

# ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

## Στοίβες

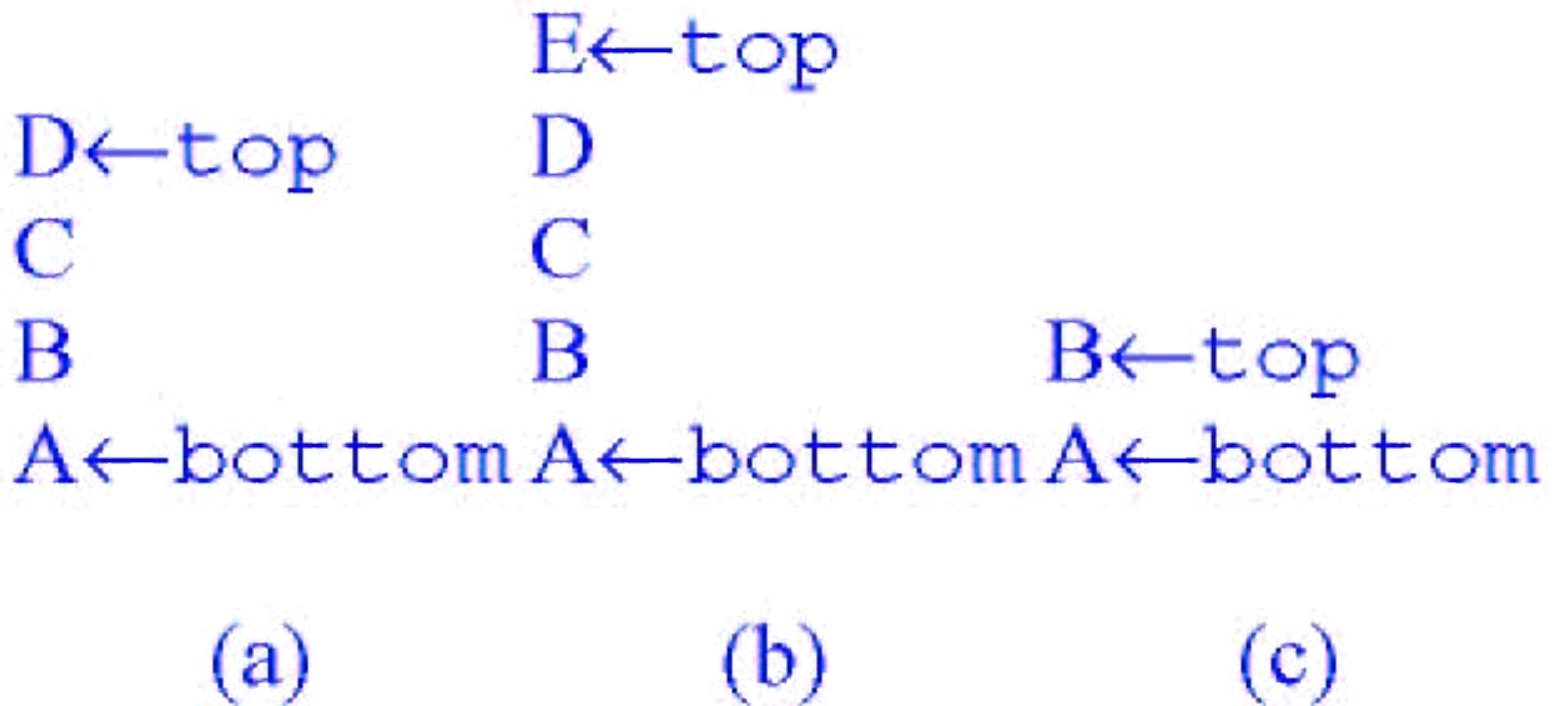


**Απόστολος Ν. Παπαδόπουλος**  
**Αναπληρωτής Καθηγητής**  
**Τμήμα Πληροφορικής Α.Π.Θ.**

# Βασικές Έννοιες

---

Η **στοίβα** (stack) αποτελεί μία δομή δεδομένων η οποία υποστηρίζει την ανάκτηση του τελευταίου στοιχείου που μπήκε: **Last-In First-Out (LIFO)**.



