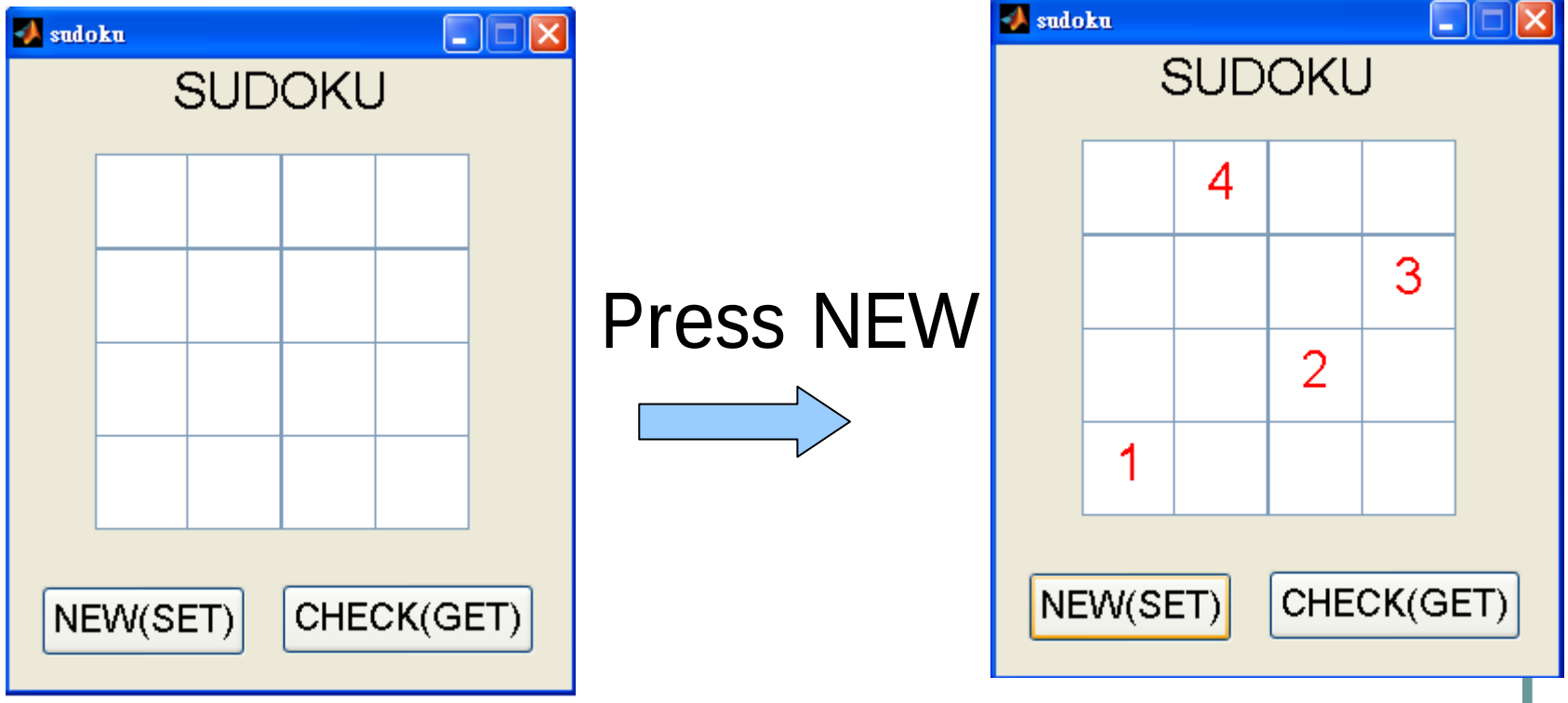
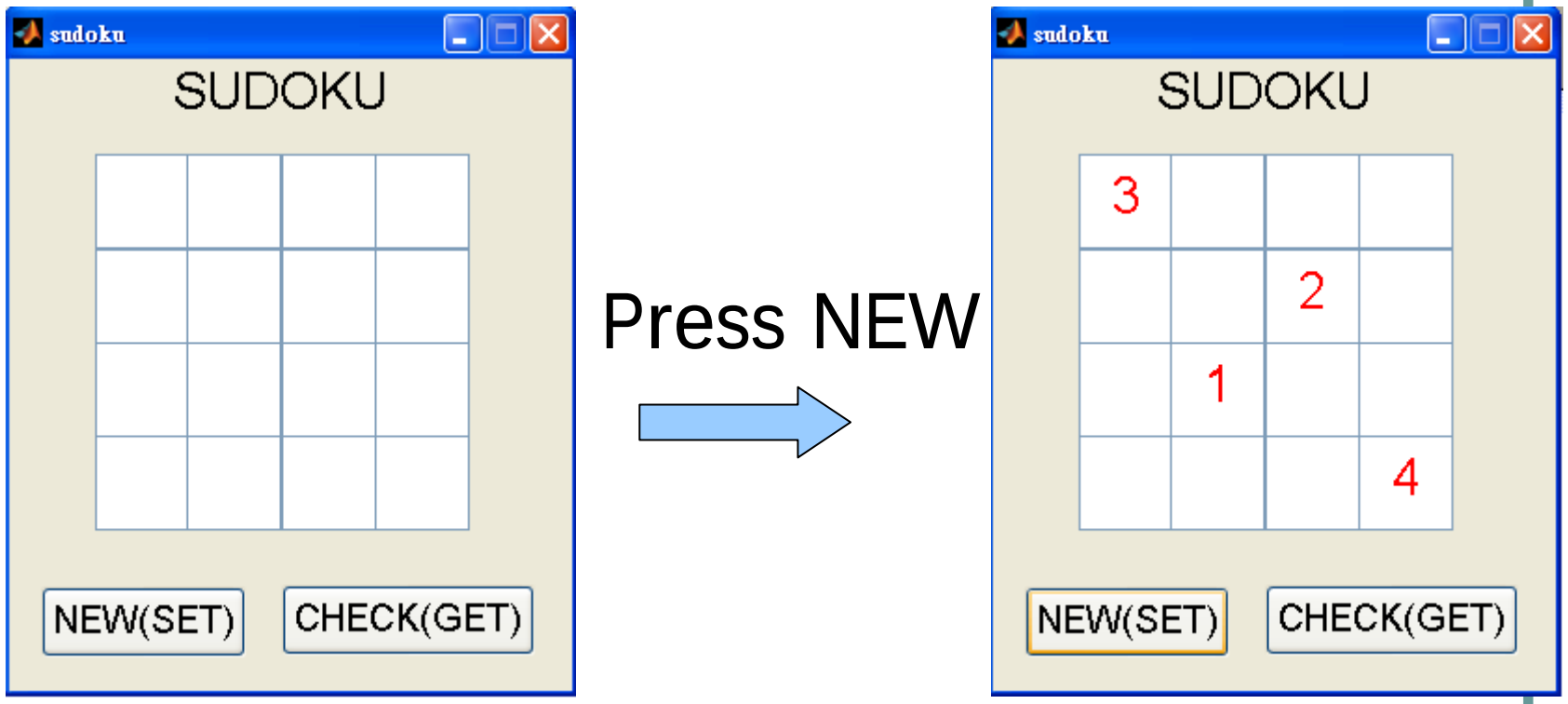
