

חסן N

30 פנאי על הקרבים

LIFO

last in, first Out

פולחן

- אחרון לחסן - כקה
- אחרון על החסן

- האם החסן כקה ?

- האם החסן - אלה ?

- אלה על הלבן

- מי הוא החסן
(מי נכנס אחרון)

- ערבון

- הכנסה

- הוצאה - על רקע האחרון

הפונקציות העוזרות
שהוכנסו

Stack operations
stack S;

S.Push(x) הכנסה

a = S.Pop() הוצאה וזיכרון a

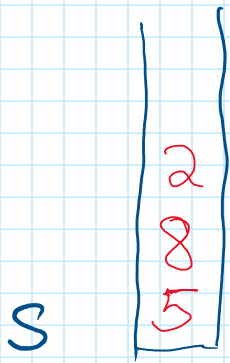
a = S.Top() a מקבל את
הערך של
האש המוחסן
מבלי לשנות את המבנה.

S.isEmpty() מחזיר T
אם S ריקה

S.isFull() מחזיר T
אם S מלאה

מחזיר את המוחסן — Pop

מחזיר את המוחסן — Top
כמו שהיא



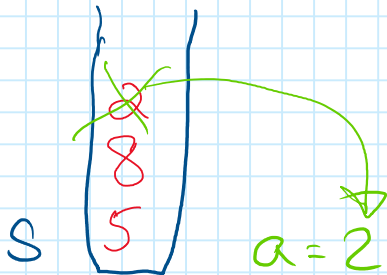
Stack S

S.push(5)

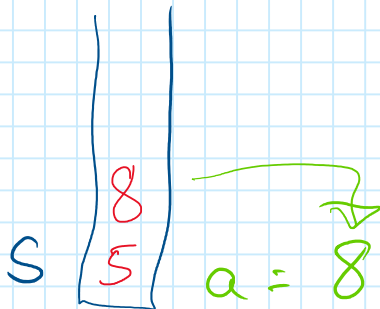
S.push(8)

S.push(2)

a = S.pop()



a = S.Top()



ה-ⁿ : מה הערך החמישי?

אכאל המחסני?

- לא אכפח לי מה קרה

למחסני.

for (i=0; i<~~5~~ⁿ; i++)

$a = S.pop();$

// מכיל את הערך התמישי

for ($i=0; i < \overset{n-1}{4}; i++$)

$S.pop();$

$a = S.pop();$ // OR $a = S.Top()$

בעיה: מה הערך ה-n במסגון

תנאי: לא רשוי להסיר את המסגון

\Leftarrow בסוף המסגון תראה
כל מה שהתחלה.

Stack $S2;$

for ($i=0; i < n-1; i++$)

{

$a = S.pop();$
 $S2.push(a);$

}

$answer = S.Top()$

for ($i=0; i < n-1; i++$)

{

$a = S2.pop();$

$S.push(a)$



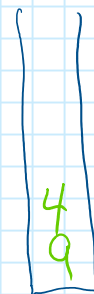
S

$n=3$

$i=0$

$a=4$

$answer = 1$



$S2$

| 9 |

```

{
    S.push(a)
}

for (i=0; i < n-1; i++)
    S.push(S2.pop())
    
```

answer = 1

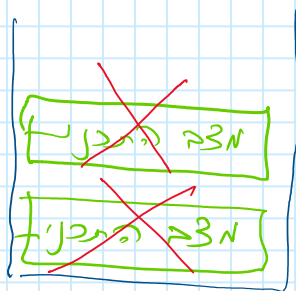
9
4
1
2
8
5
S

S2

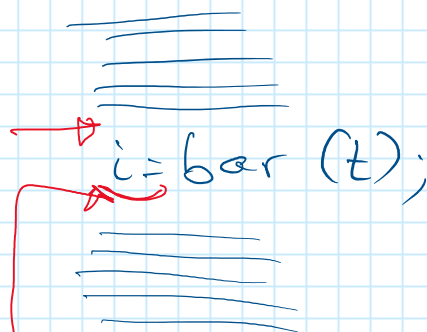
n = 3
 i = 0
 a = 4 9



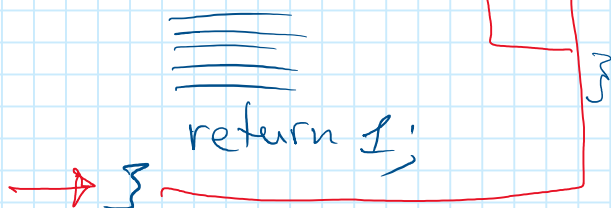
! לא נכנס - נכנס
 נכנס -
 - לא נכנס -
 - לא נכנס -



foo(int x)



bar(int x)



