

Δομές Δεδομένων σε C

Μάθημα 2:

Στοίβα

Δημήτρης Ψούνης



www.psounis.gr



Περιεχόμενα Μαθήματος

A. Θεωρία

1. Στοίβα

1. Ορισμός Στοίβας
2. Βασικές Πράξεις
3. Υλοποίηση σε C: Δηλώσεις
4. Υλοποίηση σε C: Αρχικοποίηση
5. Υλοποίηση σε C: Έλεγχοι – Κενή Στοίβα και Γεμάτη Στοίβα
6. Υλοποίηση σε C: Ωθηση Στοιχείου
7. Υλοποίηση σε C: Εξαγωγή Στοιχείου
8. Υλοποίηση σε C: Παράδειγμα

2. Εφαρμογές Στοίβας

1. Μετατροπή Δεκαδικού σε Δυαδικό
2. Υπολογισμός Μεταθεματικής Παράστασης

B. Ασκήσεις

Σημείωση: Το μάθημα αυτό απαιτεί να έχουν μελετηθεί τα μαθήματα 1..14 της γλώσσας προγραμματισμού C.



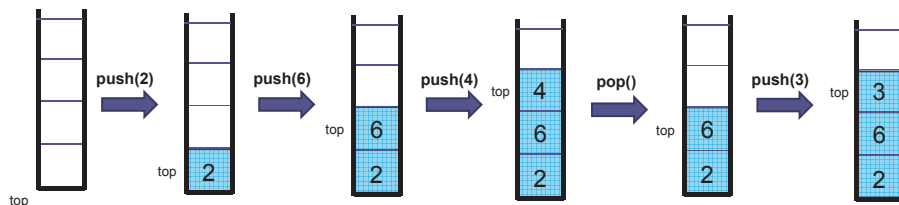
A. Θεωρία

1. Στοίβα

1. Ορισμός Στοίβας

Η «**Στοίβα**» είναι μια δομή δεδομένων με γραμμική διάταξη στην οποία:

- Η ώθηση (push) και η εξαγωγή (pop) ενός στοιχείου, γίνεται στην κεφαλή (top) της στοίβας.



Παράδειγμα:

- Η στοίβα των πιάτων στον νεροχύτη!

Σημαντική Ιδιότητα:

- Το τελευταίο στοιχείο που προστέθηκε στη στοίβα είναι το πρώτο που θα εξαχθεί (Last In – First Out: **LIFO**)



A. Θεωρία

1. Στοίβα

2. Βασικές Πράξεις

Οι **βασικές πράξεις** σε μία στοίβα είναι:

- Αρχικοποίηση της στοίβας (**init**)
- Ωθηση ενός στοιχείου στην στοίβα (**push**)
- Εξαγωγή ενός στοιχείου από τη στοίβα (**pop**)
- Έλεγχος αν η στοίβα είναι κενή (**empty**)
- Έλεγχος αν η στοίβα είναι γεμάτη (**full**)

Υπάρχουν δύο υλοποιήσεις:

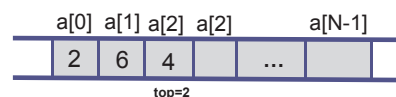
- Με στατικό πίνακα (σημερινό μάθημα)
- Με απλά συνδεδεμένη λίστα (επόμενο μάθημα)

Θα υλοποιήσουμε στην C τη στοίβα με έναν πίνακα N θέσεων. Πρόσθετα θα κρατάμε μία μεταβλητή (top) που κρατάει την κεφαλή της στοίβας.

Στοίβα:



Γλώσσα C



A. Θεωρία

1. Στοιβά

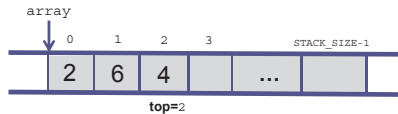
3. Υλοποίηση σε C: Δηλώσεις

Οι **δηλώσεις** σε C είναι οι ακόλουθες:

- Η στοιβά είναι μία δομή (struct) με τα εξής στοιχεία:
 - Ένας **πίνακας** με **STACK_SIZE** στοιχεία
 - Μία ακέραια μεταβλητή (top) που δείχνει τη θέση που βρίσκεται η κεφαλή της στοιβάς με τιμή:
 - Από 0...STACK_SIZE-1 αν η στοιβά έχει τουλάχιστον ένα στοιχείο
 - -1 αν η στοιβά είναι άδεια.

```
#define STACK_SIZE 10      /* Megethos pinaka stoivas */
typedef int elem;          /* typos dedomenwn stoivas */
struct stack{
    elem array[STACK_SIZE]; /* pinakas stoxeiwn */
    int top;                 /* koryfi tis stoivas */
};
typedef struct stack STACK; /* Sinwnimo tis stoivas */
```