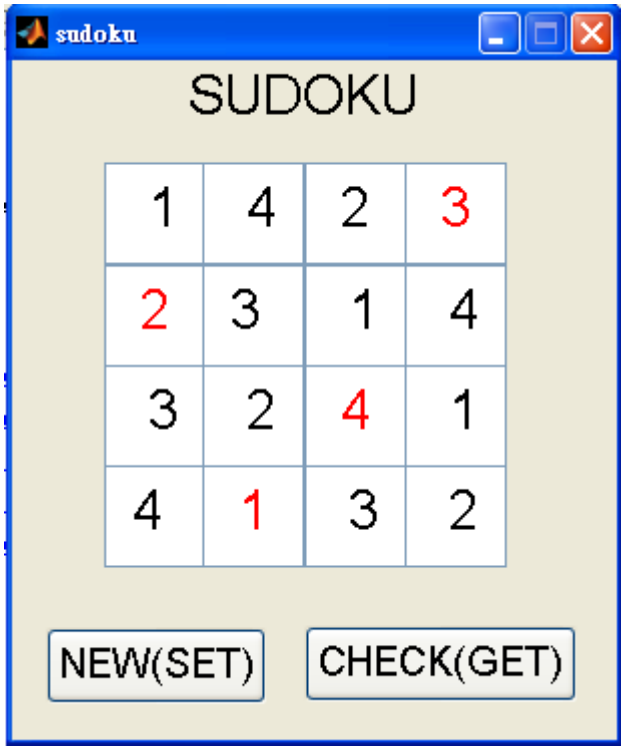
Lecture 8 Stack

