비어있을때 만나는 연산자는 무조건 스택에 add

3) 지금 처리하려는 연산자가 스택의 top 연산자보다 우선순위가 높으면 스택에 add한다. 아니면, 스택의 연산자를 delete하여 출력한다.

4) 표기식을 끝까지 다 읽으면 스택의 연산자를 모두 delete하여 출력한다.

변경 규칙(2) 앞의 규칙을 적용하면서 괄호가 나오면 다음 규칙을 추가로 적용한다.

1) 왼쪽 괄호는 스택에 들어올 때는 무조건 add되도록 한다. 일단 스택에 들어오면 우선순위가 가장 낮아져서

다음 들어오려는 연산자를 add한다.

2) 오른쪽 괄호는 왼쪽 괄호가 나올때까지 스택 안의 모든 연산자를 출력한다.

후위식 sample

a / b - c + d \* e - a \* c -> a b / c - d e \* + a c \* -

a/(b - c + d) \* (e - a) \* c -> a b c - d + / e a - \* c \*

if 괄호가 들어간다면?

2 + (4 + 8 / (3 - 2) \* 5) / 6

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#define MAX\_EXPR\_SIZE 50

#define MAX\_STACK\_SIZE 30

//////////////////////////////////////////////////////////////////////////////////////////////////

char pexpr[MAX\_EXPR\_SIZE];

int stack[MAX\_STACK\_SIZE];

int top;

int delete\_stack();

void add\_stack(int data);

int is\_operator(char c);

int cal(void);

void main()

{

printf("input the expression as a postfix notation : \n");

gets(pexpr);

printf("Evaluation Value: %d\n", cal());

}

int cal(void)

{

char symbol;

int op1, op2, n = 0;

top = -1;

symbol = pexpr[n++];

while (symbol != '\0') {

if (is\_operator(symbol)) {

//피연산자를 삭제하여 연산을 수행한 후, 그 결과를 스택에 삽입한다.

op2 = delete\_stack();

op1 = delete\_stack();

switch (symbol) {

case '+': add\_stack(op1 + op2);

break;

case '-': add\_stack(op1 - op2);

break;

case '\*': add\_stack(op1 \* op2);

break;

case '/': add\_stack(op1 / op2);

} // switch

}

else

add\_stack(symbol - '0');

symbol = pexpr[n++];

}

return delete\_stack();

}

int delete\_stack()

{

int data;

if (top == -1) {

printf("Stack is empty. \n");

exit(1);

}

data = stack[top]; top--; return data;

// return stack[top--];

}

void add\_stack(int data)

{

if (top >= MAX\_STACK\_SIZE - 1) {

printf("Stack is full. \n");

exit(1);

}

++top; stack[top] = data;

// stack[++top] = data;

}

int is\_operator(char c)

{

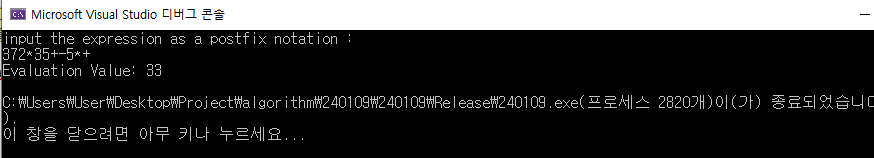
if (c == '+' || c == '-' || c == '\*' || c == '/')

return 1;

else

return 0;

}



/\*

240109 2번 문제 - 구조체, 포인터

포인터는 메모리 주소를 값으로 가지는 데이터형

C언어에서는 어떤 타입 t에 대해서 t의 포인터 타입이 존재한다.

int \* float \*

포인터 타입에는 주소연산자(&), 역 참조(간접 지시) 연산자(\*)가 사용된다.

