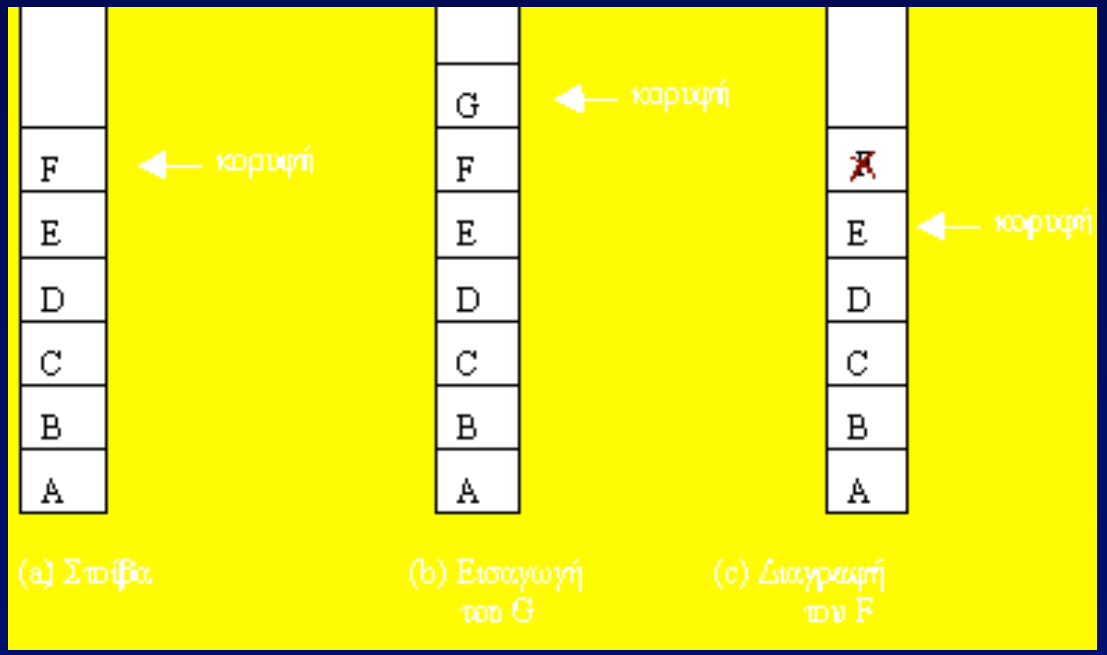
ΣΤΟΙΒΕΣ (stacks)

Η στοίβα είναι μια συλλογή δεδομένων με γραμμική διάταξη στην οποία όλες οι εισαγωγές και οι διαγραφές γίνονται στο ένα άκρο που λέγεται **κορυφή (top)** της στοίβας



Σχήμα: Λειτουργία Στοίβας

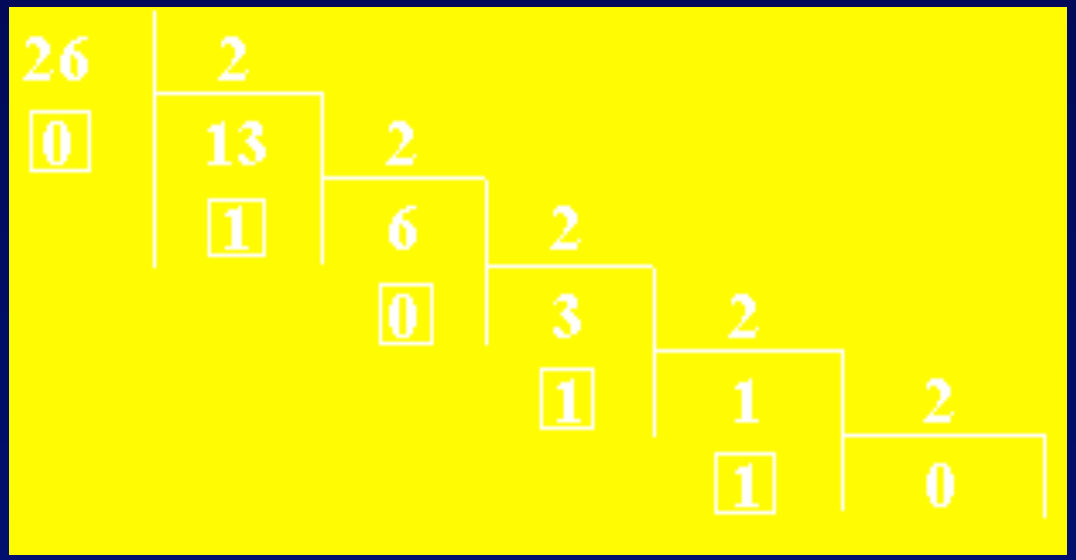
Βασικές πράξεις του ΑΤΔ στοίβα

- Δημιουργία
- Εισαγωγή (ώθηση, push)
- Εξαγωγή (pop)
- Κενή

Πρόβλημα: μετατροπή ενός θετικού ακεραίου από το δεκαδικό στο δυαδικό σύστημα

Αλγόριθμος: τα διαδοχικά υπόλοιπα που προκύπτουν από τις συνεχείς διαιρέσεις του αριθμού με το 2, αποτελούν τα δυαδικά ψηφία του αριθμού από τα δεξιά (less significant digit) προς τα αριστερά (most significant digit).

Για παράδειγμα, ο 26 στο δυαδικό σύστημα παριστάνεται σαν 11010 όπως δείχνει και το Σχήμα. Τα ψηφία του δυαδικού συστήματος που αποτελούν τον αριθμό 26 δημιουργούνται με σειρά από το λιγότερο σημαντικό στο περισσότερο σημαντικό.



Αλγόριθμος για τη μετατροπή ενός θετικού ακέραιου από το δεκαδικό στο δυαδικό σύστημα

Οσο ο αριθμός <> 0 να εκτελούνται τα παρακάτω:

- Υπολογισμός του υπολοίπου που προκύπτει από τη διαίρεση του αριθμού με το 2.
- Εισαγωγή του υπολοίπου στη στοίβα με τα υπόλοιπα.
- Αντικατάσταση του αριθμού με το ακέραιο πηλίκο του αριθμού δια του 2.

