

ΣΥΝΟΛΑ

1.2 Υλοποίηση του ΑΤΔ Σύνολο με Πίνακα

Στην παρούσα ενότητα θα παρουσιάσουμε τον τρόπο **υλοποίησης του ΑΤΔ σύνολο στη C** χρησιμοποιώντας το **τύπο δεδομένων πίνακα**.

Συγκεκριμένα, θα χρησιμοποιήσουμε ένα λογικό πίνακα S , όπου $S[i]$ είναι TRUE αν το στοιχείο που αντιστοιχεί στο i -οστό στοιχείο του καθολικού συνόλου ανήκει στο σύνολο S , διαφορετικά είναι FALSE.

Παράδειγμα

θέση	τιμή
0	FALSE
1	TRUE
2	FALSE
3	TRUE
4	FALSE
5	TRUE
6	FALSE
7	TRUE
8	FALSE
9	TRUE

Αν θεωρήσουμε το σύνολο

OddNumbers = {1, 3, 5, 7, 9},

όπου το καθολικό σύνολο είναι τα ψηφία 0..9, τότε αυτό μπορεί να παρασταθεί με τον πίνακα 10 θέσεων που παρουσιάζεται αριστερά.

Η 1^η θέση του πίνακα αντιστοιχεί στο 1^ο στοιχείο του καθολικού συνόλου, δηλαδή στο ψηφίο 0, η 2^η στο ψηφίο 1 κ.τ.λ.

Για την υλοποίηση του ΑΤΔ σύνολο με πίνακα χρησιμοποιούνται οι παρακάτω δηλώσεις:

```
#define megisto_plithos 10          /* μέγιστο πλήθος στοιχείων συνόλου */
typedef enum {
    FALSE, TRUE
} boolean;

typedef boolean typos_synolou[megisto_plithos];
typedef int stoixeio_synolou;
```

Οι βασικές πράξεις/λειτουργίες που συνδέονται με τα σύνολα υλοποιούνται εύκολα χρησιμοποιώντας τη δομή δεδομένων του πίνακα. Στη συνέχεια περιγράφεται ο τρόπος υλοποίησής τους, θεωρώντας ότι οι μεταβλητές `synolo`, `s1`, `s2`, `enosi`, `tomι` και `diafora` είναι μεταβλητές πίνακα τύπου `typos_synolou`:

Δημιουργία ενός κενού συνόλου (Dimiourgia): για τη δημιουργία του κενού συνόλου, δηλαδή ενός συνόλου που δεν έχει καθόλου στοιχεία, εκχωρείται σε όλες τις θέσεις του πίνακα `synolo` η τιμή `FALSE`.

Δημιουργία καθολικού συνόλου (Katholiko): για τη δημιουργία του καθολικού συνόλου, δηλαδή του συνόλου που περιλαμβάνει όλα τα στοιχεία του συγκεκριμένου τύπου βάσης που έχει δηλωθεί, εκχωρείται σε όλες τις θέσεις του πίνακα `synolo` η τιμή `TRUE`.

Εισαγωγή στοιχείου (Eisagogi): για την εισαγωγή, σε ένα σύνολο, του στοιχείου που βρίσκεται στην *i*-οστή θέση του καθολικού συνόλου απλά εκχωρούμε στην *i*-οστή θέση του πίνακα `synolo` την τιμή `TRUE`.

Διαγραφή στοιχείου (Diagrafi): για την διαγραφή ενός στοιχείου από ένα σύνολο εκχωρούμε στην θέση του πίνακα synolo που βρίσκεται το στοιχείο την τιμή FALSE.

Μέλος (Melos): για να διαπιστώσουμε αν ένα στοιχείο είναι μέλος ενός συνόλου, ελέγχουμε την τιμή που υπάρχει στην θέση του πίνακα synolo στην οποία αντιστοιχεί το στοιχείο. Αν η τιμή της συγκεκριμένης θέσης του πίνακα είναι TRUE τότε το στοιχείο είναι μέλος του συνόλου, διαφορετικά όχι.

Κενό (KenoSynolo): για να ελέγξουμε αν ένα σύνολο είναι κενό εξετάζουμε τα στοιχεία του πίνακα synolo μέχρι να βρούμε:

1. ότι κάποιο στοιχείο έχει τιμή TRUE, γεγονός που σημαίνει ότι το σύνολο δεν είναι κενό, ή
2. να εξαντληθούν όλα τα στοιχεία του πίνακα, γεγονός που σημαίνει ότι το σύνολο είναι κενό.