# Βασικές Έννοιες

---

```
AbstractDataType Stack {  
  instances  
    ordered list of elements; one end is  
    called the bottom; the other is the top;  
  operations  
    Create (): create an empty stack;  
    IsEmpty (): return true if stack is empty,  
        return false otherwise;  
    Top (): return top element of stack;  
    Push (x): add element x to the stack;  
    Pop (x): delete top element from stack  
        and put it in x;  
}
```

# Χρήσεις

---

Χρησιμοποιείται στις αναδρομικές κλήσεις για την αποθήκευση της προηγούμενης κατάστασης της συνάρτησης.

Χρησιμοποιείται στην εκτέλεση αλγορίθμων που απαιτούν back-tracking.

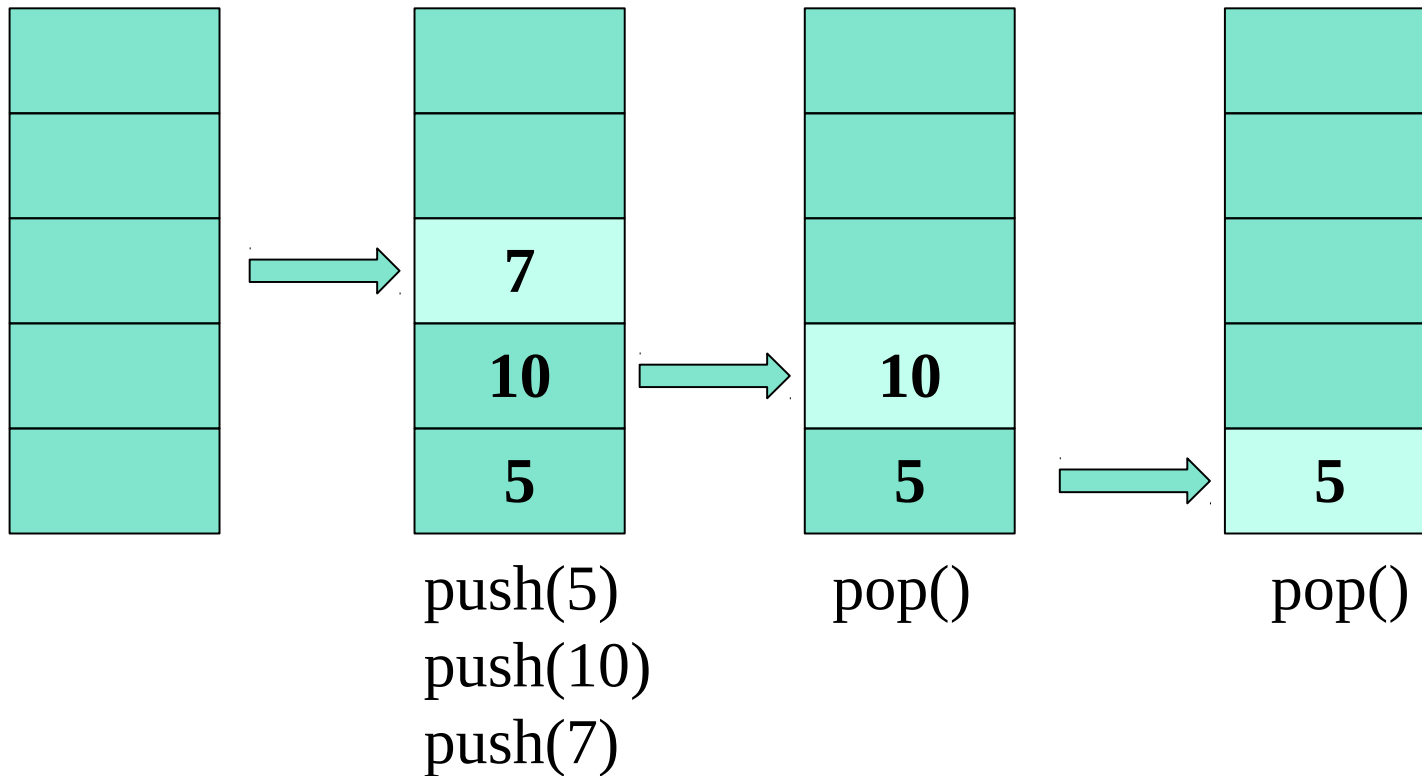
Χρησιμοποιείται για τη μετατροπή και υπολογισμό μαθηματικών εκφράσεων.