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

void main()

{

int\* p, q;

float\* fp, x;

p = &q;

\*p = 199;

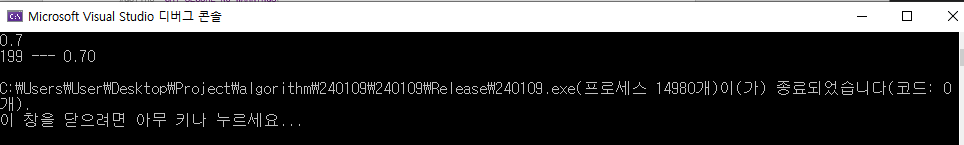
fp = &x; -> 일부러 틀린코드를 보여주기 위해 빠졌음

scanf("%f", fp);

x = \*fp;

printf("%d --- %.2f\n", q, x);

}



/\*

240109 3번 문제 x=10, y=30, z=5의 값이 입력되었을 때를 예로 하여 다음 프로그램을 수행시켜보자.

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

void swap(int\* px, int\* py)

{

int temp;

1) temp = \*px;

2) \*px = \*py;

3) \*py = temp; // 이 셋은 순서 바꾸기에서 같이 세트로 나오는 것이다.

}

void main()

{

int x, y, z;

printf("세 수를 입력하시오. \n");

scanf("%d %d %d", &x, &y, &z);

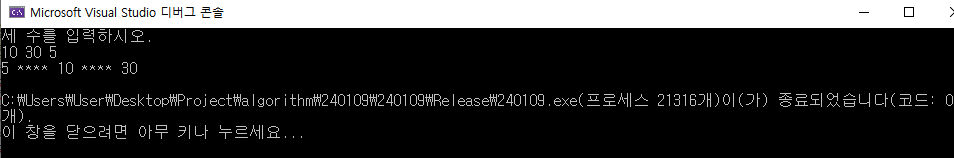
if (x > y) swap(&x, &y);

if (y > z) swap(&y, &z);

if (x > y) swap(&x, &y);

printf("%d \*\*\*\* %d \*\*\*\* %d\n", x, y, z);

}



/\*

240109 3번 문제 x=10, y=30, z=5의 값이 입력되었을 때를 예로 하여 다음 프로그램을 수행시켜보자.

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

int add\_product(int\* a, int b)

{

int temp = b;

b = \*a + b;

\*a = \*a \* temp;

return b;

}

void main()

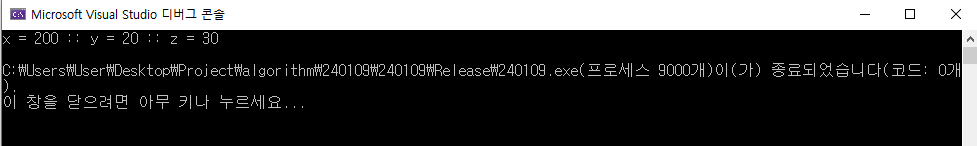
{

int x = 10, y = 20, z;

z = add\_product(&x, y);

printf("x = %d :: y = %d :: z = %d\n", x, y, z);

}



/\*

240109 4번 문제 포인터와 배열

배열의 이름은 배열 첫 데이터의 주소, 배열의 전체의 대표 정보이다.

C언어에서는 배열을 함수의 파라미터로 넘겨 줄 때 배열의 이름을 전달한다.

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

float trunc\_sum(float\* data)

{

float sum = 0.0;

int i, ivalue;

for (i = 0; i < 10; i++) {

ivalue = data[i];

sum += (data[i] - ivalue);

}

return sum;

}

void main()

{

float xarray[10], fsum = 0.0;

int i;

printf("Enter 10 reals:\n");

for (i = 0; i < 10; i++) {

printf("Enter %dth reals:\n", i+1);

