50 라인 이상의 자율 프로그램 코딩 12점

유튜브 녹화본 링크

https://www.youtube.com/watch?v=4XZcAr9KsrA

프로그램의 목적 설명

c언어로 스택의 구현

사용한 함수 설명

총 3개의 함수가 있는데,

initStack(), pop(), push()

initStack() 함수는 스택의 top(index)값과 스택의 size 그리고 동적할당의 크기를 지정해주는 함수이다.

Push() 함수는 초기화 시켜준 스택에 값을 순서대로 넣는 함수이다.

만약스택이 가득 찰 경우 Error 메시지를 띄우고 프로그램을 종료시킨다.

Pop() 함수는 순서대로 너어준 스택에 값을 가장 늦게 저장시킨 값부터 꺼내오는 함수이다. 만약 스택에 더 이상 꺼내올 값이 없다면 Error메시지를 띄우고 프로그램을 종료시킨다.

코드 라인 by 라인 설명

#include <stdio.h>

#include <stdlib.h>

//코딩에 필요한 헤더파일

typedef struct stack{

int top; //스택의 index를 표현할 top

int \*data; // 실적인 스택을 담당할 data

int max; //스택의 최대 넣을수 있는 개수

}Stack;

//stack 구조체 선언

//pop , push, initStack 구현할 함수들

void initStack(Stack\* st, int size){ // 스택을 초기화시켜주는 함수

st->data = (int\*)malloc(size\*sizeof(int)); //사이즈를 매개변수로 받아와서 동적할당

st->top = -1; //가장먼저 값을 받아왓을 때 array의 인덱스0부터 시작하게 -1로 값을 저장

st->max = size; // 받아온 size가 결국 stack의 최대 개수와 동일하기에 size를 max에 저장

}

void push(Stack\* st, int val){ //stack에 매개변수로 val을 받아온후 값을 저장

if(st->top < st->max-1){ //top의 index가 최대 개수의 -1값보다 작을경우

st->top++; //top의 인덱스를 증가시킨다.

st->data[st->top] = val;//stack의 data에 현재 top값의 index에 받아온 val값을 저장

}

Else //조건을 만족하지 못할경우

{

printf("Stack is Full\nError!\n"); //error메세지 출력후 프로그램을 종료

exit(1);

}

}

int pop(Stack\* st){ //stack의 저장된 값들을 뒤에서부터 가져옴

if(st->top >= 0){ //stack의 top이 0같거나 클경우

int val = st->data[st->top]; //새로 선언한 val에 현재스택에서top index에 위치 //한 값을 저장

st->top--; //top의 값을 1감소

return val; //val을 리턴

}

printf("Stack is Empty\nError!\n"); //위 조건으로 값을 반환하지 못햇을경우 Error메세지 //출력후 프로그램을 종료

exit(1);

return 0;

}

int main(void){

Stack s; //stack구조체 형으로 s를 선언

initStack(&s,5); //stack을 초기화

//push(&s,5); // 이라인이 실행되면 Full Error메시지출력 후 종료

push(&s,4); //값을 push

push(&s,3);

push(&s,2);

push(&s,1);

push(&s,0);

printf("pop : %d\n", pop(&s)); //pop하여 값을 printf()로 출력

printf("pop : %d\n", pop(&s));

printf("pop : %d\n", pop(&s));

printf("pop : %d\n", pop(&s));

printf("pop : %d\n", pop(&s));

//printf("pop : %d\n", pop(&s)); //이 라인이 실행될시 Empty Error메시지출력 후 종료

}

시연 데모 녹화