Αναπαράσταση:



A. Θεωρία

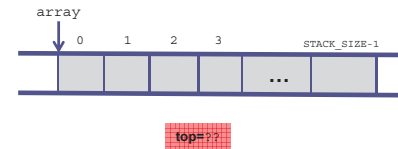
1. Στοιβά

4. Υλοποίηση σε C: Αρχικοποίηση Στοιβάς

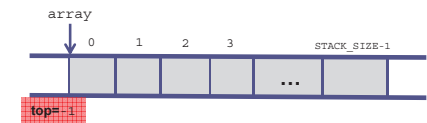
Η αρχικοποίηση γίνεται θέτοντας την κορυφή ίση με -1

```
/* ST_init(): arxikopoiei ti lista */
void ST_init(STACK *s)
{
    s->top=-1;
}
```

ΠΡΙΝ:



ΜΕΤΑ:



Προσοχή:

- Πάντα προτού ξεκινήμε την χρήση της στοιβάς θα πρέπει να καλούμε μία φορά αυτήν τη συνάρτηση!

A. Θεωρία

1. Στοιβά

5. Υλοποίηση σε C: Έλεγχος – Κενή Στοιβά και Γεμάτη Στοιβά

Ο **έλεγχος** αν η στοιβά είναι **κενή** (αντίστοιχα **γεμάτη**), γίνεται βλέποντας αν η μεταβλητή top είναι ίση με -1 (αντίστοιχα N-1)

```
/* ST_empty(): epistrefei TRUE/FALSE
 * analoga me to an i stoiva einai adeia */
int ST_empty(STACK s)
{
    return s.top==-1;
}
```

```
/* ST_full(): epistrefei TRUE/FALSE
 * analoga me to an i stoiva einai gemati */
int ST_full(STACK s)
{
    return s.top==STACK_SIZE-1;
}
```

A. Θεωρία

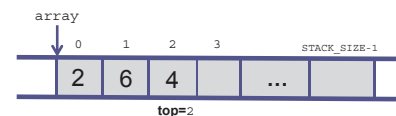
1. Στοιβά

6. Υλοποίηση σε C: Ώθηση Στοιχείου

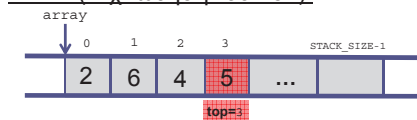
Η **ώθηση** στη στοιβά γίνεται προωθώντας το στοιχείο στη θέση top+1 (εφόσον χωράει στην στοιβά)

```
/* ST_push(): Eisagei to x sti stoiva s
 * epistrefei TRUE: se periptwsi epitixias
 * FALSE: se periptwsi apotixias */
int ST_push(STACK *s, elem x)
{
    if (ST_full(*s))
        return FALSE;
    else
    {
        s->top++;
        s->array[s->top]=x;
        return TRUE;
    }
}
```

ΠΡΙΝ:



ΜΕΤΑ (π.χ. ώθηση του «5»):



A. Θεωρία

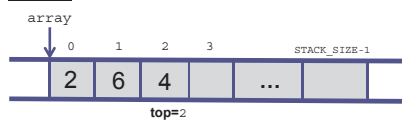
1. Στοιβά

6. Υλοποίηση σε C: Εξαγωγή Στοιχείου

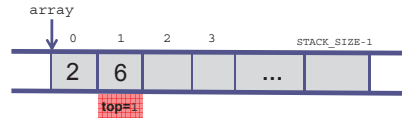
Η εξαγωγή γίνεται βγάζοντας το στοιχείο της θέσης top (εφόσον η στοιβά δεν είναι άδεια)

```
/* ST_pop(): Kanei eksagwgi poy einai stin korifi tis listas
 *          epistrefei TRUE: se periptwsi epitixias
 *          FALSE: se periptwsi apotixias */
int ST_pop(STACK *s, elem *x)
{
    if (ST_empty(*s))
        return FALSE;
    else
    {
        *x=s->array[s->top];
        s->top--;
        return TRUE;
    }
}
```

ΠΡΙΝ:



ΜΕΤΑ:



A. Θεωρία

1. Στοιβά

7. Υλοποίηση σε C: Παράδειγμα

- Μεταγλωττίστε και εκτελέστε το project stack.dev στο οποίο:
 - Το stack.h έχει τα πρωτότυπα των συναρτήσεων και την δήλωση της δομής
 - Το stack.c έχει τα σώματα των συναρτήσεων
 - Το stack_main.c έχει ένα πρόγραμμα που επιδεικνύει την χρήση μίας στοιβάς ακεραίων.
- «Παίξτε» με το πρόγραμμα ώστε να γίνει πλήρως κατανοητή η λειτουργία της στοιβάς.