Ίσα (IsaSynola): για να ελέγξουμε αν δύο σύνολα $s1$ και $s2$ είναι ίσα, συγκρίνουμε τα αντίστοιχα στοιχεία τους μέχρι να βρούμε:

(1) ότι κάποιο στοιχείο είναι μέλος ενός μόνο εκ των $s1$ και $s2$, γεγονός που σημαίνει ότι τα 2 σύνολα δεν είναι ίσα, ή

(2) να εξαντληθούν όλα τα στοιχεία του πίνακα, γεγονός που σημαίνει ότι τα σύνολα είναι ίσα.

Υποσύνολο (Yprosynolo): για να ελέγξουμε αν ένα σύνολο $s1$ είναι υποσύνολο του $s2$, εξετάζουμε όλα τα στοιχεία τους μέχρι να βρούμε:

(1) ότι κάποιο στοιχείο που είναι μέλος του $s1$ δεν είναι μέλος του $s2$, γεγονός που σημαίνει ότι το $s1$ δεν είναι υποσύνολο του $s2$, ή

(2) να εξαντληθούν όλα τα στοιχεία, γεγονός που σημαίνει ότι το $s1$ είναι υποσύνολο του $s2$.

Ένωση (EnosiSynolou): για να βρούμε την ένωση δύο συνόλων $s1$ και $s2$ εξετάζουμε τα αντίστοιχα στοιχεία των $s1$ και $s2$ (δηλαδή τα στοιχεία που βρίσκονται στις ίδιες θέσεις των πινάκων αυτών) και αν ένα τουλάχιστον από αυτά έχει την τιμή TRUE τότε εκχωρούμε την τιμή TRUE και στην αντίστοιχη θέση του συνόλου της τομής, έστω enosi, διαφορετικά εκχωρούμε την τιμή FALSE.

Τομή (TomiSynolou): για να βρούμε την τομή δύο συνόλων $s1$ και $s2$, εξετάζουμε τα αντίστοιχα στοιχεία των $s1$ και $s2$ και αν και τα 2 έχουν την τιμή TRUE τότε εκχωρούμε την τιμή TRUE και στην αντίστοιχη θέση του συνόλου της τομής, έστω tomi, διαφορετικά εκχωρούμε την τιμή FALSE.

Διαφορά (DiaforaSynolou): για να βρούμε τη διαφορά $s1 - s2$ δύο συνόλων, εξετάζουμε τα αντίστοιχα στοιχεία των $s1$ και $s2$ και για κάθε στοιχείο που είναι μέλος του $s1$ και δεν είναι μέλος του $s2$ εκχωρούμε την τιμή TRUE στην αντίστοιχη θέση του συνόλου της διαφοράς, έστω diafora, και σε κάθε άλλη περίπτωση την τιμή FALSE.

Πακέτο για τον ΑΤΔ ΣΥΝΟΛΟ με πίνακα

```
// Filename: SetADT.h
#define megisto_plithos 10      / *μέγιστο πλήθος στοιχείων συνόλου*/
typedef enum {
    FALSE, TRUE
} boolean;
typedef boolean typos_synolou[megisto_plithos];
typedef int stoixeio_synolou;
void Dimiourgia(typos_synolou synolo);
void Katholiko(typos_synolou synolo);
void Eisagogi(stoixeio_synolou stoixeio, typos_synolou synolo);
void Diagrafi(stoixeio_synolou stoixeio, typos_synolou synolo);
boolean Melos(stoixeio_synolou stoixeio, typos_synolou synolo);
boolean KenoSynolo(typos_synolou synolo);
boolean IsaSynola(typos_synolou s1, typos_synolou s2);
boolean Yposynolo(typos_synolou s1, typos_synolou s2);
void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi);
void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi);
void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora);
```


Δημιουργία Κενού Συνόλου

S1

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	FALSE
4	FALSE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

Δημιουργία Κενού Συνόλου

S1

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	FALSE
4	FALSE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

```
void Dimiourgia(typos_synolou synolo)
{
    στοιχειο_synolou i;

    for (i = 0; i < megisto_plithos; i++)
        synolo[i] = FALSE;
}
```

