

ΣΤΟΙΒΕΣ

2.1 Εισαγωγή

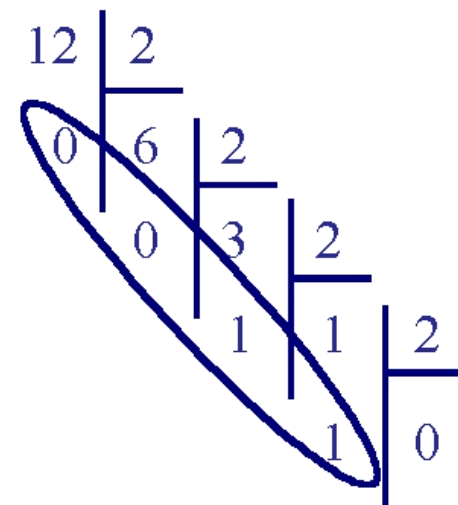
Για να γίνει κατανοητή η έννοια της στοίβας ας θεωρήσουμε το πρόβλημα αποθήκευσης στη μνήμη του υπολογιστή αριθμών που παριστάνονται στο δεκαδικό σύστημα:

- Οι αριθμοί που παριστάνονται στο δεκαδικό σύστημα, προκειμένου να αποθηκευτούν στη μνήμη του υπολογιστή πρέπει να μετατραπούν στο δυαδικό.
- Μια τέτοια μετατροπή συνεπάγεται διαδοχικές διαιρέσεις με τον αριθμό 2 και ένα τρόπο αποθήκευσης των ακέραιων υπολοίπων των διαιρέσεων αυτών, ώστε να προκύψει στο τέλος ο δυαδικός αριθμός.

Η έννοια της στοίβας

Παράδειγμα: μετατροπή του δεκαδικού 12 σε δυαδικό.

- διαιρούμε πρώτα το 12 με το 2 και έχουμε πηλίκο 6 και υπόλοιπο **0**
- στη συνέχεια, διαιρούμε το πηλίκο 6 με το 2 και έχουμε νέο πηλίκο 3 και νέο υπόλοιπο **0**
- το νέο πηλίκο 3 το διαιρούμε πάλι με 2 και παίρνουμε πηλίκο 1 και ακέραιο υπόλοιπο **1**
- τέλος, διαιρούμε το 1 με 2 και έχουμε πηλίκο 0 και ακέραιο υπόλοιπο **1**



Ύστερα από αυτές τις διαιρέσεις μπορούμε να σχηματίσουμε το δυαδικό αριθμό που αντιστοιχεί στο δεκαδικό 12, γράφοντας τα ακέραια υπόλοιπα των διαιρέσεων το ένα δίπλα στο άλλο, ξεκινώντας από το τελευταίο και προχωρώντας προς το πρώτο.

Επομένως, ο δυαδικός αριθμός που αντιστοιχεί στο δεκαδικό 12 είναι ο **1100**.

Η έννοια της στοίβας

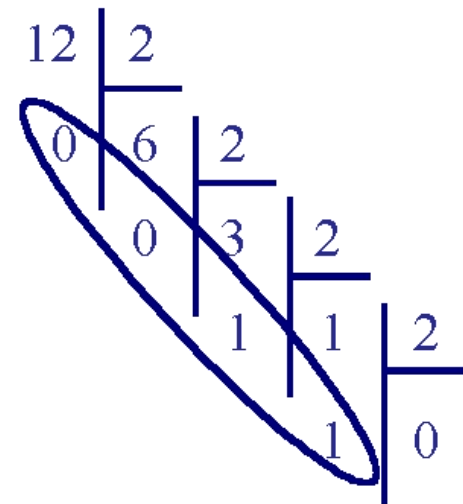
Είναι φανερό ότι για την επίλυση του παραπάνω προβλήματος χρειάζεται να

"**στοιβάσουμε**" τα ακέραια υπόλοιπα των διαιρέσεων

και στο τέλος

να τα **ανακτήσουμε** από τη στοίβα,

παίρνοντας πρώτα το τελευταία αποθηκευμένο στοιχείο και προχωρώντας προς το πρώτο.



