



Διάλεξη 11: Φροντιστήριο για Στοίβες

Διδάσκων: Παναγιώτης Ανδρέου

ΑΤΔ Στοίβα - Πράξεις

- Θυμηθείτε τον ΑΤΔ στοίβα με τις πράξεις του:

MakeEmptyStack() δημιουργήσε την
κενή στοίβα $\langle \rangle$.

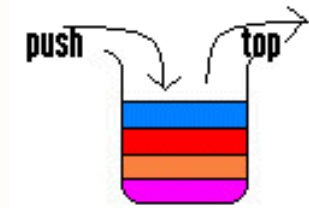
IsEmptyStack(S) επέστρεψε τη λογική τιμή που εκφράζει το
αν η S είναι κενή.

Push(x,S) εισήγαγε τον κόμβο x στη στοίβα S .

Pop(S) διέγραψε τον κόμβο κορυφής της S .

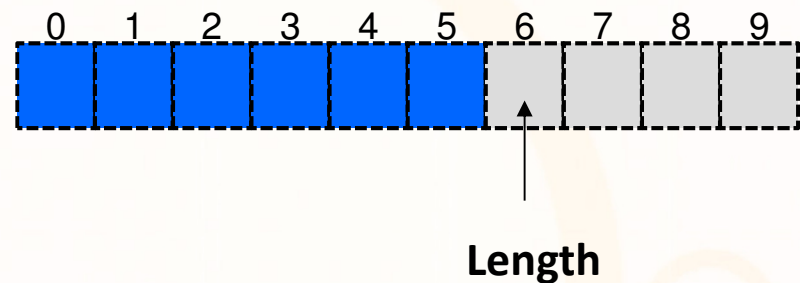
Top(S) δώσε τον κόμβο κορυφής της S .

- Είχαμε πει ότι αυτές οι πράξεις μπορούν να υλοποιηθούν με δύο τρόπους: α) με στατική δέσμευση μνήμης; Και β) με δυναμική δέσμευση μνήμης



Στοίβα με Στατική Δέσμευση Μνήμης

- Ο πιο απλός τρόπος είναι η χρήση μονοδιάστατου πίνακα. Χρειάζεται να γνωρίζουμε από την αρχή το μήκος της λίστας.
- Για την παράσταση στοίβας με στοιχεία $\alpha_1, \alpha_2, \dots, \alpha_n$ χρειαζόμαστε ένα πίνακα A στον οποίο θα αποθηκεύσουμε τα στοιχεία της στοίβας, $A[i-1] = \alpha_i$. Πρέπει να γνωρίζουμε ανά πάσα στιγμή που βρίσκεται η κορυφή της στοίβας.
- Έτσι χρησιμοποιούμε μια εγγραφή με δύο πεδία
 1. ένα πίνακα $A[0..n-1]$, και
 2. μια μεταβλητή $Length$ τύπου ακέραιος (που συγκρατεί τη θέση κορυφής).



Στοίβας με Στατική Δέσμευση Μνήμης: Υλοποίηση

- Ο τύπος δεδομένων για τη **Στοίβα** με **Στατική** Δέσμευση Μνήμης είναι:

```
typedef struct {  
    type list[ size ]; //π.χ., size=10  
    int Length;  
} STACK;
```

→ Στατική Δέσμευση size
θέσεων μνήμης

- Υλοποίηση πράξεων:

```
void MakeEmpty(STACK *S){  
    S->Length = 0;  
}
```

```
int IsEmpty(STACK *S){  
    return (S->Length == 0);  
}
```

```
type Top(STACK *S){  
    if ( !IsEmpty(S) )  
        return S->list[(S->Length)-1];  
}
```