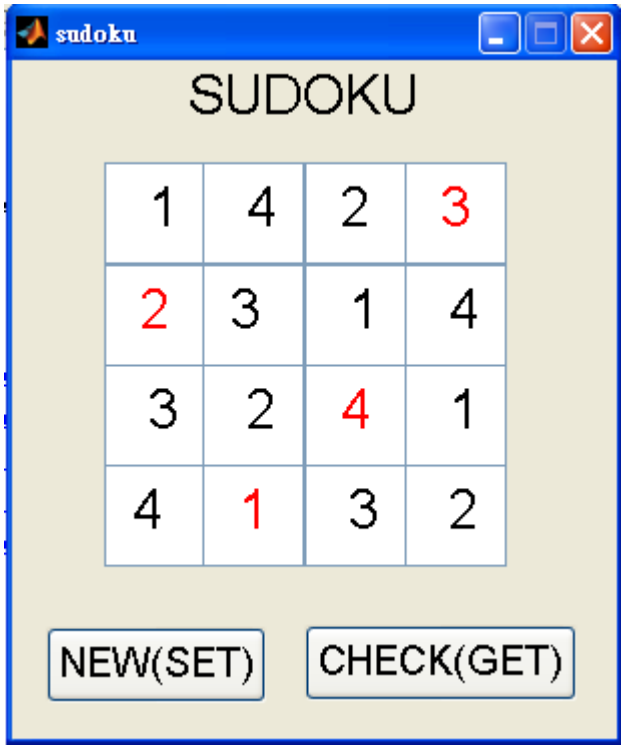
- Stack
  - New
  - Pop
  - Push


sudoku
SUDOKU
NEW(SET)
CHECK(GET)
Press NEW
sudoku
SUDOKU
4
3
2
1
NEW(SET)
CHECK(GET)


sudoku
SUDOKU
NEW(SET)
CHECK(GET)
Press NEW
sudoku
SUDOKU
3
2
1
4
NEW(SET)
CHECK(GET)