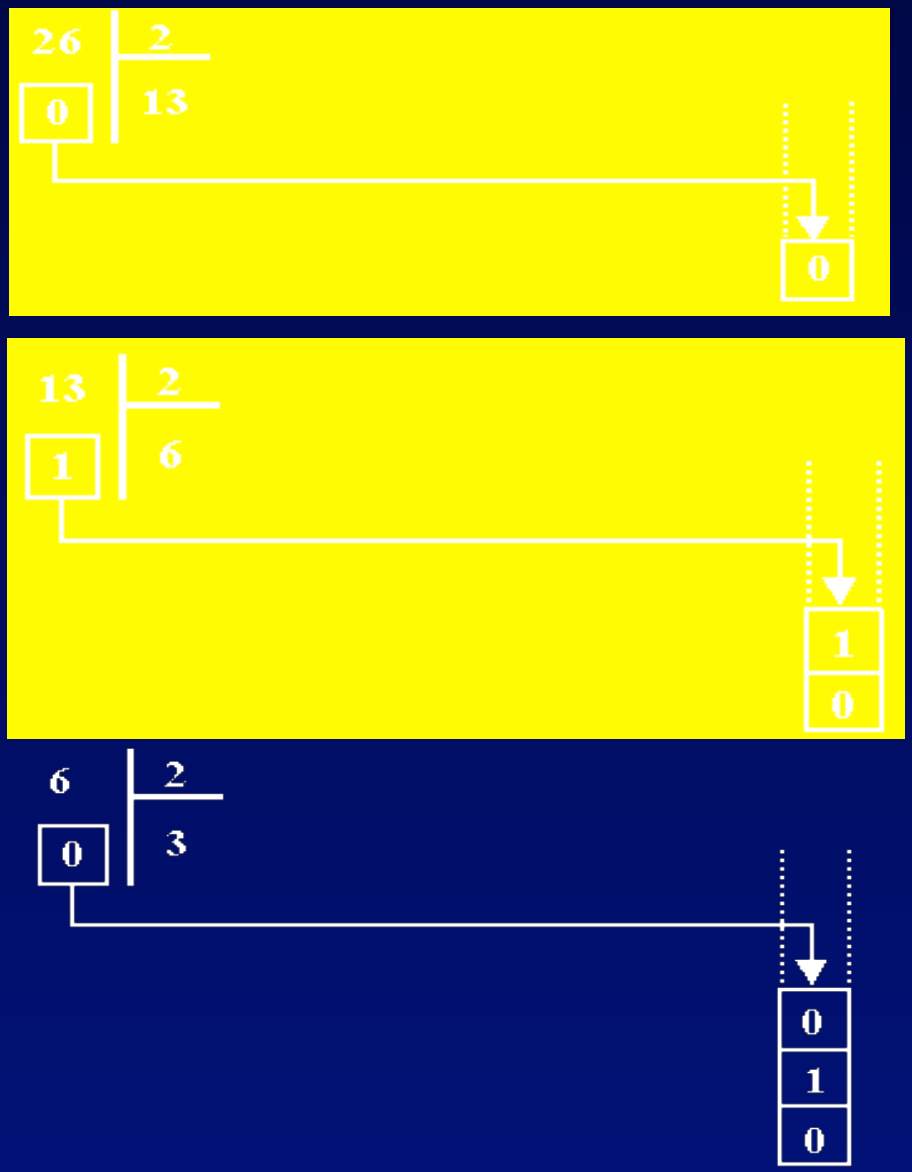
Οσο η στοίβα των υπολοίπων δεν είναι άδεια να εκτελούνται τα παρακάτω:

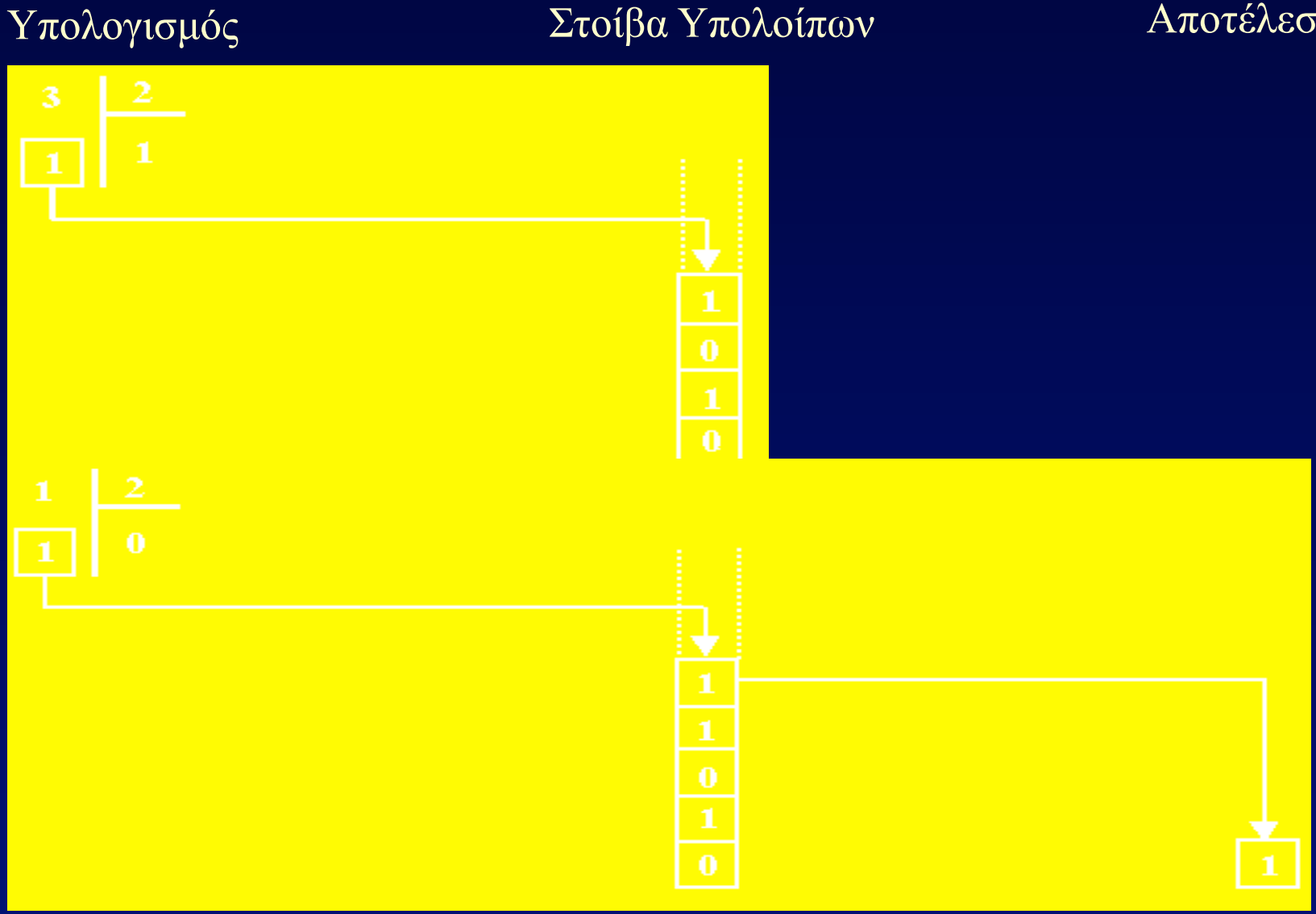
- Εξαγωγή του υπολοίπου από την στοίβα των υπολοίπων.
- Εμφάνιση του υπολοίπου.

