2023-2학기 C프로그래밍 과제#1 보고서

201912070 박승민

우선 전역변수를 선언함으로써, 밑에 따로 만들 메소드들이 전역변수를 사용할 수 있도록 하였습니다. 이때, 스택의 크기는 과제 예시코드로 들어 주신 크기 10으로 설정했습니다.

int stack[10] = { 0 }; // 배열 크기 10인 배열 생성

int top = -1; //스택의 현재 위치

while문에서 기존에는 1: push, 2 : pop, 0 : exit 선택지가 있었는데, 이대로는 조금 심심해서 3번 stack status 항목을 추가하여 현재 stack에 어떤 것이 들었는지 확인하는 기능을 추가하였습니다.

Switch case 문에서 case가 1일 때(push할 때 ) 스택의 현재 크기가 배열의 크기인 10을 초과하면 더 push 할 수 없기 때문에 조건문에서 이를 걸러주어야 합니다. 따라서 스택의 현재 위치인 top이 10이상이면 더 이상 push 할 수 없기 때문에 top<10 일 때 push 할 수 있게 하고, else 로는 스택이 꽉 찼음을 printf로 표시하게 하였습니다.

Case가 2일 때 (pop 할 때), top이 -1이면 스택 안에 아무것도 없다는 뜻이므로, 조건문을 top >=0으로 하여 스택이 비어 있지 않을 때 pop을 하도록 합니다.

Case가 3일때는 for문을 통하여 반복문이 현재 스택의 크기까지 반복하도록 설정한 후, 스택 배열을 출력하도록 하였습니다.

push 메소드는 스택에 값을 하나 받은 것이기 때문에 top의 크기를 하나 늘려주고, 스택이 얼마나 쌓였는지 모르기 때문에 스택의 값을 출력하도록 하였습니다. 그리고, 매개변수로 전달받은 값을 스택에 넣어주었습니다.

Pop 메소드는 현재 스택의 값을 0으로 만들고, top의 값을 1을 줄여주고, pop이 제대로 되었는지 확인하기 위해 현재 스택의 개수를 확인하도록 설계하였습니다.

int push(int push\_num) {

top = top + 1;

printf("스택의 개수(최대 10): %d\n", top+1); // 현재 스택이 몇까지 쌓였는지 보여줌

stack[top] = push\_num;

}

int pop() {

stack[top] = 0;

top--;

printf("스택의 개수(최대 10): %d\n", top+1); // 현재 스택이 몇까지 쌓였는지 보여줌

}