Αλγόριθμος μετατροπής δεκαδικού σε δυαδικό

(*Δέχεται: Θετικό ακέραιο αριθμό *Number*.
Λειτουργία: Μετατρέπει τον αριθμό *Number* από δεκαδικό σε δυαδικό.
Έξοδος: Η δυαδική αναπαράσταση του *Number*.*)

1. Δημιούργησε μια κενή στοίβα για την αποθήκευση των υπολοίπων
2. Όσο *Number* $\neq 0$ επανάλαβε
 - α. Υπολόγισε το *Ακέραιο Υπόλοιπο* της διαίρεσης του *Number* με 2
 - β. Εισήγαγε το *Ακέραιο Υπόλοιπο* στην κορυφή της στοίβας των ακέραιων υπολοίπων
 - γ. Αντικατέστησε τον αριθμό *Number* με το ακέραιο πηλίκο της διαίρεσης του *Number* με 2Τέλος_επανάληψης
3. Όσο η στοίβα των ακέραιων υπολοίπων δεν είναι κενή επανάλαβε
 - α. Διάγραψε το *Ακέραιο Υπόλοιπο* από την κορυφή της στοίβας των υπολοίπων
 - β. Εμφάνισε το *Ακέραιο Υπόλοιπο*Τέλος_επανάληψης

Από τον παραπάνω αλγόριθμο φαίνεται ότι χρειαζόμαστε μια δομή δεδομένων με την οποία

- να αποθηκεύουμε στοιχεία με τη σειρά και
- να τα ανακτούμε με αντίστροφη σειρά,

δηλαδή χρειαζόμαστε μια "τελευταίος μέσα - πρώτος έξω" (Last In - First Out, **LIFO**) δομή.

Μια τέτοια δομή ονομάζεται στοίβα (stack).

Μια **στοίβα** είναι μια "**τελευταίος μέσα - πρώτος έξω**" δομή και αποτελείται από μια συλλογή δεδομένων στην οποία όλες οι εισαγωγές και διαγραφές δεδομένων γίνονται στην μία άκρη της, που ονομάζεται κορυφή (**top**).

- Δημιουργία μιας κενής στοίβας (CreateStack)
- Έλεγχος αν μια στοίβα είναι κενή (EmptyStack)
- Ανάκτηση και διαγραφή του στοιχείου που βρίσκεται στην κορυφή της στοίβας (Pop)
- Εισαγωγή νέου στοιχείου στην κορυφή της στοίβας (Push)



Η λειτουργία ανάκτησης και διαγραφής του κορυφαίου στοιχείου της στοίβας ονομάζεται συνήθως **απώθηση (pop)** και η λειτουργία εισαγωγής ενός νέου στοιχείου στην κορυφή της στοίβας ονομάζεται **ώθηση (push)**.

Συλλογή στοιχείων δεδομένων:

Μια συλλογή στοιχείων δεδομένων με γραμμική διάταξη προσπελάσιμη μόνο από μια θέση, που ονομάζεται κορυφή της στοίβας.

Βασικές λειτουργίες:

Δημιουργία κενής στοίβας (CreateStack)

Λειτουργία: Δημιουργεί μια κενή στοίβα.

Επιστρέφει: Μια κενή στοίβα.

Έλεγχος αν μια στοίβα είναι κενή (EmptyStack):

Δέχεται: Μια στοίβα.

Λειτουργία: Ελέγχει αν η στοίβα είναι κενή.

Επιστρέφει: TRUE, αν η στοίβα είναι κενή, FALSE διαφορετικά.

Ανάκτηση και διαγραφή του στοιχείου που βρίσκεται στην κορυφή της στοίβας (Pop)

Δέχεται: Μια στοίβα.

Λειτουργία: Ανακτά και διαγράφει το στοιχείο στην κορυφή της στοίβας.

Επιστρέφει: Κορυφαίο στοιχείο της στοίβας και την τροποποιημένη στοίβα.

Εισαγωγή νέου στοιχείου στην κορυφή της στοίβας (Push)

Δέχεται: Μια στοίβα και ένα στοιχείο δεδομένων.

Λειτουργία: Εισάγει το στοιχείο στην κορυφή της στοίβας.

Επιστρέφει: Την τροποποιημένη στοίβα.

Αν υποθέσουμε ότι ο παραπάνω ΑΤΔ έχει υλοποιηθεί έτσι ώστε οι διαδικασίες CreateStack, Pop και Push δηλωμένες ως εξής:

```
void CreateStack(StackType *Stack);  
void Push(StackType *Stack, StackElementType Item);  
void Pop(StackType *Stack, StackElementType *Item);
```

και η boolean συνάρτηση EmptyStack δηλωμένη ως:

```
boolean EmptyStack(StackType Stack);
```

υλοποιούν τις βασικές λειτουργίες της στοίβας, τότε είναι εύκολο να γραφεί ένα μέρος προγράμματος για την υλοποίηση του αλγόριθμου μετατροπής ενός δεκαδικού αριθμού στον αντίστοιχο δυαδικό:

Υλοποίηση αλγορίθμου μετατροπής

```
CreateStack(&StackOfRemainders);  
while (Number!= 0)  
{  
    Remainder=Number%2;  
    Push(&StackOfRemainders,Remainder);  
    Number=Number / 2;  
}  
printf("Δυαδική αναπαράσταση: ");  
while (!EmptyStack(StackOfRemainders))  
{  
    Pop(&StackOfRemainders,&Remainder);  
    printf("%d",Remainder);  
}  
printf("\n");
```