Υπολογισμός

Στοίβα Υπολοίπων

Αποτέλεσμα

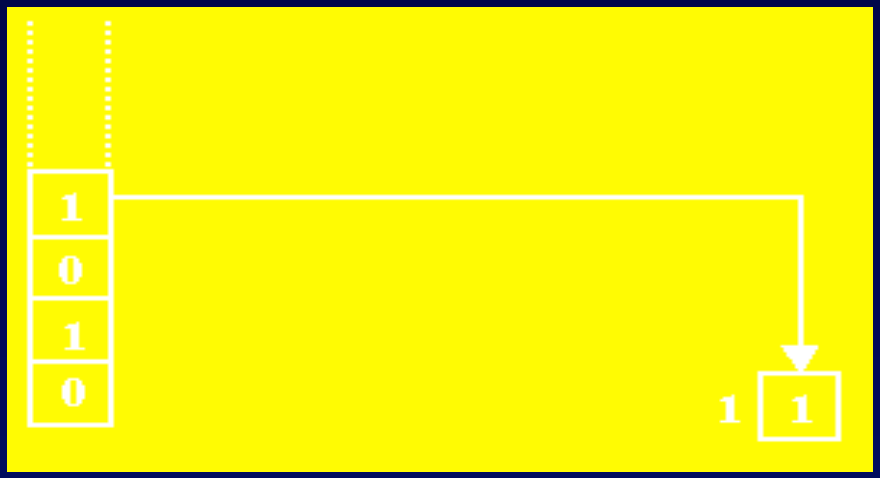


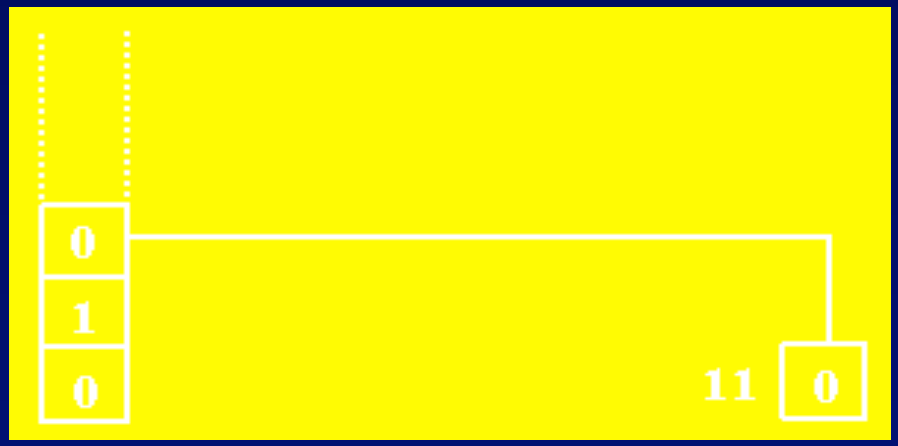

Υπολογισμός
Στοίβα Υπολοίπων
Αποτέλεσμα
3
2
1
1
1
0
1
0
1
2
1
0
1
1
0
1
0
1

Υπολογισμός

Στοίβα Υπολοίπων

Αποτέλεσμα

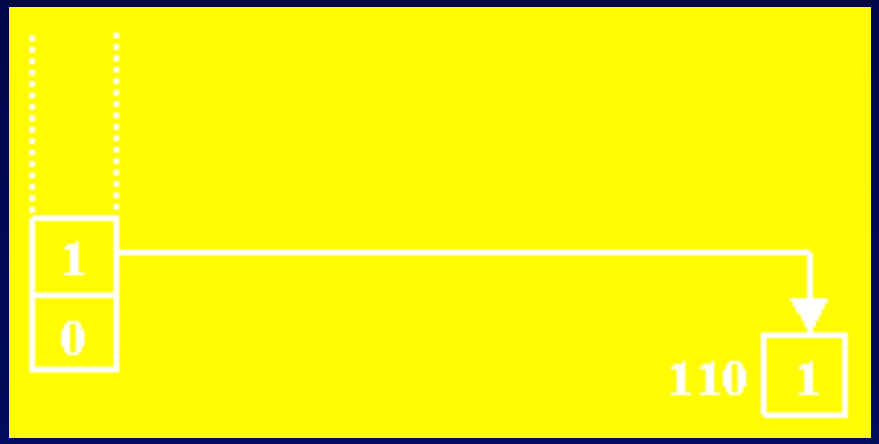


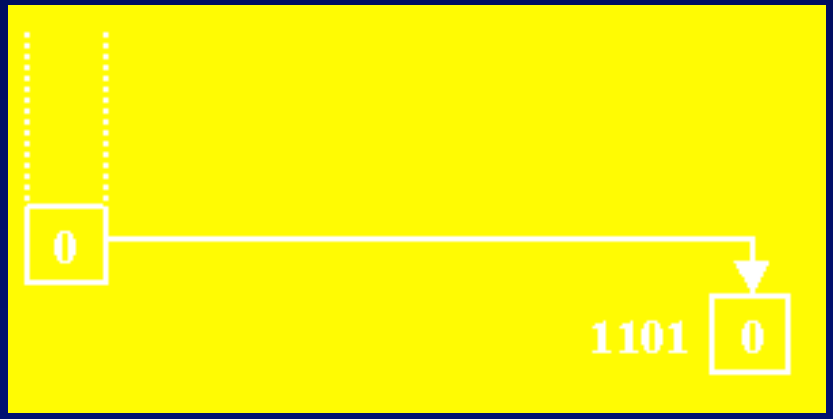


Υπολογισμός

Στοίβα Υπολοίπων

Αποτέλεσμα





Υλοποίηση του αλγορίθμου χρησιμοποιώντας τον ΑΤΔ στοίβα

```
dimiourgia(&stoiva);
while (arithmos != 0)
{
        ypoloipo = arithmos % 2;
        othisi(&stoiva, ypoloipo);
        arithmos = arithmos / 2;
}
printf (“Η παράσταση του αριθμού στο δυαδικό σύστημα είναι η :”);
while (!keni(stoiva))
{
        exagogi(&stoiva, &ypoloipo);
        print(ypoloipo);
}
```

Υλοποίηση του ΑΤΔ στοίβα με πίνακα :

```
#define plithos ...
typedef  ...  typos_stoixeiou;
typedef  typos_stoixeiou typos_pinaka[plithos];

typedef  struct
{
        int korifi;
        typos_pinaka  pinakas;
 }typos_stoivas;
```

Δημιουργία της στοίβας :

```
void dimiourgia(typos_stoivas *stoiva)
 /*Προ : Καμμία.
    Μετα : Δημιουργία κενής στοίβας*/
{
        stoiva->korifi = -1;
}
```

Ελεγχος αν μια στοίβα είναι κενή :

```
int keni(typos_stoivas stoiva)

/* Προ : Δημιουργία στοίβας.
   Μετά : Επιστρέφει 1 αν η στοίβα είναι κενή, διαφορετικά
    επιστρέφει 0.*/
{
        return (stoiva.korifi == -1);
}
```

Η λειτουργία της εξαγωγής αποτελείται από τρεις ενέργειες :

1. Αν η στοίβα είναι κενή, τότε τυπώνεται ένα μήνυμα και σταματά η εκτέλεση του προγράμματος

2. Εξάγεται το στοιχείο που δείχνει η korifi από τη στοίβα.