PLAY



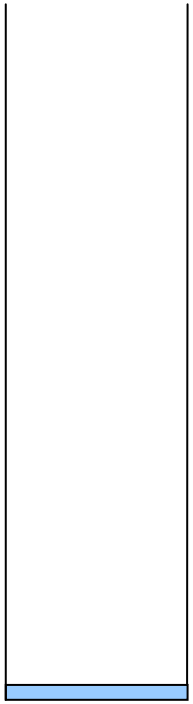


CHECK

Check row : 0 error
Check column : 0 error
Check block : 0 error

Data Stack

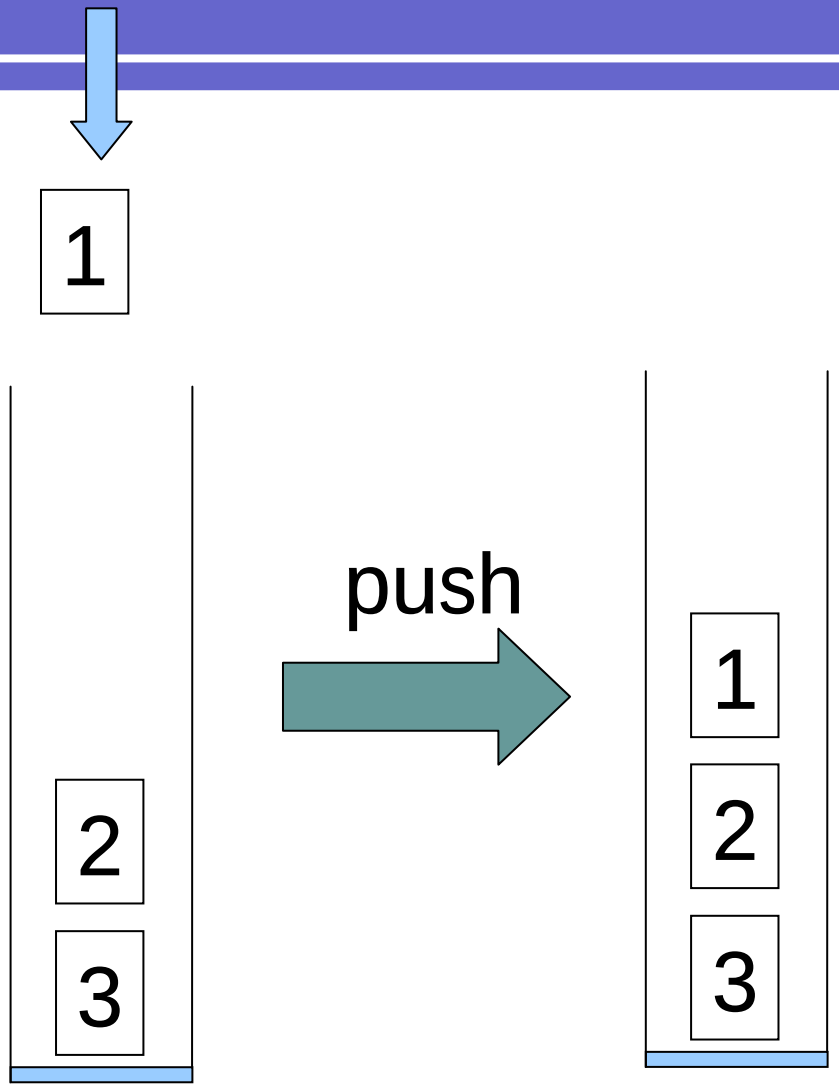
- Data structure: First in last out
- Operations
  - Initialize
  - Push
  - Pop



