```c
#include <stdio.h>
#include <stdlib.h>

// Global variables. All functions recognize these
int top = -1, MAX, *stack;

int isFull(){
    return top == MAX - 1;
}

int isEmpty(){
    return top == -1;
}

int check(){
    if(isFull()){
        printf("FULL\n");
        return 1;
    }
    else if(isEmpty()){
        printf("EMPTY\n");
        return -1;
    }
    else{
        printf("NOT EMPTY\n");
        return 0;
    }
}

void push(int n){
    if(isFull())
        printf("FULL\n");
    else
        stack[++top] = n;
}

void pop(){
    if(isEmpty())
        printf("EMPTY\n");
    else
        printf("%d\n", stack[top--]);
}

int execute(){
    char c = getchar();
    int x;
    switch(c){
        case 'E':
            scanf("%d", &x);
            getchar(); // Catch the newline
            push(x);
            return 1;
        case 'D':
            getchar(); // Catch the newline
            pop();
            return 1;
        case 'C':
            getchar(); // Catch the newline
            check();
            return 1;
        case 'X':
            return 0; // Stop executing
        default:
            printf("Error!!");
            return -1;
    }
}

int main(){
```

```
    scanf("%d\n", &MAX);
    stack = (int*)malloc(MAX * sizeof(int));

    // Keep executing till not terminated
    // ; is an empty statement
    while(execute());

    // Be careful - do not say
    // while(execute())
    // return 0;
    // This will return 0 in the very first iteration
    // Instead, put an empty statement inside while loop
    return 0;
}
```