Υπενθύμιση:

- Το σπάσιμο ενός προγράμματος σε επιμέρους αρχεία μελετήθηκε στο «Γλώσσα C – Μάθημα 14: Εμβέλεια Μεταβλητών»

A. Θεωρία

2. Εφαρμογές Στοιβάς

1. Μετατροπή Δεκαδικού σε Δυαδικό

Υπενθύμιση:

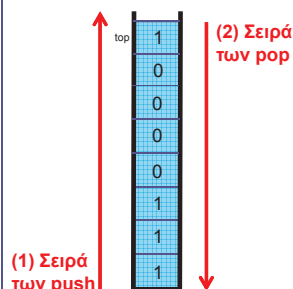
- Για την μετατροπή από δεκαδικό σε δυαδικό κάνουμε διαδοχικές διαιρέσεις με το 2
 - Εισάγουμε τα διαδοχικά υπόλοιπα σε μία στοιβά
- Εξάγοντας από την στοιβά έχουμε τον δυαδικό αριθμό!

ΠΑΡΑΔΕΙΓΜΑ:

ΥΠΟΛΟΓΙΣΜΟΙ

Αριθμός /2	Πηλίκο	Υπόλοιπο
135/2	67	1
67/2	33	1
33/2	16	1
16/2	8	0
8/2	4	0
4/2	2	0
2/2	1	0
1/2	0	1

ΣΤΟΙΒΑ



ΔΥΑΔΙΚΟΣ ΑΡΙΘΜΟΣ

(2) Σειρά των pop
1 0 0 0 1 1 1

Άρα $(135)_{10} = (1000111)_2$

A. Θεωρία

2. Εφαρμογές Στοιβάς

2. Υπολογισμός Μεταθεματικής Παράστασης

- Οι μαθηματικές παραστάσεις που ξέρουμε γράφονται σε **ενδοθεματική μορφή** (ο τελεστής είναι εσωτερικά των αριθμών):
 - π.χ.: **A+B**
- Υπάρχει και η **προθεματική μορφή** (πρώτα ο τελεστής και μετά οι αριθμοί):
 - Είναι: **+ A B**
- Αλλά και η **μεταθεματική μορφή** (πρώτα οι αριθμοί και μετά ο τελεστής):
 - Είναι: **A B +**

Σημείωση:

- Η **μεταθεματική μορφή** είναι χρήσιμη στον υπολογισμό παραστάσεων στους επεξεργαστές με χρήση καταχωρητών και είναι ιδιαίτερως χρήσιμη διότι δεν απαιτεί την χρήση παρενθέσεων.
- Ο υπολογισμός μπορεί να γίνει εύκολα με χρήση του «αριστερότερου» τελεστή. Αυτός θα εφαρμόζεται στους δύο αμέσως αριστερούς του αριθμούς.

- Π.χ. η παράσταση: $(5+3)*4-2$
 - σε μεταθεματική μορφή γράφεται διαδοχικά (με βάση την προτεραιότητα):
 - $(5 + 3) * 4 - 2$
 - $(5 3 +) * 4 - 2$
 - $(5 3 +) 4 * - 2$
 - $5 3 + 4 * 2 -$

A. Θεωρία

2. Εφαρμογές Στοιβάς

2. Υπολογισμός Μεταθεματικής Παράστασης

- Το σημαντικό είναι ότι μπορούμε να υπολογίσουμε τη μεταθεματική παράσταση με χρήση στοιβάς ενόσω διαβάζουμε την παράσταση (χωρίς να πρέπει να την έχουμε αποθηκευμένη!)

