괄호 검사

괄호검사를 위해 스택을 사용합니다. 스택안에는 오직 괄호만 들어가며 비교대상도 괄호만 해당합니다.

우선 여는 괄호를 만나면 stack 안에 순서대로 push를 이용해 계속 넣습니다. 그러다 닫는 괄호가 나온다면 pop을 이용하여 return되는 괄호가 짝이 맞을 경우 계속 진행하고 짝이 맞지 않는다면 괄호가 잘못되어 있는것이기에 실패를 출력합니다. 여기서 기본적인 가정은 왼쪽괄호와 오른쪽 괄호의 개수가 맞아야하며 왼쪽 괄호가 무조건 오른쪽 괄호보다 먼저 나와야 한다는 것 입니다. 다만 여기서 일반적인 수학적 연산 규칙인 소괄호, 중괄호, 대괄호 순서로 계산되어야 하는 사실은 고려할 필요가 없습니다. 즉 모양만 맞추면 되지 우선순위를 정해줄 필요는 없는 것입니다. 이것을 간단하게 정리하자면 다음과 같습니다.

1. (, {, [와 같이 여는 괄호가 나온다면 모두 stack에 push하기

2. 닫는 괄호가 나온다면

1) stack이 비어있는 상태라면 닫는괄호가 먼저 나왔으니 error

2) 바로 이전 괄호와 짝이 맞는지 확인하여 짝이 맞다면 pop으로 삭제하여 다음 단계로 넘어가고 짝이 다르다면 실패를 출력한다.

How ? => 결국 pop해서 return되는 값(이전 괄호)과 현재 괄호가 짝이 맞는지 확인. 같다면 이미 pop으로 이전 여는 괄호는 삭제됐으므로 바로 넘어가면 됨

3. 괄호검사가 끝난 뒤(반복문이 끝난 뒤) is\_empty가 참이면 성공, 참이 아니라면(괄호가 남았다면) error

위의 과정을 코드로 나타내면 다음과 같습니다.

우선 기본적인 stack 구조체 선언과 init, is\_empty, is\_full, push, pop함수는 기존 stack을 구현하는

방법과 동일합니다. 다만 pop에서 return값은 괄호 (, {, [ 중 하나이므로 char형을 return형으로

정해줍니다.

가장 주 기능을 담당하는 check 함수는 괄호를 검사하는 함수이며 문자열 길이만큼 반복을 돌리기 위해

strlen을 사용하여 for문을 취했고 그 후 코드의 간결화를 위해 if else 조건문 대신 switch문을

사용했습니다. 또한 main함수는 식을 입력받기 위해 fgets함수를 이용하여 100자까지만 입력받도록

하였습니다.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct stack{

char stack[100];

int top;

}StackType;

void init(StackType\*s){

s->top=-1;

}

int is\_empty(StackType\*s){

return (s->top==-1);

}

int is\_full(StackType\*s){

return(s->top == 99);

}

void push(StackType\*s, char item){

if(is\_full(s)){

printf("스택 포화 에러\n");

exit(1);

}

else

s->stack[++(s->top)]=item;

}

char pop(StackType\*s){

if(is\_empty(s)){

printf("스택이 비었습니다\n");

exit(1);

}

else

return s->stack[(s->top)--];

}

int check(char\*input){

StackType s;

char a, b;

int i=0,n=strlen(input);

init(&s);

for(i=0;i<n;i++){//문자열 길이만큼 반복문 돌면서

a=input[i];

switch(a){

case '(':

case '{':

case '[':

push(&s,a);

break;

case ')':

case '}':

case ']':

if(is\_empty(&s)){ //stack이 비어있었다면

return -1;

}

else{

b=pop(&s);

if(b=='(' && a!=')' || b=='{' && a!='}' || b == '[' && a!=']'){//닫는괄호가 이전괄호와 짝이 맞는지!(안맞으면 에러)

return -1;

}

else

break;

}

}

}

if(is\_empty(&s))

return 0;

else

return -1;

}

int main(void){

int i=0;

char input[100]={'0'};

printf("문자열 입력 : ");

fgets(input,100,stdin);

if(check(input)==0) printf("괄호가 적절함");

else printf("괄호가 적절하지 않음");

return 0;

}