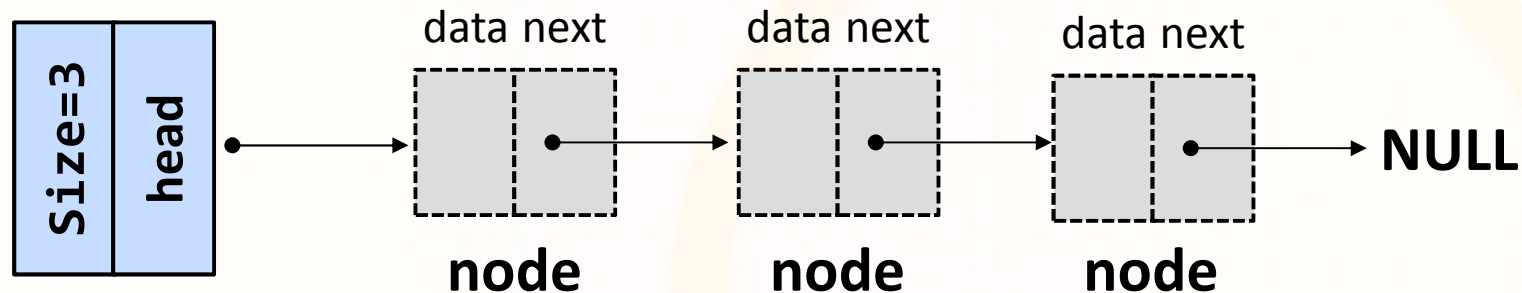
```
void Push(type x, STACK *S){  
    if ((S->Length) < size) {  
        S->list[S->Length]= x;  
        (S->Length)++;  
    }  
}
```

```
void Pop(STACK *S){  
    if ( !IsEmpty(S) )  
        (S->Length)--;  
}
```

Στοίβα με Δυναμική Δέσμευση Μνήμης

- Για την παράσταση μιας στοίβας με στοιχεία $\alpha_1, \alpha_2, \dots, \alpha_n$ χρησιμοποιούμε μια συνδεδεμένη λίστα από κόμβους.
- Κάθε κόμβος αποτελείται από ένα στοιχείο (στοιχεία της στοίβας) και από ένα δείκτη (προς τον επόμενο κόμβο της στοίβας). Η κορυφή της στοίβας είναι ο πρώτος κόμβος της λίστας,
- Χρησιμοποιούμε μια μεταβλητή για να φυλάγουμε στοιχεία σχετικά με τη στοίβα π.χ. **μέγεθος** (size) και δείκτη προς την **κορυφή της στοίβας (head)**.

STACK



Στοίβα με Δυναμική Δέσμευση Μνήμης (συν.)

- Υλοποίηση δομής (χρησιμοποιείται `int` αντί για `type`):

```
typedef struct node {  
    int data;  
    struct node *next;  
} NODE;
```

```
typedef struct stack {  
    NODE *head;  
    int size;  
} STACK;
```

- Υλοποίηση πράξεων (χρησιμοποιείται `int` αντί για `type`):

```
void MakeEmptyStack(STACK *S) {  
    S->size = 0;  
    S->head = NULL;  
}
```

```
int IsEmpty(STACK *S) {  
    return (S->size == 0);  
}
```

```
void Pop(STACK *S) {  
    NODE *p = NULL;  
    if ((S->size) > 0){  
        p = S->head;  
        S->head = p->next;  
        free(p);  
        (S->size)--;  
    }  
}
```

Στοίβα με Δυναμική Δέσμευση Μνήμης (συν.)

```
void Push(STACK *S, int x){
    NODE *p = NULL;
    p = (NODE *)
        malloc(sizeof(NODE));
    p->data = x;
    p->next = S->head;
    S->head = p;
    (S->size) ++;
}
```

```
int Top(STACK *S){
    if ((S->size) > 0){
        return S->head->data;
    }
}
```

ΒΟΗΘΗΤΙΚΗ ΣΥΝΑΡΤΗΣΗ ΓΙΑ ΕΚΤΥΠΩΣΗ

```
void PrintStack(STACK *S){
    NODE *tmp = S->head;
    for(int i=0; i<(S->size); i++){
        printf("%d ", tmp->data);
        tmp = tmp->next;
    }
    printf("\n");
}
```

Πρόβλημα: Μετατροπή Δεκαδικού σε Δυαδικό

- **Συνάρτηση `int2Binary`:** εκτυπώνει την δυαδική αναπαράσταση ενός φυσικού αριθμού που δίδεται ως παράμετρος στην συνάρτηση.