3. Επιστρέφει την τιμή του στοιχείου που εξάχθηκε στο κύριο πρόγραμμα.

Η υλοποίηση της εξαγωγής γίνεται από το ακόλουθο υποπρόγραμμα :

```
void exagogi(typos_stoivas *stoiva,  typos_stoixeiou *stoixeio)
/* Προ : Η στοίβα δεν είναι κενή.
   Μετά : Εξάγεται το stoixeio από τη stoiva*/
{
  if  (keni(*stoiva))
            printf(" Η στοίβα είναι κενή ");
  else
  {
    *stoixeio=stoiva->pinakas[stoiva->korifi];
     stoiva->korifi--;
   }
}
```

Μια άλλη μορφή της exagogi είναι η παρακάτω:

```
void exagogi(typos_stoivas *stoiva, typos_stoixeiou  *stoixeio, int *ypoxeilisi)
/* Προ : Μη κενή στοίβα.
   Μετα : Εξάγεται το stoixeio από τη stoiva και η τιμή
        της ypoxeilisi είναι 0, διαφορετικά είναι 1. */
{
        if (keni(*stoiva))
                *ypoxeilisi = 1;
        else
        {
                *ypoxeilisi = 0;
                *stoixeio = stoiva->pinakas[stoiva->korifi];
                stoiva->korifi = stoiva->korifi -1;
        }
}
```

συνέχεια

Στο κύριο πρόγραμμα όπου καλείται η εξαγωγή συνήθως γράφεται ο κώδικας :

```
exagogi(&stoiva, &stoixeio, &ypoxeilisi);
if (ypoxeilisi)
   /*να γίνει διορθωτική ενέργεια*/
else
   /*stoixeio είναι το στοιχείο που εξάγεται από
   την στοίβα*/
```

Η υλοποίηση της εισαγωγής είναι παρόμοια με εκείνη της εξαγωγής και γίνεται από το ακόλουθο υποπρόγραμμα :

```
void othisi(typos_stoivas *stoiva, typos_stoixeiou stoixeio)
/* Προ : Δημιουργία στοίβας.
   Μετα : Εισάγει το stoixeio στη στοίβα*/
{
        if (stoiva->korifi == plithos-1)
                printf(" Η στοίβα είναι γεμάτη ");
        else
        {
           stoiva->korifi++;
           stoiva->pinakas[stoiva->korifi]=stoixeio;
        }
}
```

Χρησιμοποιώντας αντί μηνύματος μια λογική μεταβλητή έχουμε την παρακάτω μορφή της othisi:

```
void  othisi(typos_stoivas *stoiva, typos_stoixeiou stoixeio, int *yperheilisi);
/*Αν η στοίβα είναι πλήρης, τότε η τιμή της yperheilisi είναι 1, διαφορετικά είναι
0*/
{
        if (stoiva->korifi == plithos-1)
                *yperheilisi = 1;
        else
        {
                *yperheilisi = 0;
                stoiva->korifi++;
                stoiva->pinakas[stoiva->korifi]=stoixeio;
        }
}
```

Εφαρμογές με τη χρήση στοίβας

Μετατροπή ακεραίου από το δεκαδικό στο δυαδικό σύστημα

Εχοντας υλοποιήσει τις βασικές πράξεις της στοίβας μπορούμε να προχωρήσουμε στην κωδικοποίηση του αλγορίθμου για την μετατροπή ενός θετικού ακεραίου από το δεκαδικό στο δυαδικό σύστημα.

```
#include <stdio.h>
#define plithos 100
typedef  int  typos_stoixeiou;
typedef  typos_stoixeiou  typos_pinaka[plithos];
typedef  struct {
      int korifi;
      typos_pinaka pinakas;
 }typos_stoivas;
```

συνέχεια

```
void dimiourgia(typos_stoivas *stoiva);
int keni(typos_stoivas stoiva);
int exagogi(typos_stoivas *stoiva, typos_stoixeiou *stoixeio);
int othisi(typos_stoivas *stoiva, typos_stoixeiou stoixeio);

main( )
{
        int arithmos;
        int ypoloipo;
        typos_stoivas stoiva;
        printf("Δώσε τον αριθμό:");
        scanf("%d",&arithmos);
        dimiourgia(&stoiva);
```

συνέχεια

```
        while (arithmos !=0)
        {       ypoloipo=arithmos % 2;
                othisi(&stoiva,ypoloipo);
                arithmos=arithmos / 2;
        }
        printf(“Ο αριθμός στο δυαδικό σύστημα είναι:\n");

        while ( !keni(stoiva) )
        {
                exagogi(&stoiva,&ypoloipo);
                printf("%d",ypoloipo);
        }
        printf("\n\n");
}
```

Υπολογισμός Αριθμητικών Παραστάσεων - Πολωνικός Συμβολισμός

A + (B * C) Ενδοθεματική μορφή
A + (BC *)
A(BC *) +
ABC * + Μεταθεματική μορφή

Οι κανόνες που διέπουν τη μετατροπή μιας αριθμητικής παράστασης από την ενδοθεματική στη μεταθεματική μορφή είναι :

1) Οι πράξεις που έχουν τη μεγαλύτερη προτεραιότητα πρέπει να μετατραπούν πρώτες
2) Μόλις μετατραπεί ένα τμήμα της παράστασης σε μεταθεματική μορφή θα πρέπει το τμήμα αυτό να εκλαμβάνεται σαν ένας όρος.

Παράδειγμα

Να μετατραπεί η παράσταση:

A ^ B * C - D + E / F / (G + H)

A ^ B * C - D + E / F / (GH +)
(AB ^) * C - D + E / F / (GH +)
(AB ^ C *) - D + E / F / (GH +)
(AB ^ C *) - D + (EF /) / (GH +)
(AB ^ C *) - D + (EF / GH + /)
(AB ^ C * D -) + (EF / GH + /)
AB ^ C * D - EF / GH + / + μεταθεματική μορφή

Περιττές οι παρενθέσεις στη Μεταθεματική μορφή

Ας υποθέσουμε ότι έχουμε την ενδοθεματική μορφή:

(1 + 5) * (8 - (4 - 1))

η οποία έχει τη μεταθεματική μορφή:

15 + 841 - - *

*15 + 841 - - **

Αντικαθιστώντας την υπογραμμισμένη παράσταση με την τιμή του αποτελέσματος 1 + 5, δηλαδή με το 6 έχουμε την

*6 8 4 1 - - **

*6 8 4 1 - - **

Αντικαθιστώντας την υπογραμμισμένη παράσταση με την τιμή της 4 - 1 έχουμε

*6 8 3 - **

Συνεχίζοντας λαμβάνουμε διαδοχικά:

