**LinkedList\_Stack**

5645640 제갈건 0509

1. 실행코드

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

//top, size, full, empty, swap, printf

#define max\_size 5

typedef struct ListStack{

int data;

struct ListStack\* under;

} Stack;

bool IsEmptyStack(Stack\* Top){

if (Top == NULL) {

printf("Stack is Empty\n");

return true;

}

return false;

}

bool isFullStack(Stack\* Top) {

int size = 0;

if (Top == NULL) return false;

while (Top != NULL && size <= max\_size){

size++;

Top = Top->under;

}

if (size >= max\_size) {

printf("Stack is Full\n");

return true;

}

return false;

}

void printStack(Stack\* Top) {

if (IsEmptyStack(Top)) return;

printf("Stack :TOP");

do{

printf("->|%d|", Top->data);

Top = Top->under;

} while (Top != NULL);

printf("\n");

if (isFullStack(Top)) return;

}

void push(Stack\*\* Top, int data){

Stack\* newNode = NULL;

newNode = (Stack\*)malloc(sizeof(Stack));

if (isFullStack(\*Top)) return;

//리스트 맨 앞에 새로운 노드를 추가합니다.

newNode->data = data;

newNode->under = \*Top;

\*Top = newNode;

printf("%d pushed to stack\n", data);

}

int pop(Stack\*\* Top){

Stack\* temp = NULL;

int data = 0;

if (IsEmptyStack(\*Top)) return INT\_MIN;

else

{ //Top 위치(리스트 맨 앞)에 있는 노드를 제거합니다

temp = \*Top;

data = (\*Top)->data;

\*Top = (\*Top)->under;

free(temp); // 노드 제거

}

return data;

}

int peek(Stack\* Top) {//peek함수

if (IsEmptyStack(Top)) return INT\_MIN;

printf("\npeek\_top: %d\n", Top->data);

return Top->data;

}

int size(Stack\* Top) { // 현재 스택의 개수 리턴

if (isFullStack(Top)) return max\_size;

int size = 0;

while (Top != NULL) {

size++;

Top = Top->under;

}

return size;

}

int main(void)

{

Stack\* S = NULL;

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

int Stack\_Data1 = peek(S);

printf("size: %d\n\n", size(S));

printStack(S);

printf("\n%d poped from stack\n", pop(&S));

int Stack\_Data2 = peek(S);

printf("size: %d\n\n", size(S));

printStack(S);

return 0;

}

1. 실행화면

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

1. 고찰

스택의 최대 사이즈는 #define max\_size 5를 이용하여 구현하였고, 현재 스택의 크기를 확인하는 것은 size()함수를 이용하였습니다.