Πακέτο για τον ΑΤΔ ΣΥΝΟΛΟ με πίνακα

```
// * Filename: SetADT.c
```

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

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

```
#include "SetADT.h"
```

```
void Dimiourgia(typos_synολου synolo)
```

```
/* Λειτουργία:      Δημιουργεί ένα σύνολο χωρίς στοιχεία, δηλαδή το κενό σύνολο.
```

```
Επιστρέφει:      Το κενό σύνολο.*/*
```

```
{  
    στοιχείο_synολου i;
```

```
    for (i = 0; i < megisto_plithos; i++)  
        synolo[i] = FALSE;
```

```
}
```

Καθολικό Σύνολο

θέση	τιμή
0	TRUE
1	TRUE
2	TRUE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	TRUE
8	TRUE
9	TRUE

Καθολικό Σύνολο

θέση	τιμή
0	TRUE
1	TRUE
2	TRUE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	TRUE
8	TRUE
9	TRUE

```
void Katholiko(typos_synolou synolo)
{
    στοιχειο_synolou i;

    for (i = 0; i < megisto_plithos; i++)
        synolo[i] = TRUE;
}
```

void Katholiko(typos_synolou synolo)

*/*Δέχεται: Ένα σύνολο.*

Λειτουργία: Δημιουργεί ένα σύνολο με όλα τα στοιχεία παρόντα, έτσι όπως ορίστηκε στο τμήμα δηλώσεων του προγράμματος.

Επιστρέφει: Το καθολικό σύνολο που δημιουργήθηκε./**

```
{  
    στοιχείο_synολου i;  
  
    for (i = 0; i < megisto_plithos; i++)  
        synolo[i] = TRUE;  
}
```

Εισαγωγή σε Σύνολο

S1

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	FALSE
4	FALSE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

Eisagogi(2, S1)

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

Eisagogi(4, S1)

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

Εισαγωγή σε Σύνολο

S1

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	FALSE
4	FALSE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

Eisagogi(2, S1)

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

Eisagogi(4, S1)

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

```
void Eisagogi(stoixeio_synolou stoixeio, typos_synolou synolo)
{
    synolo[stoixeio] = TRUE;
}
```


Διαγραφή από Σύνολο

$S1 = \{2, 4, 6, 8\}$

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

Diagrafi(4, S1)

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

Diagrafi(8, S1)

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	FALSE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

Διαγραφή από Σύνολο

S1={2, 4, 6, 8}

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

Diagrafi(4, S1)

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

Diagrafi(8, S1)

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	FALSE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

```
void Diagrafi(stoixeio_synolou stoixeio, typos_synolou synolo)
```

```
{  
    synolo[stoixeio] =FALSE;  
}
```

void Eisagogi(stoixeio_synolou stoixeio, typos_synolou synolo)

/*Δέχεται: Ένα σύνολο και ένα στοιχείο.
Λειτουργία: Εισάγει το στοιχείο στο σύνολο.
Επιστρέφει: Το τροποποιημένο σύνολο.*/

```
{  
    synolo[stoixeio] = TRUE;  
}
```

void Diagrafi(stoixeio_synolou stoixeio, typos_synolou synolo)

/*Δέχεται: Ένα σύνολο και ένα στοιχείο.
Λειτουργία: Διαγράφει το στοιχείο από το σύνολο.
Επιστρέφει: Το τροποποιημένο σύνολο.*/

```
{  
    synolo[stoixeio] =FALSE;  
}
```

Μέλος Συνόλου

$S1 = \{2, 4, 6, 8\}$

$2 \in S1$

Melos(2,S1): TRUE

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

$5 \notin S1$

Melos(5,S1): FALSE

$S1 = \{2, 4, 6, 8\}$

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

Μέλος Συνόλου

$S1 = \{2, 4, 6, 8\}$

$2 \in S1$

Melos(2,S1): TRUE

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

$5 \notin S1$

Melos(5,S1): FALSE

$S1 = \{2, 4, 6, 8\}$

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

boolean Melos(stoixeio_synolou stoixeio, typos_synolou synolo)

```
{  
    return synolo[stoixeio];  
}
```

boolean Melos(stoicheio_synολου stoicheio, typos_synολου synolo)

*/*Δέχεται: Ένα σύνολο και ένα στοιχείο.*

Λειτουργία: Ελέγχει αν το στοιχείο είναι μέλος του συνόλου.

