

Arraystack

Testing array implementation

Commands:

H : Help (displays this message)

+x : Push x

- : Pop

C : Clear

E : Empty stack?

F : Full stack?

Q : Quit the test program

Empty stack.

Command: e

Stack is empty

Empty stack.

Command: f

Stack is NOT full

Empty stack.

Command: +a

Push a

Top = 0

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

[a]

Command: +b

Push b

Top = 1

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

a	[b]						
---	-----	--	--	--	--	--	--

Command: +c

Push c

Top = 2

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

a b [c]

Command: +d

Push d

Top = 3

0	1	2	3	4	5	6	7
a	b	c	[d]				

Command: +e

Push e

Top = 4

0	1	2	3	4	5	6	7
a	b	c	d	[e]			

Command: +f

Push f

Top = 5

0	1	2	3	4	5	6	7
a	b	c	d	e	[f]		

Command: +g

Push g

Top = 6

0	1	2	3	4	5	6	7
a	b	c	d	e	f	[g]	

Command: +h

Push h

Top = 7

0	1	2	3	4	5	6	7
a	b	c	d	e	f	g	[h]

Command: +i

Push i

Error: Stack full, cannot push

Top = 7

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

a	b	c	d	e	f	g	[h]
---	---	---	---	---	---	---	-----

Command: -

Popped h

Top = 6

0	1	2	3	4	5	6	7
a	b	c	d	e	f	[g]	

Command: +h

Push h

Top = 7

0	1	2	3	4	5	6	7
a	b	c	d	e	f	g	[h]

Command: e

Stack is NOT empty

Top = 7

0	1	2	3	4	5	6	7
a	b	c	d	e	f	g	[h]

Command: f

Stack is full

Top = 7

0	1	2	3	4	5	6	7
a	b	c	d	e	f	g	[h]

Command: c

Clear the stack

Empty stack.

Command: -

Popped Error: Empty stack, cannot pop

Empty stack.

Command: q

Nodestack

Testing linked implementation

Commands:

H : Help (displays this message)

+x : Push x

- : Pop

C : Clear

E : Empty stack?

F : Full stack?

Q : Quit the test program

Empty stack

Command: e

Stack is empty

Empty stack

Command: f

Stack is NOT full

Empty stack

Command: +a

Push a

Top [a] Bottom

Command: +b

Push b

Top [b] a Bottom

Command: +c

Push c

Top [c] b a Bottom

Command: +d

Push d

Top [d] c b a Bottom

Command: -

Popped d

Top [c] b a Bottom

Command: e

Stack is NOT empty

Top [c] b a Bottom

Command: f

Stack is NOT full

Top [c] b a Bottom

Command: c

Clear the stack

Empty stack

Command: q

Postfix Tests

******tested expressions in **bold**

*******test cases are hard coded

Tested expressions:

- 5
- **3 4 + 5 2 / ***
 - $((3+4)(5/2))^2 = 306.25$
- 1 +
- 5 5
- 5 0 /

Running test cases...

Operation 5 = 5

Operation **3 4 + 5 2 / * 2 ^** = 306.25

Operation 1 + = Error: Insufficient operands

Operation 5 5 = Error: Incomplete expression

Operation 5 0 / = Error: Division by zero

Testing the Array implementation

Enter expression: 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5

Error: Stack full, cannot process further

Please enter expression again

Delimiter Tests

****test expressions are hard coded**

Tested expressions:

- `[{}]`
- `(f(b)-(c+d))/2`
- `(f(b)-(c+2))/2`
- `(`
- `{()}()[]`

This program checks for properly matched delimiters.

Running test cases...

`[{}]` | Invalid

`(f(b)-(c+d))/2` | Valid

`(f(b)-(c+2))/2` | Invalid

`(` | Invalid

`{()}()[]` | Valid

Enter delimited expression (<EOF> to quit) :

Testing the Array implementation

`((((((((((((((((((((((((((((((((((((((((((((`

Stack full, cannot process further

Invalid

Enter delimited expression (<EOF> to quit) :