- Πρότυπο συνάρτησης: **`void int2Binary(int a)`**

- Παραδείγματα:

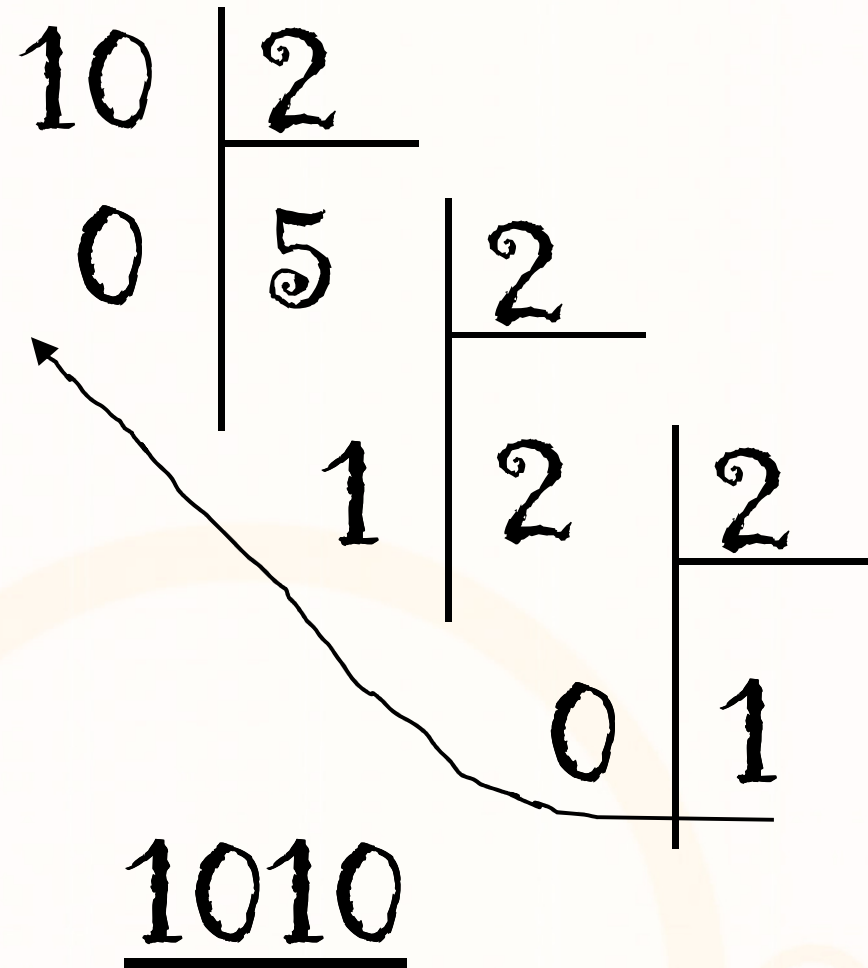
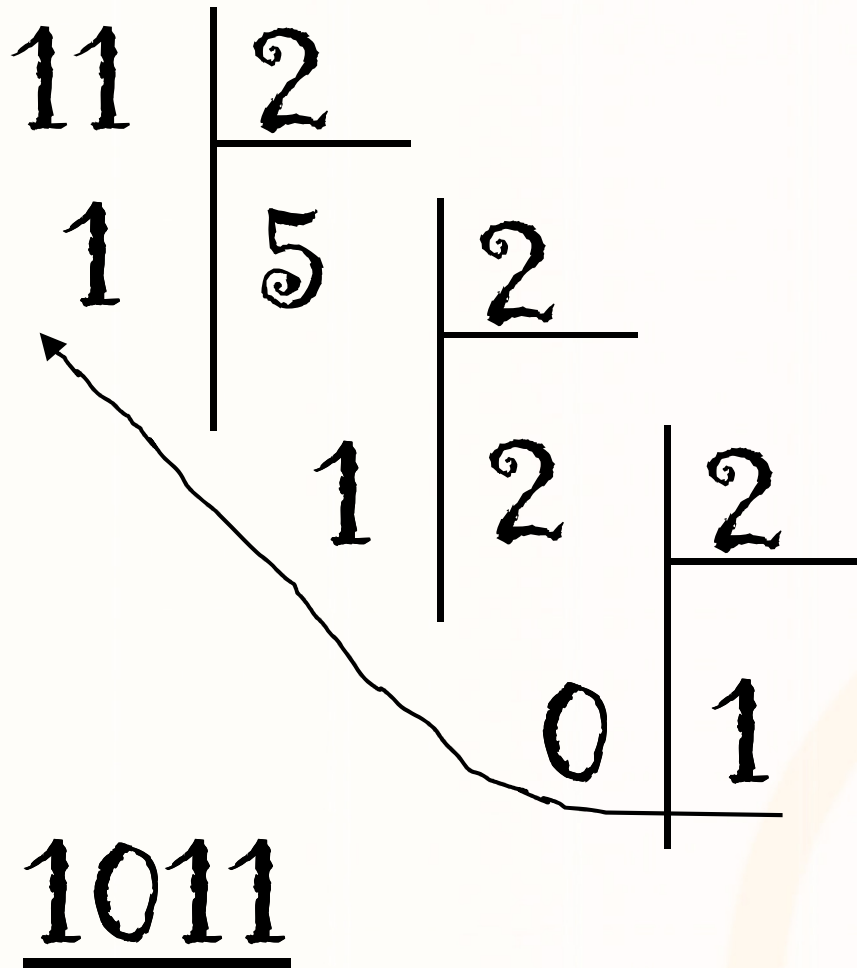
<code>int2Binary(5)</code>	τυπώνει στην οθόνη	101
<code>int2Binary(11)</code>	τυπώνει στην οθόνη	1011
<code>int2Binary(15)</code>	τυπώνει στην οθόνη	1111
<code>int2Binary(124)</code>	τυπώνει στην οθόνη	1111100