Επιστρέφει: Επιστρέφει TRUE αν το στοιχείο είναι μέλος του και FALSE διαφορετικά./**

```
{  
    return synolo[stoicheio];  
}
```

Πακέτο για τον ΑΤΔ ΣΥΝΟΛΟ με πίνακα

boolean KenoSynolo(typos_synολου synolo)

*/*Δέχεται: Ένα σύνολο.*

Λειτουργία: Ελέγχει αν το σύνολο είναι κενό.

Επιστρέφει: Επιστρέφει TRUE αν το σύνολο είναι κενό και FALSE διαφορετικά./*

```
{  
    στοιχείο_synολου i;  
    boolean keno;  
  
    keno = TRUE;  
    i = 0;  
    while (i < megisto_plithos && keno) {  
        if (Melos(i, synolo))  
            keno = FALSE;  
        else  
            i++;  
    }  
    return keno;  
}
```

Πακέτο για τον ΑΤΔ ΣΥΝΟΛΟ με πίνακα

boolean IsaSynola(**typos_synolou** s1, **typos_synolou** s2)

/*Δέχεται: Δύο σύνολα s1 και s2.

Λειτουργία: Ελέγχει αν τα δύο σύνολα είναι ίσα.

Επιστρέφει: Επιστρέφει TRUE αν τα δύο σύνολα έχουν τα ίδια στοιχεία και FALSE διαφορετικά.*/

```
{  
    στοιχείο_synολου i;  
    boolean isa;  
    isa = TRUE;  
    i = 0;  
    while ((i < megisto_plithos) && isa) {  
        if (Melos(i,s1) != Melos(i,s2))  
            isa = FALSE;  
        else  
            i++;  
    }  
    return isa;  
}
```


Υposynolo(S2,S1): TRUE

$S1=\{2, 4, 6, 8\}$

$S2 \subseteq S1$

$S2=\{2, 4\}$

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE



Υposynolo(S2,S1): FALSE

$S1=\{2, 4, 6, 8\}$

$S2 \not\subseteq S1$

$S2=\{2, 5\}$

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	TRUE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

Υποσύνολο

```
boolean Υposynolo(typos_synolou s1, typos_synolou s2) {
```

```
    στοιχείο_synολου i;
```

```
    boolean yposyn;
```

```
    yposyn = TRUE;
```

```
    i = 0;
```

```
    while (i < megisto_plithos && yposyn) {
```

```
        if (Melos(i, s1) && !Melos(i, s2))
```

```
            yposyn = FALSE;
```

```
        else
```

```
            i++;
```

```
    }
```

```
    return yposyn;
```

```
}
```

Υposynolo(S2,S1): TRUE

S1={2, 4, 6, 8}

$S2 \subseteq S1$

S2={2, 4}

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

Melos(2, S2) && !Melos(2, S1)

Melos(4, S2) && !Melos(4, S1)

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

Υποσύνολο

```
boolean Υposynolo(typos_synolou s1, typos_synolou s2) {
```

```
    στοιχείο_synολου i;
```

```
    boolean yposyn;
```

```
    yposyn = TRUE;
```

```
    i = 0;
```

```
    while (i < megisto_plithos && yposyn) {
```

```
        if (Melos(i, s1) && !Melos(i, s2))
```

```
            yposyn = FALSE;
```

```
        else
```

```
            i++;
```

```
    }
```

```
    return yposyn;
```

```
}
```

Υposynolo(S2,S1): FALSE

S1={2, 4, 6, 8}

$S2 \not\subseteq S1$

S2={2, 5}

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

Melos(2, S2) && !Melos(2, S1)

Melos(5, S2) && !Melos(5, S1)

TRUE

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	TRUE
6	FALSE
7	FALSE
8	FALSE
9	FALSE

Πακέτο για τον ΑΤΔ ΣΥΝΟΛΟ με πίνακα

boolean Υposynolo(typos_synολου s1, typos_synολου s2)

*/*Δέχεται: Δύο σύνολα s1 και s2.*

Λειτουργία: Ελέγχει αν το σύνολο s1 είναι υποσύνολο του s2.