*6 8 3 - **

*6 5 **

30 *τιμή της παράστασης.*

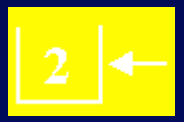
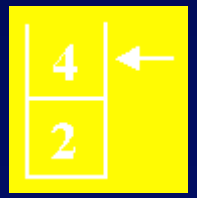
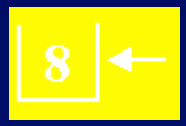
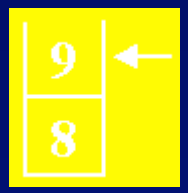
Αλγόριθμος για τον υπολογισμό της μεταθεματικής μορφής

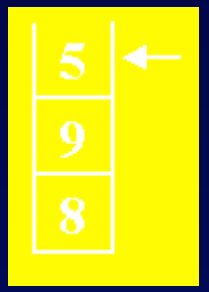
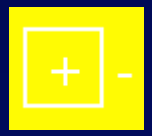
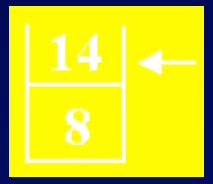
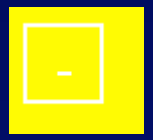
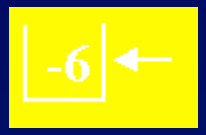
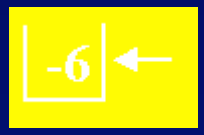
1. Να δημιουργηθεί μια στοίβα.
2. Οσο υπάρχουν σύμβολα στην είσοδο να εκτελούνται οι παρακάτω εργασίες:
   a. Πάρε το επόμενο σύμβολο (σταθερά, μεταβλητή, αριθμητικός τελεστής) της μεταθεματικής μορφής.
   b. Αν είναι όρος (δηλ. σταθερά ή μεταβλητή), τότε να εισαχθεί στη στοίβα διαφορετικά /*τελεστής*/
      (i) Βγάλε πρώτα δύο στοιχεία - όρους της στοίβας (Αν η στοίβα δεν περιέχει δύο στοιχεία - όρους, τότε η μεταθεματική παράσταση είναι λάθος και ο υπολογισμός της σταματά).
      (ii) Εφάρμοσε τον τελεστή για τους δύο όρους.
      (iii) Τοποθέτησε το αποτέλεσμα στη στοίβα.
3. Το αποτέλεσμα του υπολογισμού της μεταθεματικής μορφής βρίσκεται στην κορυφή της στοίβας.

Εφαρμογή του Αλγορίθμου

Ενδοθεματική : 2 * 4 - (9 + 5)

Μεταθεματική : 2 4 * 9 5 + -

Παράσταση	Στοίβα	Σχόλια
		Τοποθέτηση του 2 στη στοίβα
		Τοποθέτηση του 4 στη στοίβα
		Εξαγωγή των 4 και 2, υπολογισμός της 4*2=8 και τοποθέτηση του 8 στη στοίβα
		Τοποθέτηση του 9 στη στοίβα

Παράσταση	Στοίβα	Σχόλια
		Τοποθέτηση του 5 στη στοίβα
		Εξαγωγή των 5 και 9, υπολογισμός της 9+5=14 και τοποθέτηση του 14 στη στοίβα
		Εξαγωγή των 14 και 8, υπολογισμός της 8-14=-6 και τοποθέτηση του -6 στη στοίβα
		Η τιμή της παράστασης βρίσκεται στη κορυφή της στοίβας

Πρόγραμμα για τον υπολογισμό της τιμής μιας μεταθεματικής παράστασης (υποθέτουμε ότι η είσοδος περιέχει μια σειρά από θετικούς μονοψήφιους ακέραιους και αριθμητικούς τελεστές μόνο)

```
#include <stdio.h>
#include <stdlib.h>

#define plithos 80
typedef float typos_stoixeiou;
typedef typos_stoixeiou typos_pinaka[plithos];
typedef struct {
        int korifi;
        typos_pinaka pinakas;
        }typos_stoivas;
```

συνέχεια

```
float telestis(char symbolo, float oros1, float oros2);
void dimiourgia(typos_stoivas *stoiva);
int keni(typos_stoivas stoiva);
void exagogi(typos_stoivas *stoiva, typos_stoixeiou *stoixeio);
void othisi(typos_stoivas *stoiva, typos_stoixeiou stoixeio);
main( )
{
        typos_stoivas stoiva;
        float oros1, oros2, apotelesma;
        char symbolo, apantisi, metathematiki[256];
        int i;
```

συνέχεια

```
do {
    dimiourgia(&stoiva);
    i=0;
    printf("Δώσε τη παράσταση:");
    while ( (symbolo=getchar( )) !=  '\n' )
    {
        metathematiki[i]=symbolo;
        i++;
        if (symbolo != ' ' ) /*αγνοεί τα κενά*/
              if (symbolo >= '0' && symbolo <= '9')
              {   apotelesma= (float) symbolo - '0';
                  othisi(&stoiva, apotelesma);
              }
```

συνέχεια

```
                else   /*  είναι τελεστής */
                {
                    exagogi(&stoiva,&oros2);
                    exagogi(&stoiva,&oros1);
                    apotelesma = telestis(symbolo, oros1, oros2);
                    othisi(&stoiva, apotelesma);
                }
        } /* while ..!= ‘\n’ */
    metathematiki[i]='\0';
    exagogi(&stoiva, &apotelesma);
    printf("%s=5.2f\n",metathematiki, apotelesma);
    printf(“Θέλεις να συνεχίσεις; (Y/N)\n");
    scanf ("%c",&apantisi);
    } while (apantisi == 'N' || apantisi == 'n');
}
```

