5645640 제갈건 0427

1. 실행코드

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include<time.h>

#define MAX\_STACK\_SIZE 100

typedef int element;

typedef struct {

int top;

element\* data; // data은 포인터로 정의된다.

int capacity; //현재 스택의 전체 크기 size

}StackType;

void init\_stack(StackType\* s, int size) {

s->top = -1;

s->capacity = size;

s->data = (element\*)malloc(s->capacity \* sizeof(element));

}

int is\_empty(StackType\* s) {

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

}

int is\_full(StackType\* s) {

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

}

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

if (is\_full(s)) {

fprintf(stderr, "스택 포화 에러\n");

return;

}

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

}

element pop(StackType\* s) {

if (is\_empty(s)) {

fprintf(stderr, "스택 공백 에러\n");

return -1;

}

else return s->data[(s->top)--]; // top의 위치만 이동 한 것이고 배열 안의 데이터는 그대로 남아있음

}

int main(void) {

StackType s1;

srand(time(NULL));

int random\_size = rand()%100;

init\_stack(&s1, random\_size);

for (int i = 0; i <= random\_size; i++) push(&s1, i \* 10);

for (int i = random\_size; i >= 0; i--) printf("Stack top : %d data : %d\n", i, pop(&s1));

printf("Stack Random\_size : %d\n", random\_size);

StackType s;

int size = 0;

printf("Stack의 크기를 입력하시오: ");

scanf("%d", &size);

init\_stack(&s, size);

for(int i = 0; i <= size; i++) push(&s, i \* 10);

for (int i = size; i >= 0; i--) printf("Stack top : %d data : %d\n", i, pop(&s));

//free(&s.data);

return 0;

}

1. 실행화면

텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

1. 고찰

#include<time.h>의 srand(time(NULL))을 추가하여 완전 랜던 값을 생성하였습니다.