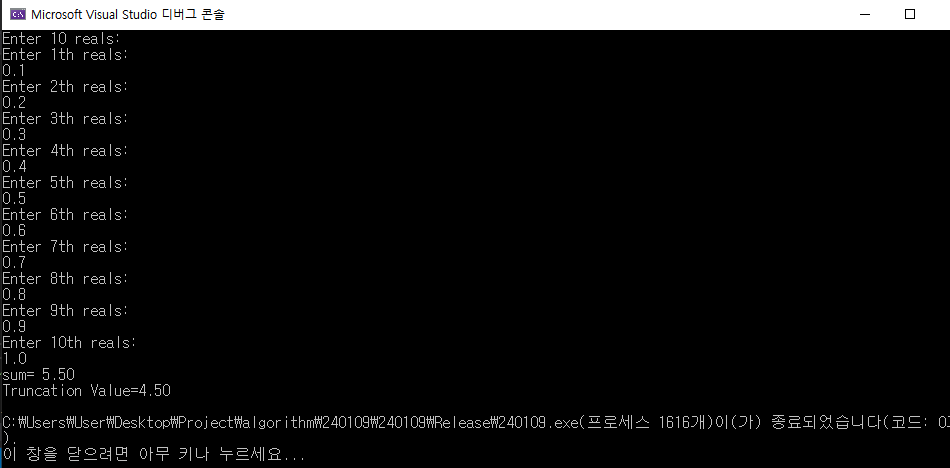
scanf("%f", xarray + i); // xarray = &xarray[0]

fsum = fsum + \*(xarray + i);

}

printf("sum= %.2f\n", fsum);

printf("Truncation Value=%.2f\n", trunc\_sum(xarray));



/\*

240109 5번 문제 포인터와 배열

memory allocation

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

//type( \*) malloc(sizeof(type)) {

//}

void main()

{

int \*ip;

float \*fp;

ip = (int\*)malloc(sizeof(int)); //동적 메모리 할당

fp = (float\*)malloc(sizeof((float));

\*ip = 2020; \*fp = 7.123; // 동적 메모리 사용

printf("year = %d : point = %3.f\n", \*ip, \*fp);

free(ip); free(fp); //동적 메모리 반납

}

컴파일 오류 발생

/\*

240109 6번 문제 161p 순차리스트 구조에서 연결리스트 구조로

순차리스트는 자료구조 안의 데이터가 메모리에 연속적으로 저장되고 그 순서에 의하여 데이터가 처리됨

배열의 인덱스 --- for 문의 LCV

배열로 구현된 순차리스트는 실행 전에 그 크기가 정해져 있어야 함.

크기가 가변적인 자료구조인 경우

데이터가 중간에 삽입되는 경우

자료구조의 중간으로부터 데이터

연결리스트를 생성하기 위해 필요한 기능

1) 노드의 구조 정의

2) 노드 생성 : malloc() 함수 사용

3) 노드의 데이터 필드와 링크 필드에 값을 할당

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

//////////////////////////////////////////////////////////////////////////////////////////////////

typedef struct simple\_list\* simple\_pointer;

typedef struct list\_node\* list\_pointer;

struct simple\_list {

char state[3];

int count;

simple\_pointer next;

};

simple\_pointer state\_list();

void append(simple\_pointer ptr, simple\_pointer inode);

void print\_list(simple\_pointer ptr);

void main()

{

simple\_pointer ptr, inode;

ptr = state\_list();

inode = (simple\_pointer)malloc(sizeof(struct simple\_list));

strcpy(inode->state, "TX");

inode->count = 7;

append(ptr, inode);

print\_list(ptr);

}

simple\_pointer state\_list()

{

simple\_pointer node1, node2;

node1 = (simple\_pointer)malloc(sizeof(struct simple\_list));

node2 = (simple\_pointer)malloc(sizeof(struct simple\_list));

strcpy(node1->state, "NY");

node1->count = 5;

node1->next = node2;

strcpy(node2->state, "FL");

node1->count = 10;

node1->next = NULL;

return node1;

}

void append(simple\_pointer ptr, simple\_pointer inode)

{

simple\_pointer before;

while (ptr != NULL) {

before = ptr;

ptr = ptr->next;

}

before->next = inode;

inode->next = NULL;

}

void print\_list(simple\_pointer ptr) {

printf("The singly linked list contatins \n");