$S.push(S2.pop())$

X ((← נכשל בחינה 1
 X)) ← נכשל בחינה 1, 2
 X)) ((← נכשל בחינה 2
 ✓ ((((()))))
 ↑ ↓

1. אולי מספר של פותחים וסוגרים

2. פותר מופיע לפני סוגר

היציף של n סגורים n...1
 קבל היציף k...1
 $Num('(') \geq Num(')')$ ← diff

ההפרש בין מספר הפותחים למספר הסוגרים
 $diff = 0$

while there are tokens
 ⚡
 // get next token
 if (token == '(')
 diff++;
 else if (token == ')')
 diff--;
 if (diff < 0)

diff
 0
 1
 2
 1
 2
 1
 2
 3
 0
 1
 2
 3

if (diff < 0)
 // error. not balanced

}

if (diff > 0)
 // error. not balanced

(()) (
 ↑ ↑ ↑ ↑ ↑

diff

0

1

2

1

0

1

(() () (()) ())

(()) (

)) ((

Stack S;

while there are token

{

get next token

if (token == '(')

S.push(token);

else if (token == ')')

{

if (S.isEmpty())

// error. unmatched ')'

else

S.pop();

}

```

    }
    → }
    if (!s.isEmpty())
        // error. not enough ')'

```

סוגי תווים פתוחים וסוגי תווים סגורים

(([] { [] }) { })

→ ([)]

diff round
diff square
diff squiggly

$$3 + 5 * 2 + 6 * 3$$

8
16
22
66

מספר
ל
במספר
3N
כגודל
מספר

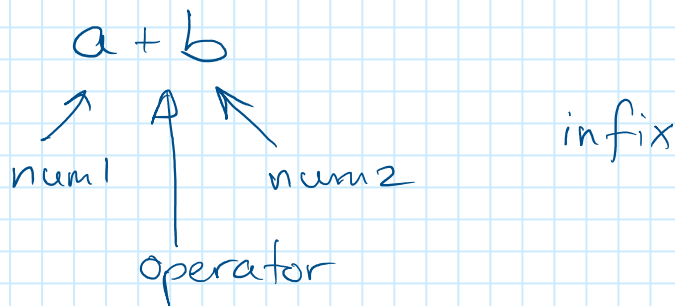
$$3 + 5 * 2 + 6 * 3$$

10
13
31

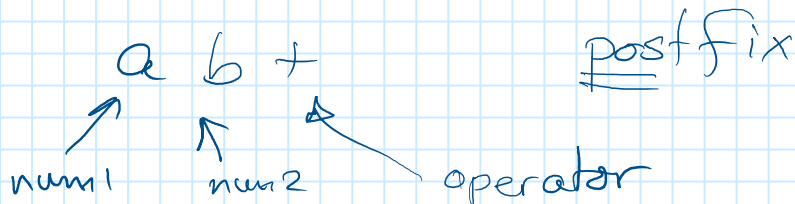
מספר
במספר
שני

$$a = x + y * 2 + w * t$$

RPN - reverse Polish notation



RPN



$$3 + 5 * 2 + 6 * 3$$

Below the expression, intermediate results are shown in red: $5 * 2$ and $6 * 3$. A red box contains the sequence of tokens: $3 \ 5 * 2 + 6 * 3 +$.

Stack S;

while there are tokens

{

get next token

if (isNumber(token))

S.push(token);

→ else if (isOperator(token))

{

num1 = S.pop();

num2 = S.pop();

// need to check
!S.isEmpty()

```

num2 = s.pop();
ans = calculate(num1, num2, token);
s.push(ans);
}
}

```

לפני כן נקרא s ונראה
הוא לא ריק

```
result = s.pop();
```

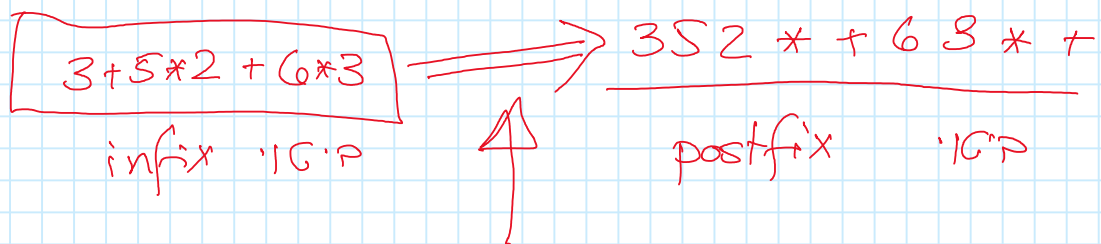
// הפסקה

```
if !s.isEmpty()
```

// error

Stack diagram showing elements: 3, 18, 3, 6, 13, 10, 2, 5, 3. The top element 3 is boxed.

num1	num2	token
2	5	*
10	3	+
3	6	*
18	13	+



הפסקה
הפסקה