Χρησιμοποιείται για ταίριασμα παρενθέσεων σε μαθηματικές εκφράσεις.

κλπ

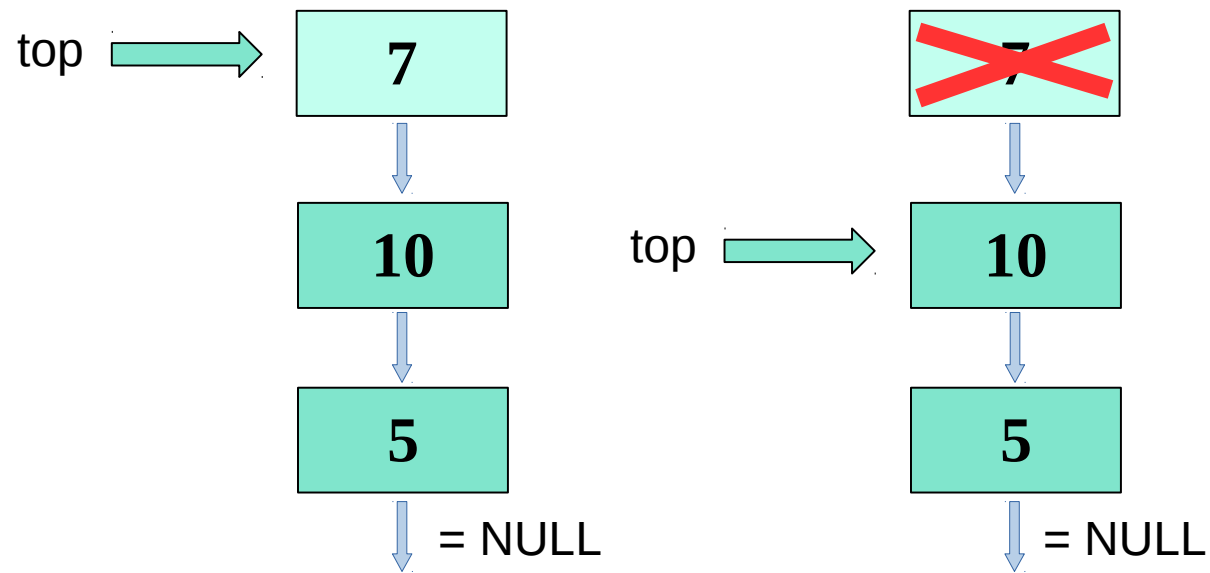
# Υλοποίηση με Πίνακα

## Παράδειγμα



# Υλοποίηση με Λίστα

---



# Υλοποίηση με Λίστα

---

```
struct Node {  
    int data;  
    struct Node* link;  
};
```

```
struct Node* top;
```

# Υλοποίηση με Λίστα

---

```
void Push(int data) {  
    struct Node* temp;  
    temp = new Node(); // Create new node temp and allocate memory  
  
    // Check if stack (heap) is full. Then inserting an element would lead to stack overflow  
    if (!temp) {  
        cout << "\nNot Enough Memory!";  
        exit(1);  
    }  
    temp->data = data; // Initialize data into temp data field  
    temp->link = top; // Put top pointer reference into temp link  
    top = temp; // Make temp as top of Stack  
}
```



# Υλοποίηση με Λίστα

---

```
void Pop() {  
    struct Node* temp;  
  
    // Check for stack underflow  
    if (top == NULL) {  
        cout << "\nStack Underflow" << endl;  
        exit(1);  
    }  
    else {  
        temp = top; // Top assign into temp  
        top = top → link; // Assign second node to top  
        temp->link = NULL; // Destroy connection between first and second  
        free(temp); // Release memory of top node  
    }  
}
```

# Εφαρμογή infix → postfix

---

Υπολογισμός της ΕΠΙΘΕΜΑΤΙΚΗΣ μορφής μίας έκφρασης από την ΕΝΔΟΘΕΜΑΤΙΚΗ μορφή.