Επιστρέφει: Επιστρέφει TRUE αν το σύνολο s1 είναι ένα υποσύνολο του s2, δηλαδή αν κάθε στοιχείο του s1 είναι και στοιχείο του s2./**

```
{
    στοιχείο_synολου i;
    boolean yposyn;
    yposyn = TRUE;
    i = 0;
    while (i < megisto_plithos && yposyn) {
        if (Melos(i, s1) && !Melos(i, s2))
            yposyn = FALSE;
        else
            i++;
    }
    return yposyn;
}
```

Ένωση Συνόλων

EnosiSynolou(S1,S2)

S1={2, 4, 6, 8}

S1 ∪ S2

S2={3, 4, 5, 6}

enosi={2, 3, 4, 5, 6, 8}

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

Ένωση Συνόλων

EnosiSynolou(S1,S2)

S1={2, 4, 6, 8}

S1 ∪ S2

S2={3, 4, 5, 6}

enosi={2, 3, 4, 5, 6, 8}

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

```
void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi){
    στοιχειο_synolou i;

    for (i = 0; i < megisto_plithos; i++)
        enosi[i] = Melos(i, s1) || Melos(i, s2);
}
```

```
void EnosiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou enosi)
```

```
/*Δέχεται: Δύο σύνολα s1 και s2.
```

```
Λειτουργία: Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν ή στο s1 ή στο s2 ή και στα δύο σύνολα.
```

```
Επιστρέφει: Επιστρέφει το σύνολο enosi που προκύπτει από την ένωση των συνόλων s1 και s2.*
```

```
{  
    στοιχείο_synolou i;
```

```
    for (i = 0; i < megisto_plithos; i++)  
        enosi[i] = Melos(i, s1) || Melos(i, s2);
```

```
}
```


Τομή Συνόλων

Tomisynolou(S1,S2)

S1={2, 4, 6, 8}

$S1 \cap S2$

S2={3, 4, 5, 6}

tomis={4, 6}

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

Τομή Συνόλων

TomiSynolou(S1,S2)

S1={2, 4, 6, 8}

$S1 \cap S2$

S2={3, 4, 5, 6}

tomi={4, 6}

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

```
void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi){
    στοιχείο_synolou i;

    for (i = 0; i < megisto_plithos; i++)
        tomi[i] = Melos(i, s1) && Melos(i, s2);
}
```

```
void TomiSynolou(typos_synolou s1, typos_synolou s2, typos_synolou tomi)
```

*/*Δέχεται: Δύο σύνολα s1 και s2.*

Λειτουργία: Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν και στα δύο σύνολα s1 και s2.

Επιστρέφει: Επιστρέφει το σύνολο tomi που προκύπτει από την τομή των συνόλων s1 και s2./**

```
{  
    στοιχείο_synolou i;
```

```
    for (i = 0; i < megisto_plithos; i++)  
        tomi[i] = Melos(i, s1) && Melos(i, s2);
```

```
}
```

Διαφορά Συνόλων

DiaforaSynolou(S1,S2)

S1={2, 4, 6, 8}

$S1 - S2$

S2={3, 4, 5, 6}

diafora={2, 8}

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	FALSE
6	FALSE
7	FALSE
8	TRUE
9	FALSE

Διαφορά Συνόλων

DiaforaSynolou(S1,S2)

S1={2, 4, 6, 8}

S1 - S2

S2={3, 4, 5, 6}

diafora={2, 8}

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	TRUE
5	FALSE
6	TRUE
7	FALSE
8	TRUE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	FALSE
3	TRUE
4	TRUE
5	TRUE
6	TRUE
7	FALSE
8	FALSE
9	FALSE

θέση	τιμή
0	FALSE
1	FALSE
2	TRUE
3	FALSE
4	FALSE
5	FALSE
6	FALSE
7	FALSE
8	TRUE
9	FALSE

```
void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora){  
    στοιχειο_synolou i;
```

```
    for (i = 0; i < megisto_plithos; i++)
```

```
        diafora[i] = Melos(i, s1) && (!Melos(i, s2));
```

```
}
```

void DiaforaSynolou(typos_synolou s1, typos_synolou s2, typos_synolou diafora)

*/*Δέχεται: Δύο σύνολα s1 και s2.*

Λειτουργία: Δημιουργεί ένα νέο σύνολο με τα στοιχεία που ανήκουν στο σύνολο s1 και δεν ανήκουν στο s2.

Επιστρέφει: Επιστρέφει το σύνολο diafora που προκύπτει από την διαφορά των συνόλων s1-s2./**

{

στοixeio_synolou i;

for (i = 0; i < megisto_plithos; i++)

diafora[i] = Melos(i, s1) && (!Melos(i, s2));

}