03. Stack

dbserver.korea@gmail.com

Agenda

- Instruction
- Stack Problems
    - Stack ADT
    - Brace check program

Instruction

All programs are written with C.

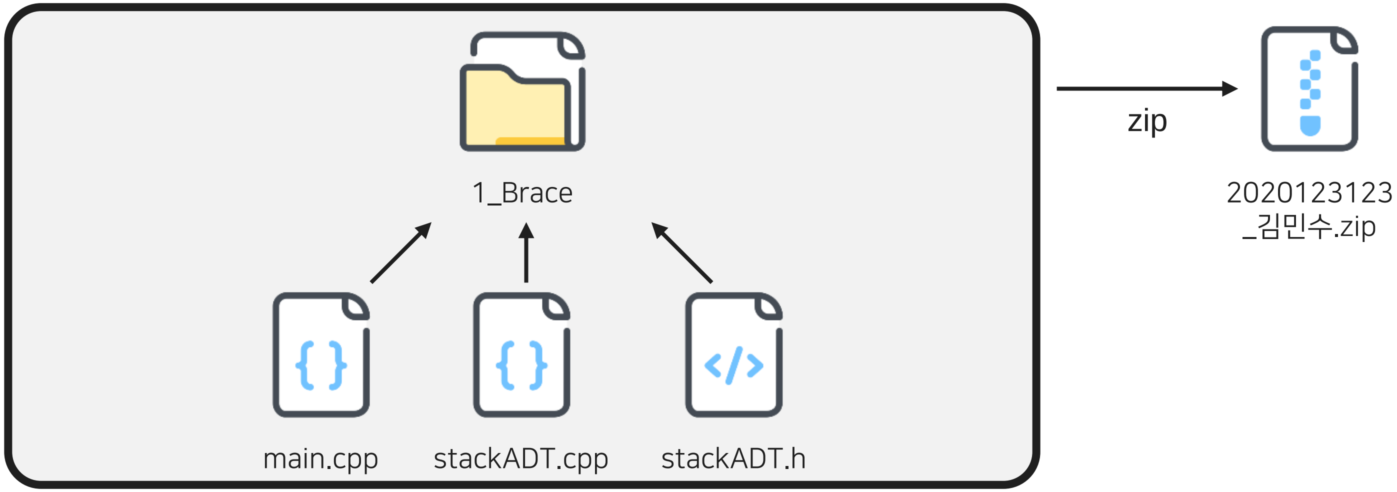
Please submit your works on Blackboard.

Archive your code and header files into a ZIP file and submit the ZIP file.

- Do not submit a whole project.
- Do not include executable files (.exe) or debug folders.

Make sure that your ZIP file follows a naming convention.

- (student number)_(name).zip -> E.g. 2020123123_김민수.zip



Stack ADT

Implement a stack ADT

Functions:

- CreateStack: allocate memory and initialize.
- Push
- Pop
- Stack Top: return top value of the stack.

  Do not pop the top value.
- DestroyStack: remove all items and deallocate memory.
- CatStack: concatenate two stacks.

  Details are the next page.

You may implement additional functions for your convenience.

Stack ADT (cont)

Make your stackADT hold a single character type value.

Note for CatStack

- Pay attention to the order of the result of CatStack function.
- After concatenating, you should delete the Stack B.

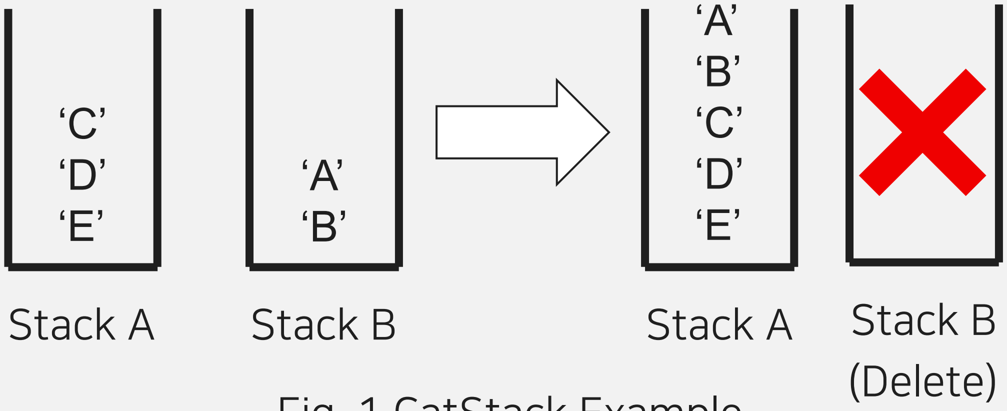


Fig. 1 CatStack Example

Folder> 1_Brace

Brace Check Program

Using your Stack ADT, implement a function that checks if a string has correct brace pairs(i.e. (, {, [) .

- If the source code has correct pairs, print ‘yes’.
- If the function find a wrong brace, print ‘no’.
- You have to check all kinds of brace ‘( )’, ‘{ }’, ‘[ ]’.

Input : a single line of characters.

Output: braces are paired or not

```
So when I die (the [first] I will see in (heaven) is a score list).
[ first in ] ( first out ).
Half Moon tonight (At least it is better than no Moon at all].
A rope may form )( a trail in a maze.
Help( I[m being held prisoner in a fortune cookie factory)].
([ (([( [ ] ) ( ) (( ))] )) ]).
 .
```

Inputs

```
yes
yes
no
no
no
yes
yes
```

Outputs

E.g. 02 Example of Brace Check Program