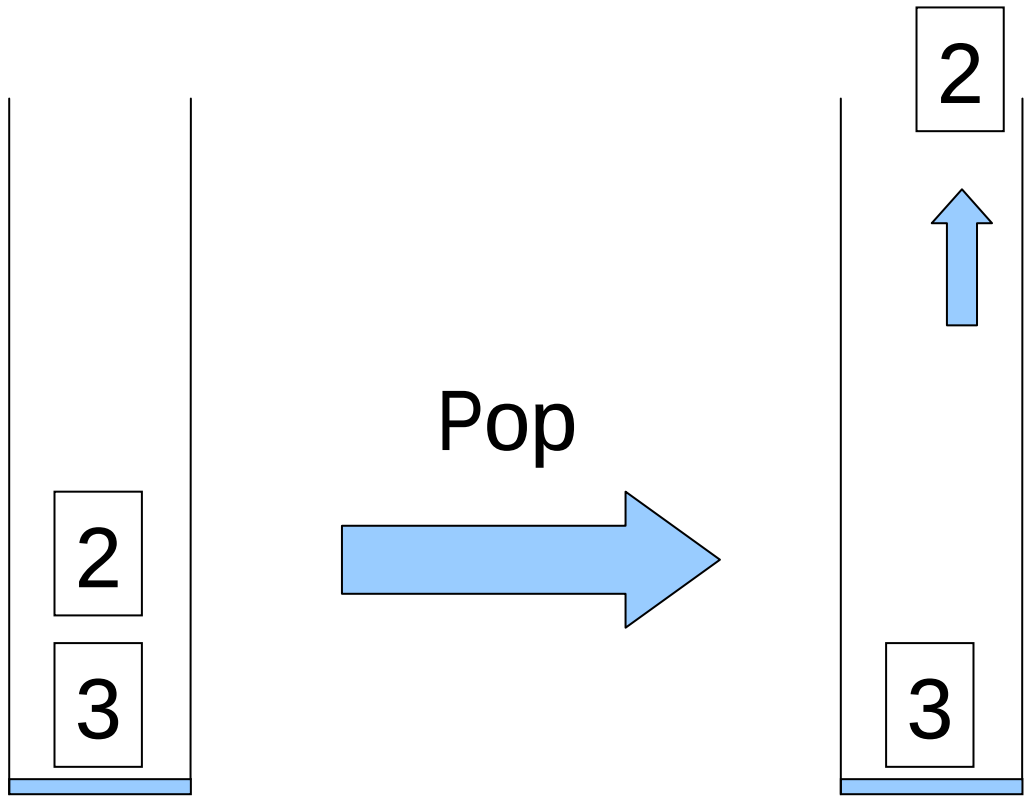
empty

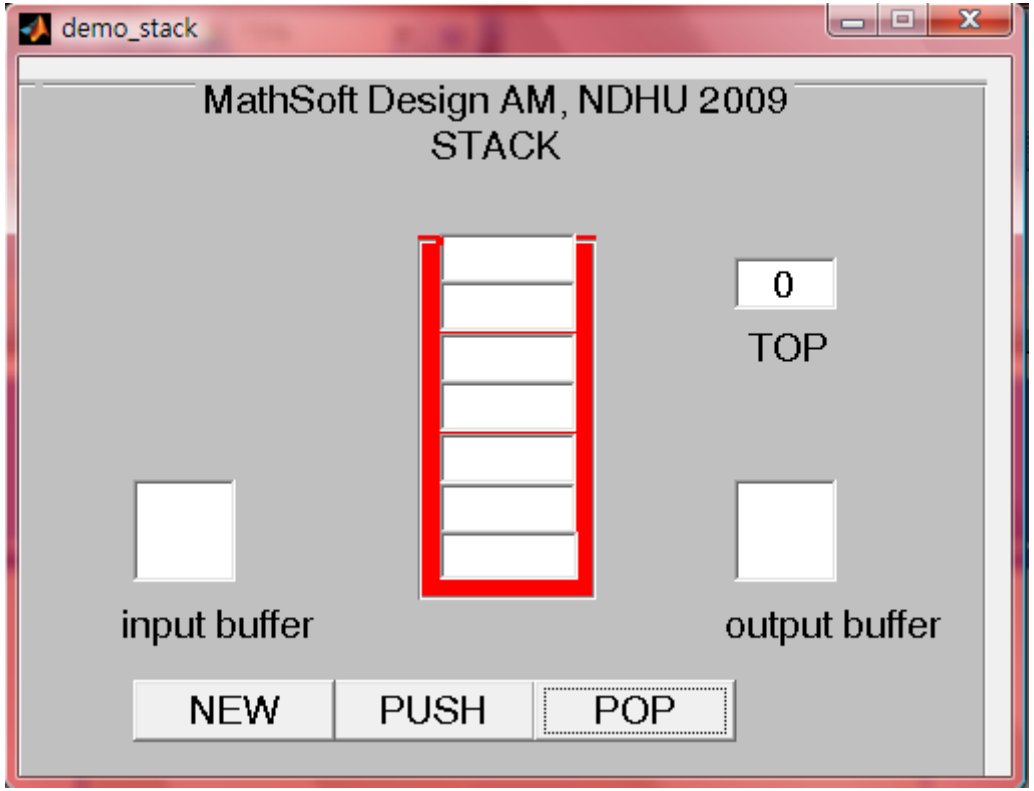
Structure

- Stack
  - String
- Push