ΕΠΙΘΕΜΑΤΙΚΗ (POSTFIX)

ΕΝΔΟΘΕΜΑΤΙΚΗ (INFIX)

# Εφαρμογή infix $\rightarrow$ postfix

---

Παράδειγμα:

Infix:  $A * B + C * D \wedge E / F + G * H$

Postfix:  $AB * CDE \wedge * F / + GH * +$

--	--

# Εφαρμογή infix → postfix

## Αλγόριθμος

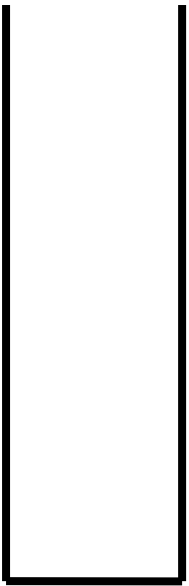
1. Αν ο χαρακτήρας εισόδου είναι μεταβλητή, τότε κάνε αντιγραφή στην έξοδο.
2. Αν ο χαρακτήρας εισόδου είναι αριστερή παρένθεση, τότε κάνε **push** την αριστερή παρένθεση στη στοίβα.
3. Αν ο χαρακτήρας εισόδου είναι δεξιά παρένθεση, συνέχισε να κάνεις **pop** από τη στοίβα και να αντιγράφεις στην έξοδο μέχρι να συναντήσεις την πρώτη αριστερή παρένθεση.
4. Αν ο χαρακτήρας εισόδου είναι ένας τελεστής  $T$  τότε:
  - Αν ο  $T$  έχει μεγαλύτερη προτεραιότητα από τον τελεστή που υπάρχει στην κορυφή της στοίβας (αν υπάρχει τελεστής) τότε κάνε **push( $T$ )**.
  - Διαφορετικά, κάνε διαδοχικά **pop** και αντιγραφή στην έξοδο, μέχρι στην κορυφή της στοίβας να υπάρχει τελεστής με χαμηλότερη προτεραιότητα από αυτήν του  $T$  ή να υπάρχει κάποιος άλλος χαρακτήρας που δεν είναι τελεστής.
5. Αν ο χαρακτήρας είναι το ; (ελληνικό ερωτηματικό) τότε είμαστε στο τέλος της αριθμητικής έκφρασης. Κάνε διαδοχικά **pop** και αντιγραφή στην έξοδο μέχρι να αδειάσει η στοίβα.

# Εφαρμογή infix $\rightarrow$ postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$

postfix expression:

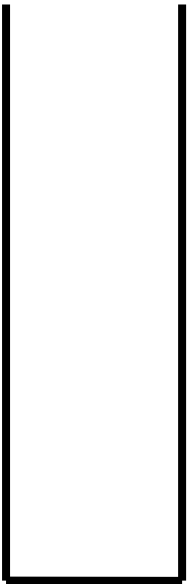


# Εφαρμογή infix $\rightarrow$ postfix

---

infix expression:  $\downarrow$  **z** = a \* ( x + y ) + z \* c ^ ( 2 - ( ~ d + w ) ) / x ;

postfix expression: **z**

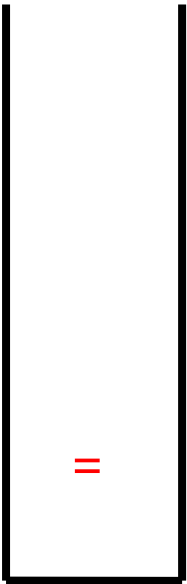


# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c \wedge (2 - (\sim d + w)) / x ;$

postfix expression:  $z a$



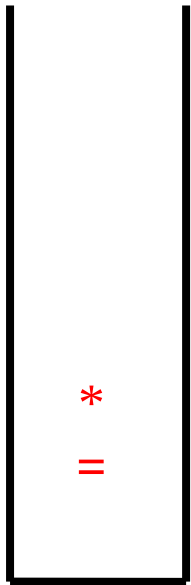


# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$

postfix expression:  $z a$

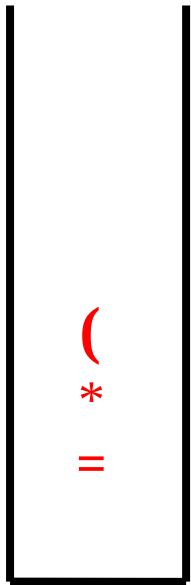


push ( '\*' )