- Διάβασμα 2 (αριθμός). Τοποθέτηση στη στοιβά

ΕΙΣΟΔΟΣ: $2\ 4 + 6\ 1\ */$ 

ΣΤΟΙΒΑ:



- Διάβασμα 4 (αριθμός). Τοποθέτηση στη στοιβά

ΕΙΣΟΔΟΣ: $2\ 4 + 6\ 1\ */$ 

ΣΤΟΙΒΑ:



A. Θεωρία

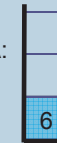
2. Εφαρμογές Στοιβάς

2. Υπολογισμός Μεταθεματικής Παράστασης

- Διάβασμα + (τελεστής). Εξαγωγή δύο στοιχείων (4 και 2). Εκτέλεση πράξης (4+2). Εισαγωγή του αποτελέσματος στη στοιβά

ΕΙΣΟΔΟΣ: $2\ 4 + 6\ 1\ */$ 

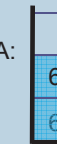
ΣΤΟΙΒΑ:



- Διάβασμα 6 (αριθμός). Τοποθέτηση στη στοιβά

ΕΙΣΟΔΟΣ: $2\ 4 + 6\ 1\ */$ 

ΣΤΟΙΒΑ:



A. Θεωρία

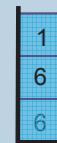
2. Εφαρμογές Στοιβάς

2. Υπολογισμός Μεταθεματικής Παράστασης

- Διάβασμα 6 (αριθμός). Τοποθέτηση στη στοιβά

ΕΙΣΟΔΟΣ: $2\ 4 + 6\ 1\ */$ 

ΣΤΟΙΒΑ:



- Διάβασμα * (τελεστής). Εξαγωγή δύο στοιχείων (1 και 6). Εκτέλεση πράξης (1*6). Εισαγωγή του αποτελέσματος (6) στη στοιβά

ΕΙΣΟΔΟΣ: $2\ 4 + 6\ 1\ */$ 

ΣΤΟΙΒΑ:



A. Θεωρία

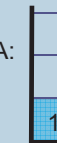
2. Εφαρμογές Στοιβάς

2. Υπολογισμός Μεταθεματικής Μορφής

- Διάβασμα / (τελεστής). Εξαγωγή δύο στοιχείων (6 και 6). Εκτέλεση πράξης (6/6). Εισαγωγή του αποτελέσματος (1) στη στοιβά

ΕΙΣΟΔΟΣ: $2\ 4 + 6\ 1\ */$ 

ΣΤΟΙΒΑ:



- Ο υπολογισμός ολοκληρώθηκε και το αποτέλεσμα είναι ίσο με 1.
- Ισοδύναμα η παράσταση: $2\ 4 + 6\ 1\ */$
 - Αντιστοιχεί στην ενδοθεματική παράσταση: $(2+4)/(6*1)$ που είναι όντως ίσο με 1.



Β. Ασκήσεις

Εφαρμογή 1: Μία στοίβα χαρακτήρων

- Μετατρέψτε την βιβλιοθήκη που δημιουργήσαμε στο μάθημα ώστε να υλοποιείται μία στοίβα χαρακτήρων.



Β. Ασκήσεις

Εφαρμογή 2: Μία στοίβα φοιτητών

- Μετατρέψτε την βιβλιοθήκη που δημιουργήσαμε στο μάθημα ώστε να υλοποιείται μία στοίβα φοιτητών!
 - Για κάθε φοιτητή θέλουμε να διατηρείται μία συμβολοσειρά με το ονοματεπώνυμό του και ένας ακέραιος αριθμός με το βαθμό του.



Β. Ασκήσεις

Εφαρμογή 3: Μετατροπή σε Δυαδικό

- Γράψτε ένα πρόγραμμα σε C το οποίο
 - Θα διαβάζει ένα θετικό ακέραιο αριθμό
 - Θα τον μετατρέπει σε δυαδικό με χρήση μίας στοίβας ακεραίων όπως περιγράψαμε στο μάθημα.



Β. Ασκήσεις

Εφαρμογή 4: Υπολογισμός Μεταθεματικής Παράστασης

- Γράψτε ένα πρόγραμμα σε C το οποίο
 - Θα διαβάζει μία παράσταση σε μεταθεματική μορφή από την είσοδο χωρίς να την αποθηκεύει (θα διαβάζεται χαρακτήρα-χαρακτήρα).
 - Χάριν απλότητας, θεωρείστε ότι ο χρήστης θα εισάγει μόνο τους τελεστές +, -, *, / και τα ψηφία 0..9.
 - Θα υπολογίζει την παράσταση χρησιμοποιώντας μία στοίβα πραγματικών αριθμών όπως περιγράψαμε στο μάθημα.