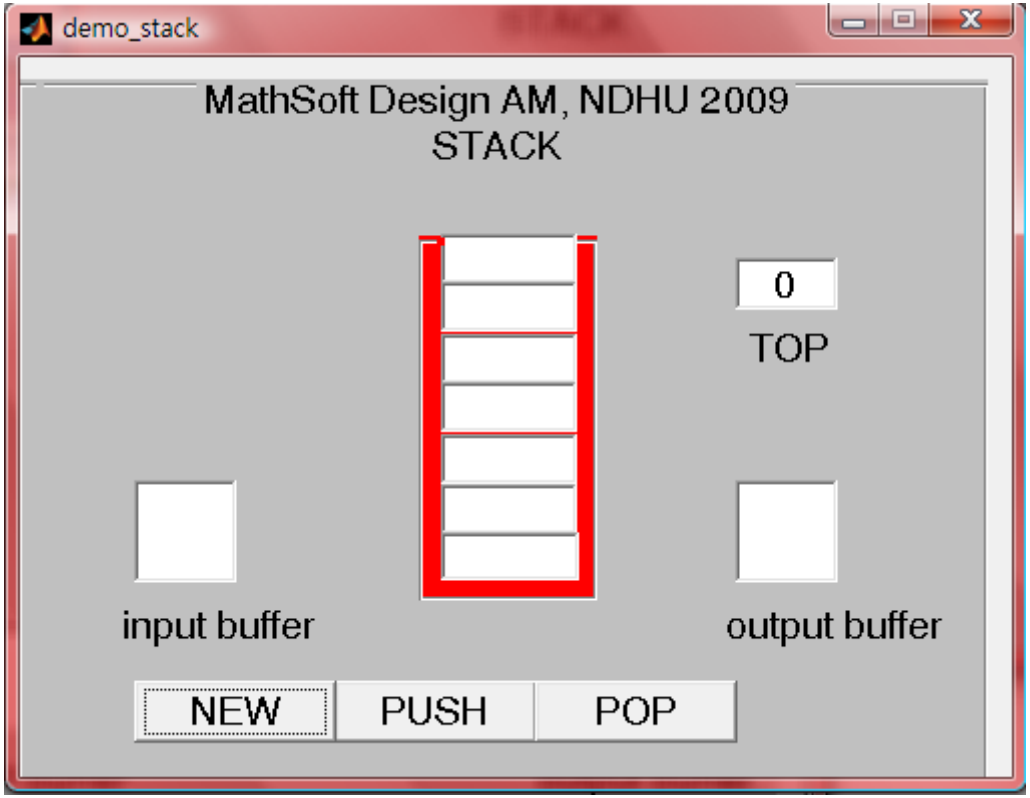
Pop




demo_stack
MathSoft Design AM, NDHU 2009
STACK
0
TOP
input buffer
output buffer
NEW
PUSH
POP