συνέχεια

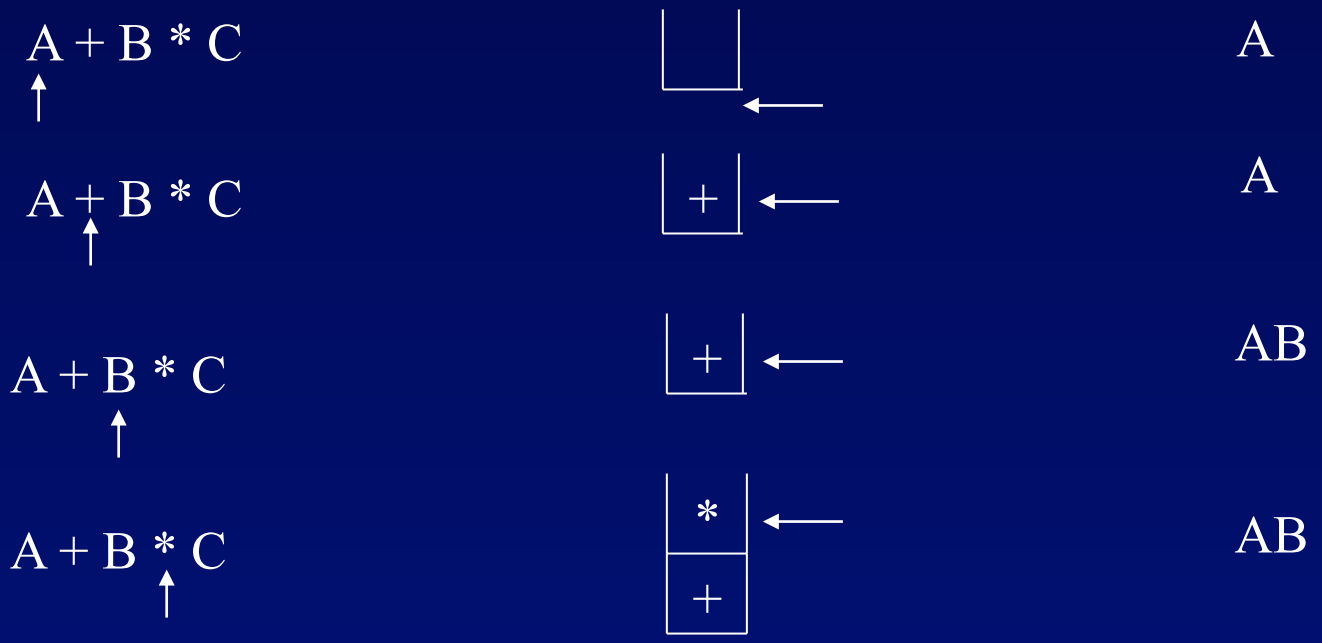
```
float telestis (char symbolo, float oros1,oros2)
{
        switch (symbolo)
        {
                case '+' :      return  (oros1+oros2);
                                break;
                case '-' :      return (oros1-oros2);
                                break;
                case '*' :      return (oros1*oros2);
                                break;
                case '/' :      return (oros1/oros2);
                                break;
                default  :      printf ("Λάθος!\n");
                                return (-1.0);
        }
}
```

Μετατροπή Ενδοθεματικής στη Μεταθεματική μορφή

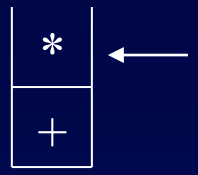
A + B * C ⟶ A B C * +

Οι όροι τυπώνονται

Οι τελεστές αποθηκεύονται



A + B * C



ABC

A + B * C

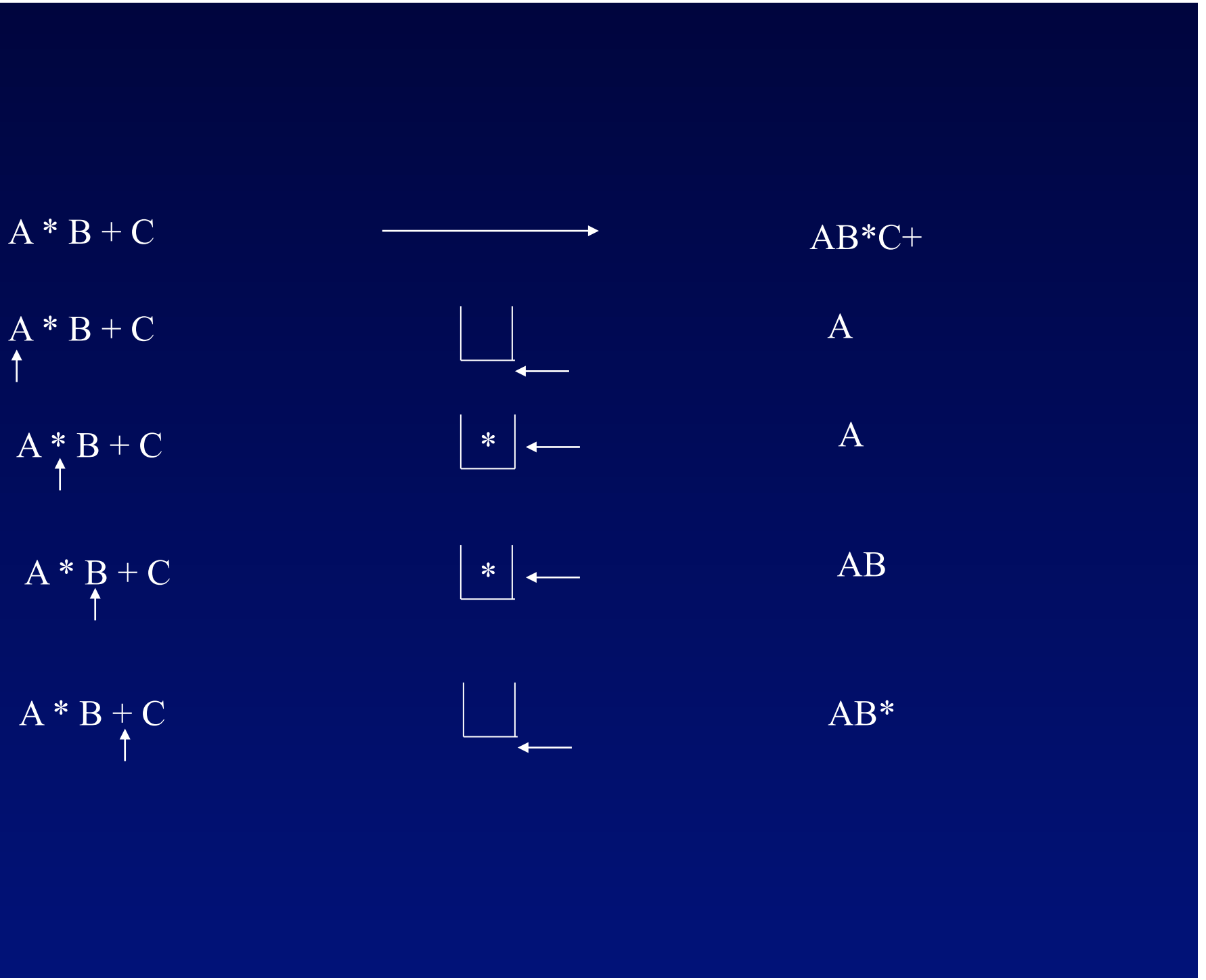
ABC*+

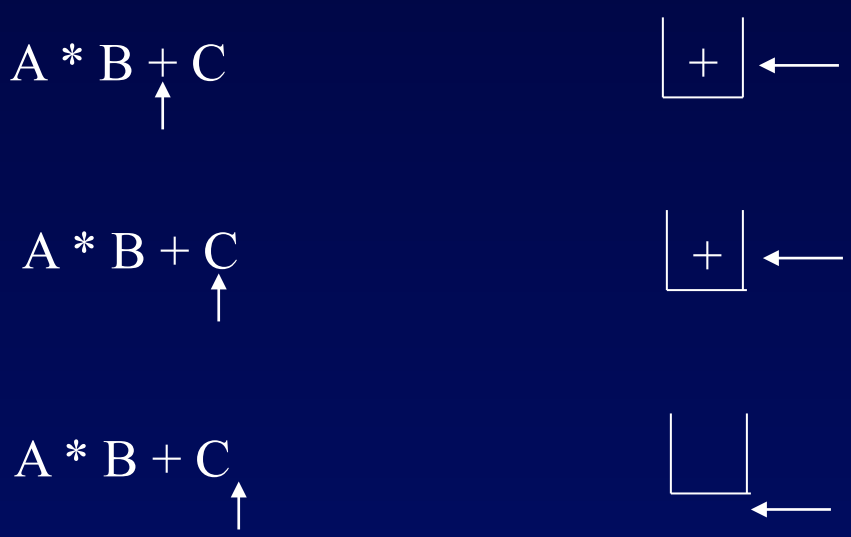
Αν προ(τελεστή) > προ(τελεστή στοίβας) τότε