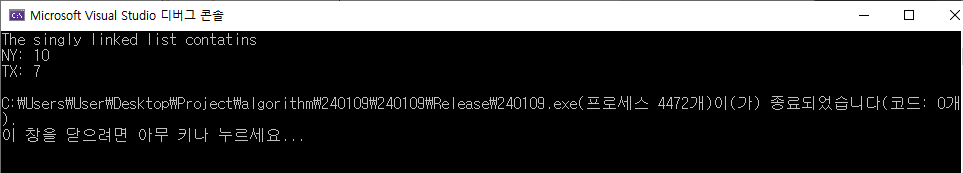
while (ptr != NULL) {

printf("%s: %d\n", ptr->state, ptr->count);

ptr = ptr->next;

}

}



참고 : C4703 오류(before 초기화 요구) 발생, 옵션-디버그-c/c++ 프로젝트속성에서 sdl옵션검사를 아니오로 설정해야함

/\*

240109 6번 문제 구조체

\*/

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_CNUM 50

//////////////////////////////////////////////////////////////////////////////////////////////////

typedef struct course {

char courseid[10]; char tname[20]; unsigned snum; unsigned roomnum;

} cinform;

main(int argc, char\* argv[])

{

int i = 0, dnum;

char sprof[20];

unsigned sroomnum;

cinform cdb[MAX\_CNUM];

FILE\* coursedb;

if ((coursedb = fopen(argv[1], "r")) == NULL) {

printf("데이터 파일을 열 수 없습니다 \n");

exit(1);

}

while (fscanf(coursedb, "%s %s %u %u", cdb[i].courseid,

cdb[i].tname, &(cdb[i].snum), &(cdb[i].roomnum)) != EOF)

i++; dnum = i;

printf("수강인원이 40명 이상인 강좌 코드 번호를 출력하시오.\n");

for (i = 0; i < dnum; i++)

if (cdb[i].snum >= 40) printf("%s\n", cdb[i].courseid);

printf("검색하고자 하는 교수이름 입력 : ");

scanf("%s", sprof);

printf("%s 교수가 강의하는 강좌코드와 강의실 번호를 출력하시오.\n", sprof);

for (i = 0; i < dnum; i++)

if (!strcmp(cdb[i].tname, sprof)) printf("%s\t%u\n", cdb[i].courseid, cdb[i].roomnum);

printf("검색하고자 하는 강의실 번호 입력 : ");

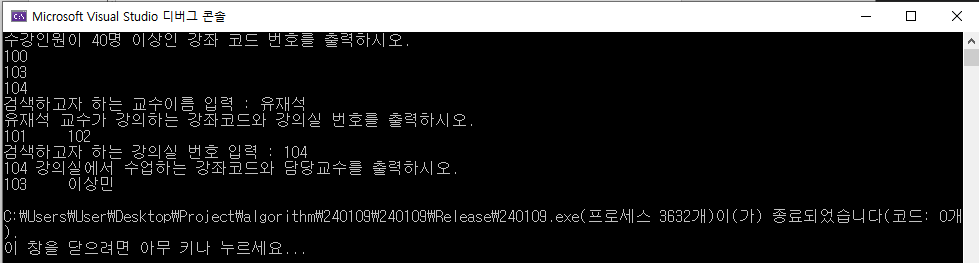
scanf("%u", &sroomnum);

printf("%u 강의실에서 수업하는 강좌코드와 담당교수를 출력하시오.\n", sroomnum);

for (i = 0; i < dnum; i++) if (cdb[i].roomnum == sroomnum)

printf("%s\t%s\n", cdb[i].courseid, cdb[i].tname);

}



명령인수 data.txt의 경로 : 프로젝트의 탐색기 열기의 경로에 txt를 붙여넣는다

Data.txt의 내용

100 강호동 40 101

101 유재석 35 102

102 신동엽 30 103

103 이상민 50 104

104 이수근 45 105