# Εφαρμογή infix → postfix

infix expression: **z = a \* (** x + y ) + z \* c ^ ( 2 - ( ~ d + w ) ) / x ;

postfix expression: **z a**



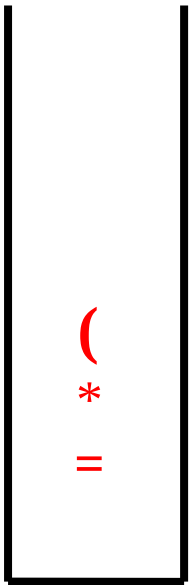
push ( ' ( ' )

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c \wedge (2 - (\sim d + w)) / x ;$

postfix expression:  $z a x$

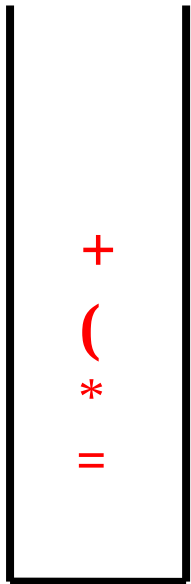


# Εφαρμογή infix → postfix



infix expression: **z = a \* ( x + y ) + z \* c ^ ( 2 - ( ~ d + w ) ) / x ;**

postfix expression: **z a x**

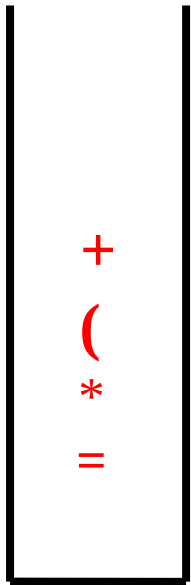


push ( '+' )

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y$

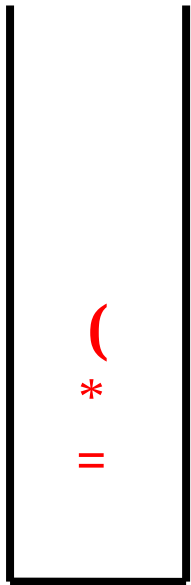


pop ( )

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c \wedge (2 - (\sim d + w)) / x ;$

postfix expression:  $z a x y +$



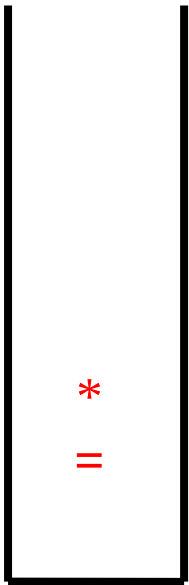
pop ( )

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c \wedge (2 - (\sim d + w)) / x ;$

postfix expression:  $z a x y +$



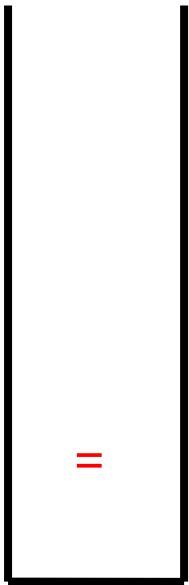
pop ( )

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + *$



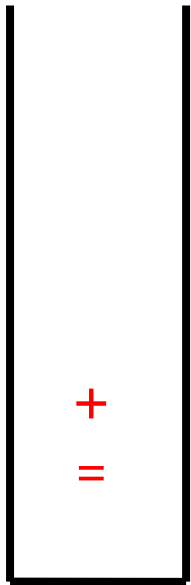


# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + *$



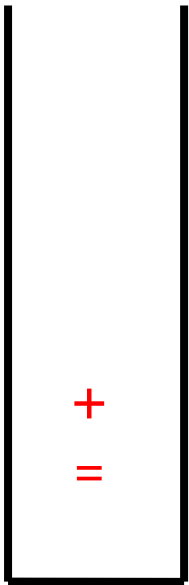
push ( '+' )

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c \wedge (2 - (\sim d + w)) / x ;$

postfix expression:  $z a x y + * z$

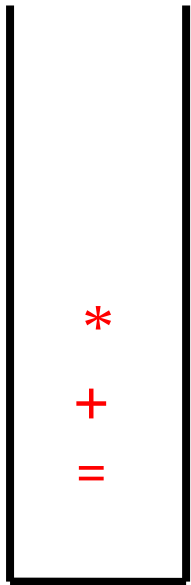


# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c \wedge (2 - (\sim d + w)) / x ;$

postfix expression:  $z a x y + * z$



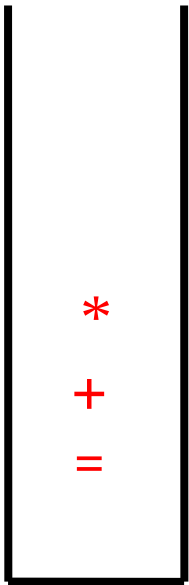
push ( '\*' )