Πρόβλημα: Μετατροπή Δεκαδικού σε Δυαδικό(συν.)

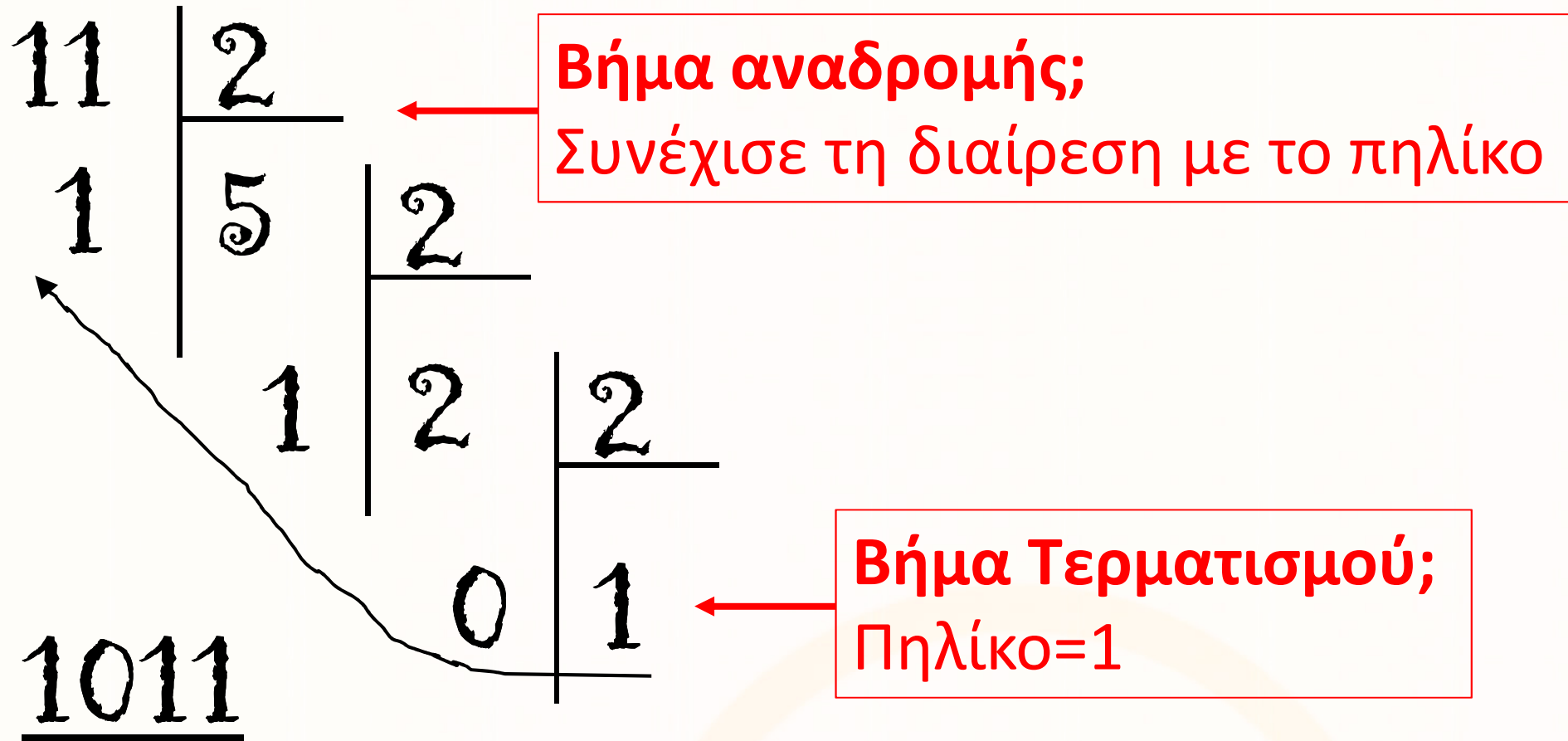
- Παράδειγμα Υπολογισμού:

`int2Binary(11) → 1011`

`int2Binary(10) → 1010`

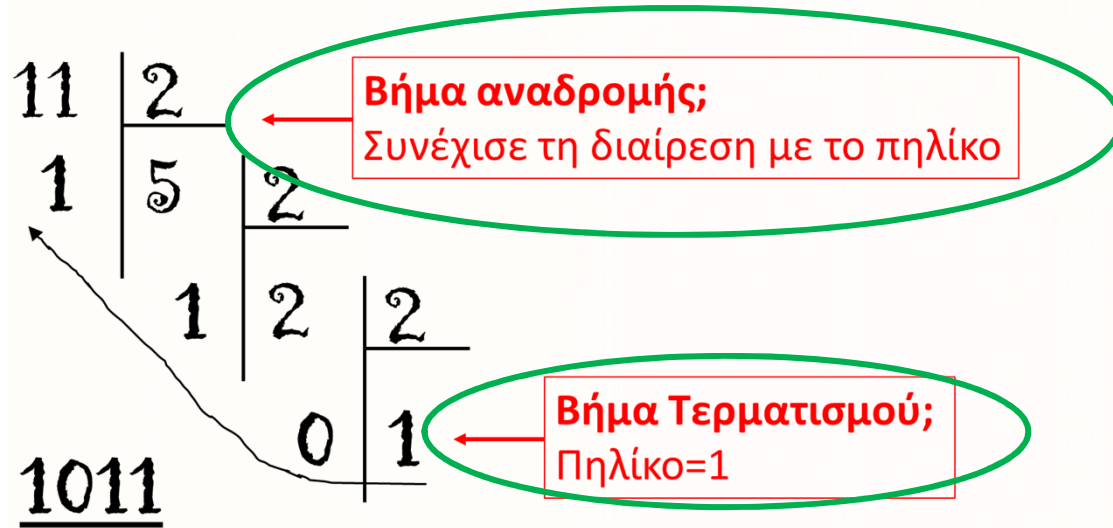


Υλοποίηση συνάρτησης int2Binary: Α. Αναδρομή



Τι κάνουμε στο τέλος;
Τυπώνουμε το πηλίκο=1 και όλα τα υπόλοιπα

Υλοποίηση συνάρτησης int2Binary: Α. Αναδρομή



Τι κάνουμε στο τέλος;
Τυπώνουμε το πηλίκο=1 και όλα τα υπόλοιπα

```
void int2Binary_R( int x ){  
  
  
  
  
  
  
  
  
  
}
```