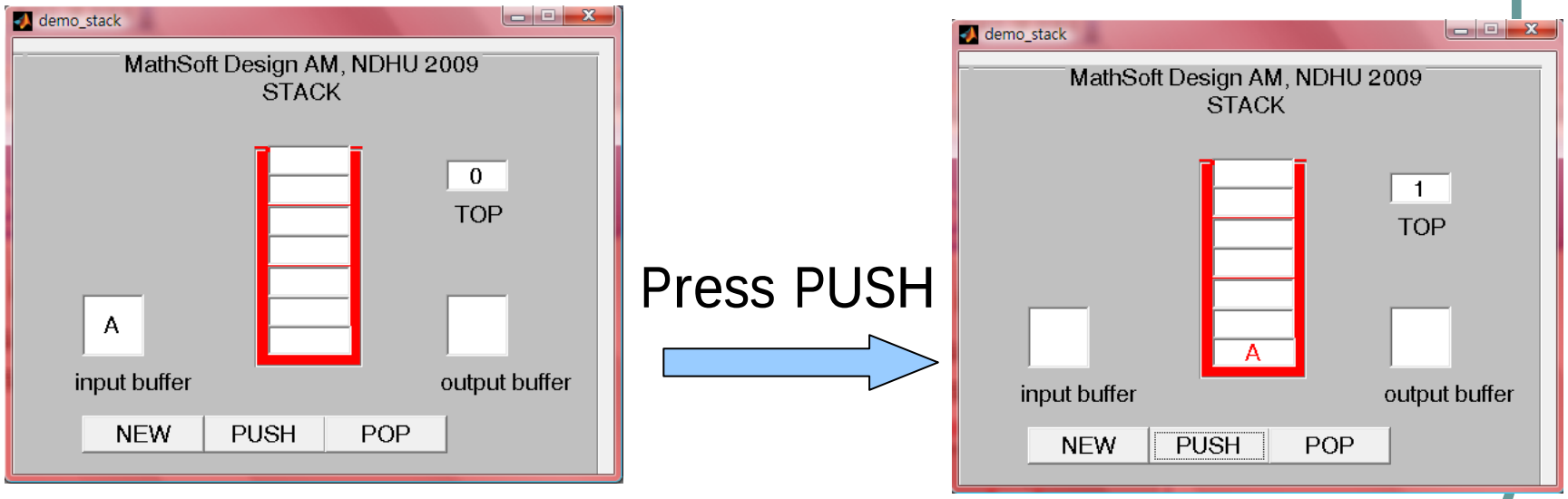
NEW

- Activate PUSH and POP buttons



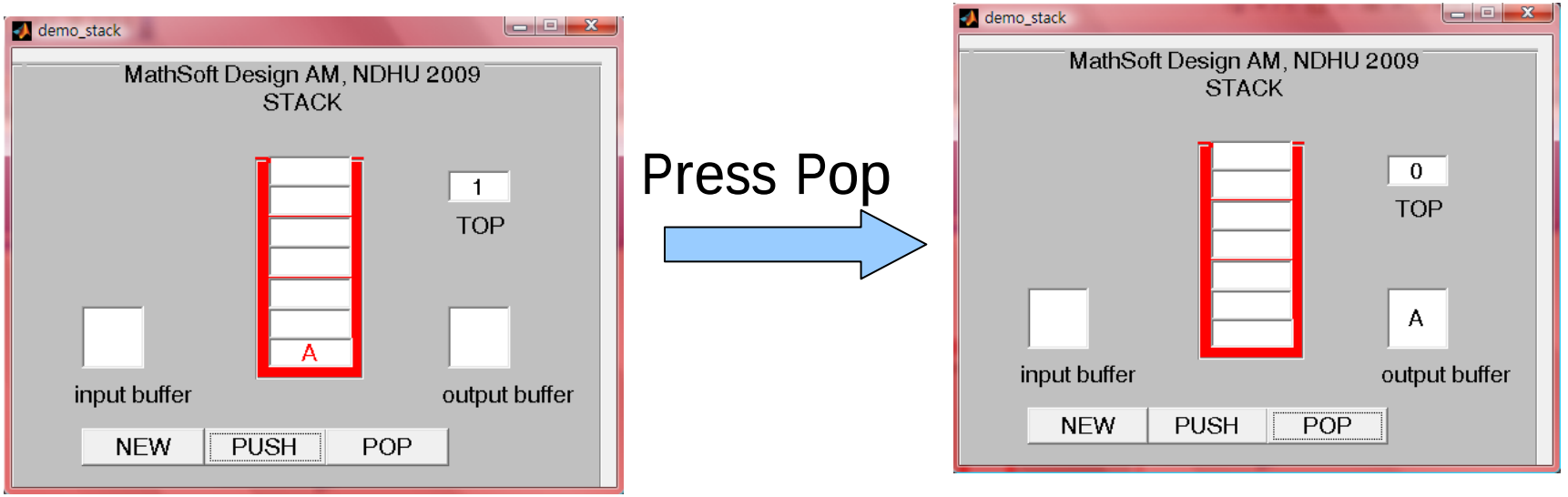
PUSH

- Key in a character to the input buffer
- Press PUSH button to store the character to the stack



POP

- Get the top element from the stack
- Put to the output buffer


demo_stack
MathSoft Design AM, NDHU 2009
STACK
1
TOP
A
input buffer
output buffer
NEW
PUSH
POP
Press Pop
demo_stack
MathSoft Design AM, NDHU 2009
STACK
0
TOP
A
input buffer
output buffer
NEW
PUSH
POP

Stack initialization

1. Get a positive integer n that denotes the maximal size of a stack
2. Set stk.S to an empty string
3. Set stk.n to n
4. Set stk.top to zero

Push

1. Let stk denote a stack, and c denote an object
2. Append the object to the string that emulates a stack
   A. If stk.top is less than stk.n, increase stk.top by one and replace stk.S(stk.top) with c
3. Return stk

```
function stk=stack_push(stk,c)
    if stk.top == stk.n
        return
    else
        stk.top=stk.top + 1;
        stk.S(stk.top)=c;
    end
return
```

Pop

1. Input a stack, stk
2. If stk.top is zero, output an error message and exit
3. Set c to the last element of stk, replace the last element of stk with a blank and decrease stk.top by one
4. Return c and stk

```
function [c,stk]=stack_pop(stk)
    if stk.top==0
        c='';
        return
    end
    c=stk.S(stk.top);
    stk.S(stk.top)=' ';
    stk.top=stk.top-1;
return
```

Demo_stack

demo_stack.m

demo_stack.fig