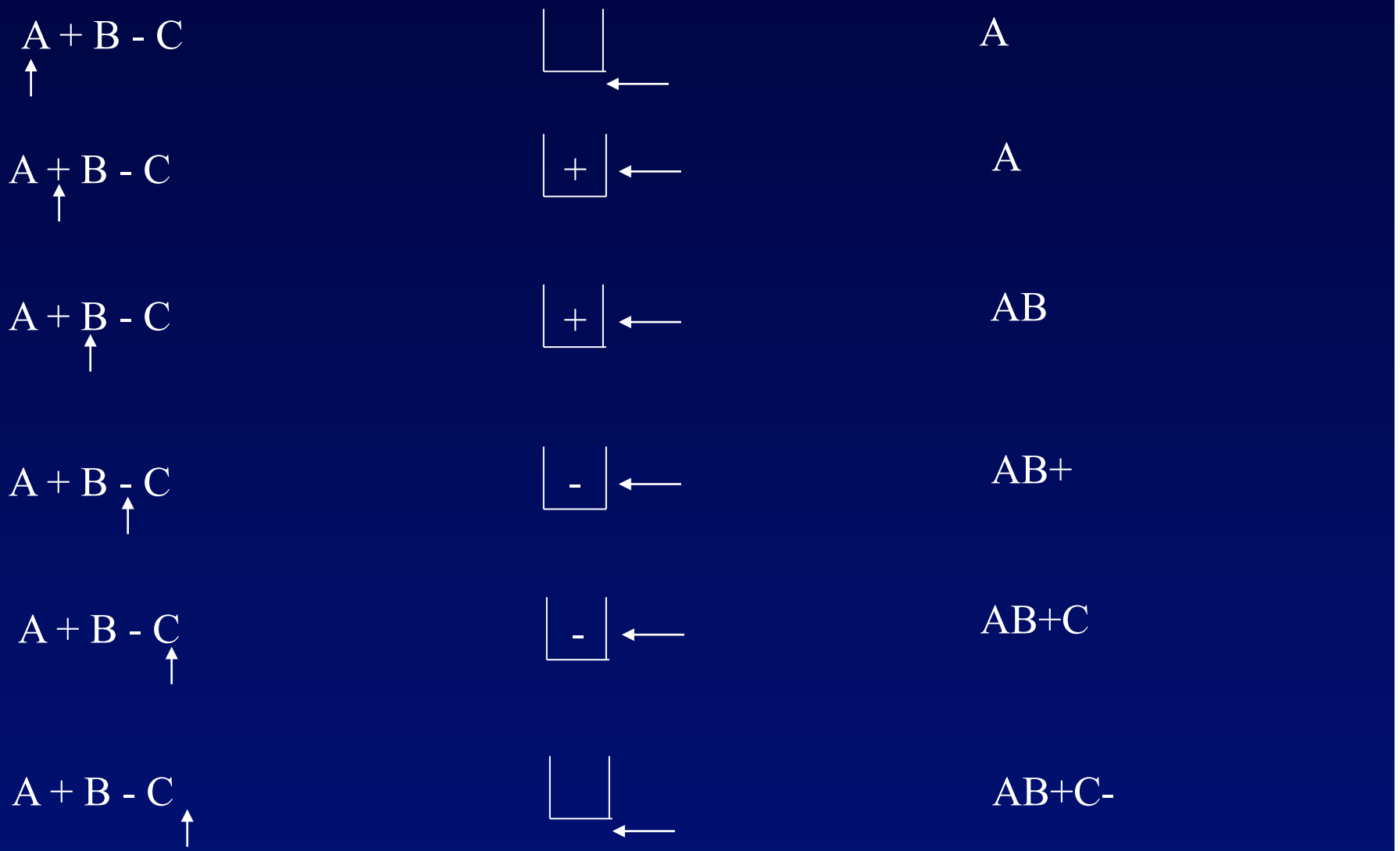
ώθηση του τελεστή στη στοίβα

διαφορετικά

εξαγωγή του τελεστή από τη στοίβα


A * B + C
AB*C+
A * B + C
A
A * B + C
*
A
A * B + C
*
AB
A * B + C
AB*


A * B + C
+
AB*
A * B + C
+
AB*C
A * B + C
AB*C+


A + B - C
A
A + B - C
+
A
A + B - C
+
AB
A + B - C
-
AB+
A + B - C
-
AB+C
A + B - C
AB+C-

Αλγόριθμος για την μετατροπή της ενδοθεματικής παράστασης σε μεταθεματική

1. Δημιουργία μιας κενής στοίβας τελεστών.
2. Οσο δεν έχει συναντηθεί το τέλος της ενδοθεματικής μορφής και δεν υπάρχει λάθος, να εκτελούνται οι παρακάτω εργασίες :
   (a) Πάρε το επόμενο σύμβολο (σταθερά, μεταβλητή, αριθμητικός τελεστής, αριστερή ή δεξιά παρένθεση) της ενδοθεματικής μορφής.
   (b) Αν το σύμβολο αυτό είναι:
      (i) μια αριστερή παρένθεση : τότε να τοποθετηθεί στη στοίβα.
      (ii) μια δεξιά παρένθεση: τότε να εξαχθούν από τη στοίβα όλοι οι τελεστές μέχρις ότου στην κορυφή της στοίβας βρεθεί μια αριστερή παρένθεση. (Σε περίπτωση που η στοίβα γίνει κενή χωρίς της ανεύρεση αριστερής παρένθεσης, τότε η ενδοθεματική μορφή είναι λανθασμένη).