# Εφαρμογή infix $\rightarrow$ postfix

---

infix expression:  $z = a * (x + y) + z * c \wedge (2 - (\sim d + w)) / x ;$

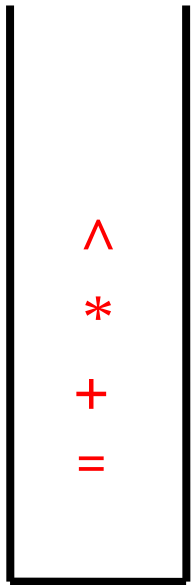
postfix expression:  $z a x y + * z c$



# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c \wedge (2 - (\sim d + w)) / x ;$

postfix expression:  $z a x y + * z c$



push ( ' ^ ' )

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c \wedge (2 - (\sim d + w)) / x ;$

postfix expression:  $z a x y + * z c$

(  
^  
\*  
+  
=

push ( ' ( ' )

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - (\sim d + w)) / x ;$

postfix expression:  $z a x y + * z c 2$

(  
^  
\*  
+  
=

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - (\sim d + w)) / x ;$

postfix expression:  $z a x y + * z c 2$

—  
(  
^  
\*  
+  
=

push ( ' - ' )



# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + * z c 2$

(  
-  
(  
^  
\*  
+  
=

push ( ' ( ' )

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + * z c 2$

~  
(  
-  
(  
^  
\*  
+  
=

push ( '~ ' )

Σημείωση: με ~ συμβολίζουμε το πρόσημο -

# Εφαρμογή infix $\rightarrow$ postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + * z c 2 d$

~  
(  
-  
(  
^  
\*  
+  
=

# Εφαρμογή infix $\rightarrow$ postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + * z c 2 d$

$\sim$   
(  
-  
(  
^  
\*  
+  
=

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + * z c 2 d \sim$

(  
-  
(  
^  
\*  
+  
=

pop ( )

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + * z c 2 d \sim$

+  
(  
-  
(  
^  
\*  
+  
=

push ( '+' )

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + * z c 2 d \sim w$

+  
(  
-  
(  
^  
\*  
+  
=

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$

postfix expression:  $z a x y + * z c 2 d \sim w$

+  
(  
-  
(  
^  
\*  
+  
=



# Εφαρμογή infix $\rightarrow$ postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + * z c 2 d \sim w +$

(  
-  
(  
^  
\*  
+  
=

pop ( )

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w ) ) / x ;$

postfix expression:  $z a x y + * z c 2 d \sim w +$

—  
(  
^  
\*  
+  
=

pop ( )

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$

postfix expression:  $z a x y + * z c 2 d \sim w +$

—  
(  
^  
\*  
+  
=

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$

postfix expression:  $z a x y + * z c 2 d \sim w + -$

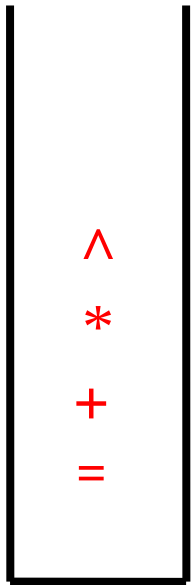
(  
^  
\*  
+  
=

pop ( )

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$


postfix expression:  $z a x y + * z c 2 d \sim w + -$



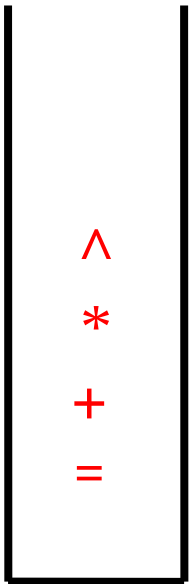
pop ( )

