**[201533661 이승수’s Data Structure Lab1-1]**

**[Lab1-1]**

**Code)**

#include <stdio.h>

#include <stdlib.h>

/\*Global Variables & define\*/

#define size 10 //stack size

int stack[size]={0,};

int top=-1;

//int rear=-1;

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

//description: check if stack is full

//input: none

//output: boolean result if stack is full(int)

int stack\_full(){

if(9<=top){//if stack is full, return true(1)

return 1;

}else{//else, return false(0)

return 0;

}

return 1;

}

//description: check if stack is empty

//input: none

//output: boolean result if stack is full(int)

int stack\_empty(){

if(top<0){//if stack is empty, return true(1)

return 1;

}else{//else, return false(0)

return 0;

}

}

//description: append input data to the top of stack

//input: data to append(int)

//output: none

void push(int num){

if(!stack\_full()){

top++;

stack[top]=num;

}else{

printf("Stack is Full!!");

}

}

//description: removes data at top of stack

//input: none

//output: removed data value at top of stack(int)

int pop(){

if(!stack\_empty()){

int atTop=stack[top];

stack[top]=0;

top--;

return atTop;

}else{

printf("Stack is Empty!!");

return 0;

}

}

//description: print numbers in stack in a row

//input: none

//output: data values in size of stack

void printStack(){

int i;

if(stack\_empty()){

printf("NULL\n");

}

else{

for(i=0; i<=top; i++){

printf("%d ->",stack[i]);

}

printf("\n");

}

}

//description: test stack functions until stack size be full

//input: none

//output: none

void main(){

printf("Initial Stack: ");

printStack();//result: NULL

while(!stack\_full()){

int tmp=0;

printf("put number: ");

scanf("%d",&tmp);

printf("\n");

if(0<tmp&&tmp<10){//push tmp into stack if tmp between 1 to 9

push(tmp);

}else{//pop at top from stack if tmp is out of the range

pop();

}

printStack();

}

printf("Stack size is full, end program.");

getchar();

}

**Result)**