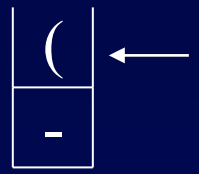
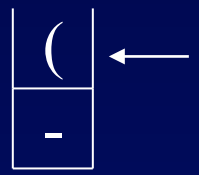
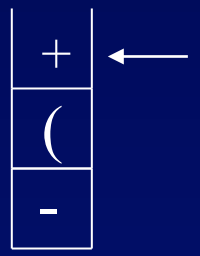
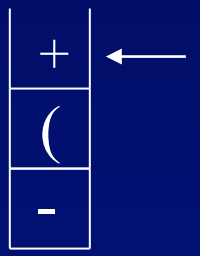
(iii) ένας τελεστής: Αν η στοίβα είναι κενή ή ο τελεστής έχει μεγαλύτερη προτεραιότητα από τον τελεστή που βρίσκεται στην κορυφή της στοίβας, τότε τοποθετείται στη στοίβα. Διαφορετικά, εξάγεται και τοποθετείται στη μεταθεματική σειρά ο τελεστής που βρίσκεται στην κορυφή της στοίβας. Η εργασία αυτή επαναλαμβάνεται με τον επόμενο τελεστή που βρίσκεται στην κορυφή της στοίβας. Ας σημειωθεί ότι μια αριστερή παρένθεση που βρίσκεται στη στοίβα υποτίθεται ότι έχει μικρότερη προτεραιότητα από εκείνη του τελεστή.

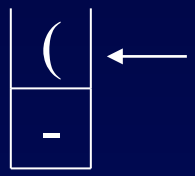
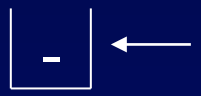
(iv) ένας όρος: τότε να προστεθεί στη μεταθεματική σειρά (να εκτυπωθεί).

3. Οταν συναντηθεί το τέλος της ενδοθεματικής μορφής, τότε να εξαχθούν και να τοποθετηθούν με τη σειρά όλα τα στοιχεία της στοίβας μέχρις ότου γίνει κενή.

Στο ακόλουθο σχήμα εμφανίζεται η εφαρμογή του αλγόριθμου για την έκφραση 7 * 8 - (2 + 3)

Παράσταση	Στοίβα	Εκτύπωση	Σχόλια
7 *8-(2+3)	κενή	7	Εκτύπωση του 7
* 8-(2+3)	* ←	7	ώθηση του *
8 -(2+3)	* ←	78	Εκτύπωση του 8
- (2+3)		78*	Εξαγωγή και εκτύπωση του *
(2+3)	- ←	78*	ώθηση του -

Παράσταση	Στοίβα	Εκτύπωση	Σχόλια
(2+3)		78*	ώθηση της (
2 +3)		78*2	Εκτύπωση του 2
+ 3)		78*2	ώθηση του +
3)		78*23	Εκτύπωση του 3

Παράσταση	Στοίβα	Εκτύπωση	Σχόλια
)		78*23+	Εξαγωγή και εκτύπωση του +
		78*23+	Εξαγωγή της (
τέλος της παράστασης		78*23+-	Εξαγωγή και εμφάνιση του -

Στη συνέχεια ακολουθεί το πρόγραμμα για τη μετατροπή μιας ενδοθεματικής παράστασης στην αντίστοιχη μεταθεματική.

```
#include <stdio.h>
#include <stdlib.h>

#define plithos 80
#define mikos 80
#define telos ';' /* ένδειξη τέλους ενδοθεματικής παράστασης */
#define synolo_teleston "+-*/" /* σύνολο τελεστών */
typedef char typos_stoixeiou;
typedef typos_stoixeiou  typos_pinaka[plithos];
typedef struct {
                    int korifi;
                    typos_pinaka pinakas;
                } typos_stoivas;
typedef typos_stoixeiou typos_parastasis[mikos];
```

συνέχεια

```
void dimiourgia(typos_stoivas *stoiva);
int keni(typos_stoivas stoiva);
int exagogi(typos_stoivas *stoiva,  typos_stoixeiou *stoixeio);
int othisi(typos_stoivas *stoiva, typos_stoixeiou stoixeio);
int protereotita(char telestis);
void metatropi (typos_parastasis endo_parastasi);
```

συνέχεια

```
main( ){
        typos_parastasis endo_parastasi;
        int i;
        fflush(stdin);
        printf(“Δώσε την παράσταση\n”);
        i=-1;
        do
        {
                i++;
                endo_parastasi[i] = getchar( );
        }  while (endo_parastasi[i]!=telos);
        i++;
        endo_parastasi[i]='\0';
        printf (“Μετατροπή της παράστασης:");
        metatropi(endo_parastasi);
        printf ("\n");
}
```

συνέχεια

```
int protereotita(char telestis)
/* Επιστρέφει την προταιρεότητα του τελεστή telestis.. */
{
        switch (telestis)
        {
                case '(':       return 0;
                case '+': case '-': return 1;
                case '*': case '/': return 2;
        }
}
```

συνέχεια

```
void metatropi (typos_parastasis endo_parastasi)
/* Αυτή η υπορουτίνα μετατρέπει την ενδοθεματική μορφή μιας
        παράστασης στην μεταθεματική της μορφή */
 {
        typos_stoivas stoiva;
        int i;
        char symbolo, symbolo_korifi ;
        int lathos, telos_exagogis;

        dimiourgia (&stoiva);
        lathos = 0;
        /* αρχή της μετατροπής */
        i = 0;
        symbolo = endo_parastasi[0];
```

συνέχεια

```
while ((symbolo!=telos) && ( ! lathos ))
{
        while ( endo_parastasi[i] == ' ' )  /*Αγνόηση κενών*/
                i++;
        symbolo = endo_parastasi[i];
        if (  symbolo == '('  )  /* αριστερή παρένθεση */
                othisi(&stoiva, symbolo);
        else
                if (  symbolo == ')'  ) /* δεξιά παρένθεση*/
                {
                        telos_exagogis = 0;
```

συνέχεια

```
do {
        if (keni(stoiva))
                lathos = 1;
        else {
                exagogi( &stoiva, &symbolo_korifi );
                if ( symbolo_korifi != '('  )
                        printf("%2c", symbolo_korifi);
                else
                        telos_exagogis = 1;
                }
        } while ( ( ! telos_exagogis ) &&  ( ! lathos ) );
} /* δεξιά παρένθεση*/
```

συνέχεια

```
else
   if ( strchr( synolo_teleston, symbolo ) != NULL ) /* τελεστής */
  {
      telos_exagogis = 0;
      while ( (! keni(stoiva)) && (! telos_exagogis) )
      {
            exagogi( &stoiva, &symbolo_korifi );
            if (((protereotita( symbolo))<=(protereotita(     symbolo_korifi)))
                        printf("%2c", symbolo_korifi);
            else
           {            othisi( &stoiva, symbolo_korifi );
            telos_exagogis = 1;}
            }
       }
       othisi(&stoiva, symbolo);
  }
```

συνέχεια

```
        else  /* όρος*/
              if (symbolo != telos)
                    printf("%2c", symbolo);
        i++;
    }   /* main while */
    /* Εξαγωγή και εκτύπωση των στοιχείων της στοίβας. */
    while ( ( ! keni(stoiva) ) && ( ! lathos ) )
    {
        exagogi(&stoiva, &symbolo_korifi);
        if (symbolo_korifi != '(')
                printf("%2c", symbolo_korifi);
        else lathos = 1;
     }
     if ( lathos )
        printf ("Λάθος στην ενδοθεματική παράσταση.\n");
     else
        printf ("\n");
}
```