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$  

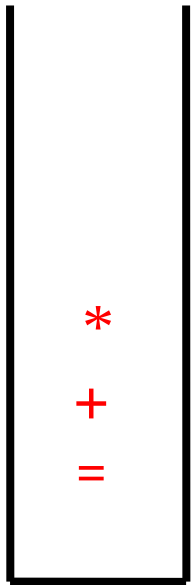
postfix expression:  $z a x y + * z c 2 d \sim w + -$



# Εφαρμογή infix → postfix


infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$  ↓

postfix expression:  $z a x y + * z c 2 d \sim w + - ^$

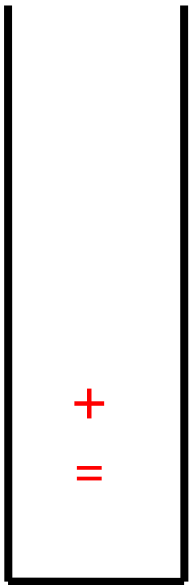


pop ( )

# Εφαρμογή infix $\rightarrow$ postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - (\sim d + w)) / x ;$  


postfix expression:  $z a x y + * z c 2 d \sim w + - ^ *$



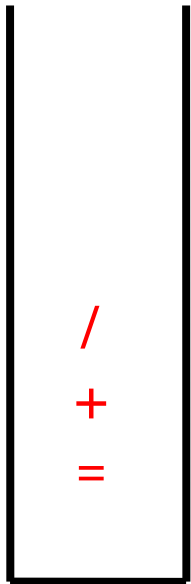
pop ( )



# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$  


postfix expression:  $z a x y + * z c 2 d \sim w + - ^ *$



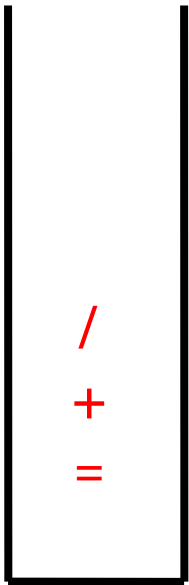
push ( '/' )

# Εφαρμογή infix $\rightarrow$ postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$  

postfix expression:  $z a x y + * z c 2 d \sim w + - ^ * x$

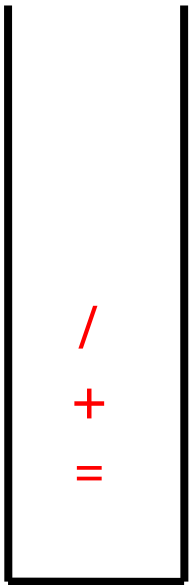


# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$  ↓

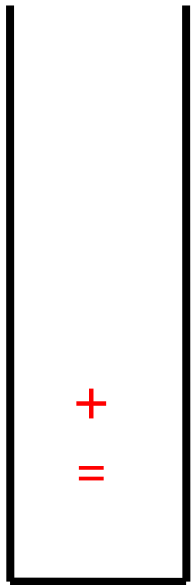
postfix expression:  $z a x y + * z c 2 d \sim w + - ^ * x$



# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$  ↓

postfix expression:  $z a x y + * z c 2 d \sim w + - ^ * x /$

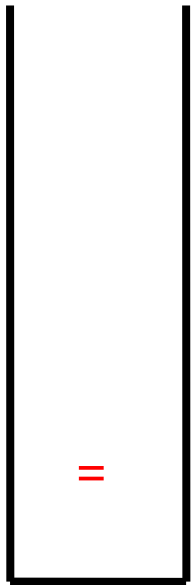


pop ( )

# Εφαρμογή infix → postfix

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$  ↓

postfix expression:  $z a x y + * z c 2 d \sim w + - ^ * x / +$



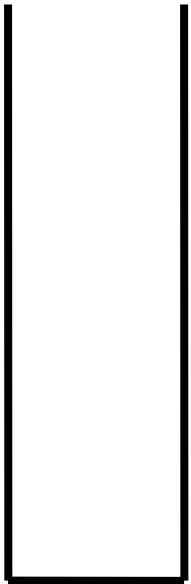
pop ( )

# Εφαρμογή infix → postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$  ↓

postfix expression:  $z a x y + * z c 2 d \sim w + - ^ * x / + =$



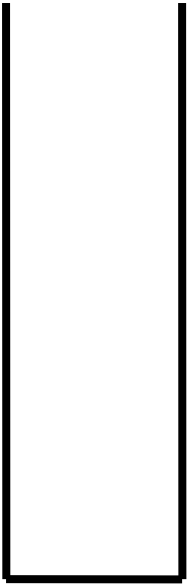
pop ( )

