

전역변수 배열 stack 구현

5671144-강범창

1.스택을 다음과 같은 방법으로 구현한다.

배열의 전역변수 사용.

```
#define MAX_STACK_SIZE 10

typedef int element;
element stack[MAX_STACK_SIZE];
int top = -1;
```

2. 난수(1~100) 를 발생시켜서 난수가 짝이면 push , 홀이면 pop 를 실행 내용을 출력한다.

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

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

void push(element item)
{
    if (isFull())
    {
        fprintf(stderr, "stack 포화 에러 %n");
        return;
    }
    top++;
    stack[top] = item;
}

element pop()
{
    element resultElement;
    if (isEmpty())
    {
        fprintf(stderr, "stack 공백 에러 %n");
        return -1;
    }
    resultElement = stack[top];
    top--;
}
```

```

}

int main()
{
    srand(time(NULL));

    for (int i = 0; i < 30; i++)
    {
        int ranNum = rand() % 100 + 1; // 1~100
        if (ranNum % 2 == 0)
        {
            push(ranNum);
        }
        else
        {
            element item = pop();
            if (item != -1)
            {
                printf("pop : %d\n", item);
            }
        }
    }
    system("pause"); // console 창 꺼지지 않기 위해 사용합니다.
}

```

3. 앞의 2를 30회 실행하고 실행 결과를 구현 방법에 대하여 평가한다.

2를 30회 실행해본 결과.

```

stack 공백 에러
pop : 3
pop : 2
pop : 2
pop : 3
pop : 3
pop : 2
pop : 7
pop : 6
pop : 5
stack 포화 에러

```

이런 식으로 결과가 나왔습니다. 결과만 보자면 값들이 순차적으로 잘들어 가다가 후반부에 top(요소의 위치)가 전역 변수로 선언 되어있다 보니, stack 의 크기를 유동적으로 변경을 하지 못하는 문제 가 발생하여 포화 에러가 발생하였습니다. 포화 에러를 발생 시키는 것을 볼 수 있었습니다.

이 결과를 구현 방법 에 대하여 평가해보자면, 우선 전역변수로 top 과 stack 을 선언하여 스택을 구현하였습니다, 이때 push , Pop , isEmpty , isFull 함수를 만들어

이용하여 스택에 데이터를 추가 , 삭제 , 배열에 값이 비었는지, 배열에 값들이
꼭차 있는지 에 대한 기능을 구현하였습니다.