# Εφαρμογή infix $\rightarrow$ postfix

---

infix expression:  $z = a * (x + y) + z * c ^ (2 - ( \sim d + w )) / x ;$

postfix expression:  $z a x y + * z c 2 d \sim w + - ^ * x / + =$



**Παρατήρηση:** στη μεταθεματική μορφή  
δε χρειαζόμαστε παρενθέσεις!

# Υπολογισμός Έκφρασης

---

Αν δίνεται μία αριθμητική έκφραση σε postfix πως μπορούμε να υπολογίσουμε το αποτέλεσμα;

Π.χ. αν έχουμε την έκφραση

$AB*CDE^*F/+GH*+$

και  $A=2$ ,  $B=1$ , κλπ, πως μπορούμε να βρούμε το αποτέλεσμα με χρήση στοίβας;



# Άσκηση

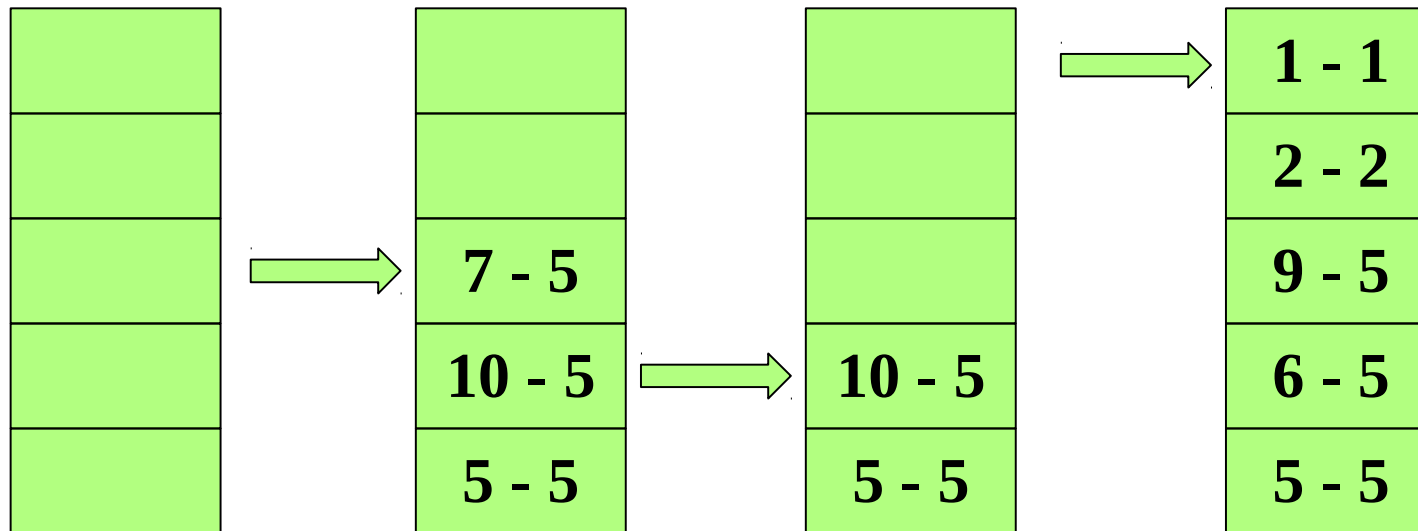
---

Να σχεδιάσετε δομή στοίβας η οποία εκτός από τις παραδοσιακές λειτουργίες **Push** και **Pop** να υποστηρίζει και τη λειτουργία **FindMin**. Όλες οι λειτουργίες πρέπει να εκτελούνται σε σταθερό χρόνο  **$O(1)$** .

*(Σημείωση: επιτρέπεται να χρησιμοποιήσετε λίγο περισσότερο χώρο ο οποίος θα πρέπει να είναι  $O(n)$ .)*

# Άσκηση

Δίπλα σε κάθε στοιχείο γράφουμε και το min μέχρι το τρέχον στοιχείο.



# THANKS FOR LISTENING



# ANY QUESTIONS?

[makeameme.org](http://makeameme.org)