Διαδικαστικός Προγραμματισμός

Ενότητα 6: Πίνακες-Συναρτήσεις

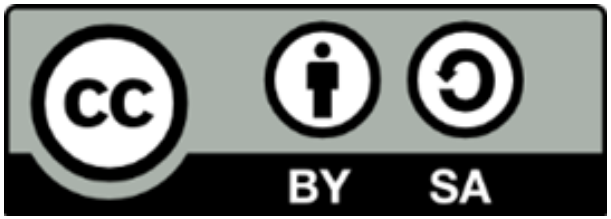
Καθηγήτρια Μαρία Σατρατζέμη

Τμήμα Εφαρμοσμένης Πληροφορικής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σκοποί ενότητας

- Να εκτιμήσετε τη σημασία που έχουν για τον προγραμματισμό οι πίνακες δεδομένων.
- Να κατανοήσετε τον τρόπο με τον οποίο αποθηκεύονται στη μνήμη οι τιμές δεδομένων.
- Να αναγνωρίσετε σε τι διαφέρει η μεταβίβαση πινάκων ως παραμέτρων συναρτήσεων από τη μεταβίβαση απλών μεταβλητών.
- Να κατανοήσετε τη δομή των πολυδιάστατων πινάκων.

Χειρισμός πινάκων μιας διάστασης (1)

- Όταν θέλουμε να προσπελάσουμε όλες τις θέσεις μνήμης ενός πίνακα μιας διάστασης, ο καλύτερος τρόπος είναι μέσω του βρόχου for, με τη βοήθεια του οποίου θα αλλάζει ο αριθμός αναφοράς της θέσης μνήμης.
- Οι παρακάτω περιπτώσεις χρήσης ενός μονοδιάσταστου πίνακα εμφανίζονται ως συναρτήσεις στις οποίες μεταβιβάζουμε ως παράμετρο έναν πίνακα 100 θέσεων μνήμης.

Χειρισμός πινάκων μιας διάστασης (2)

Συμπλήρωση πίνακα με τυχαίες τιμές:

```
void fill_it(int a[])
{
        int i;
        for (i = 0; i<100; i++)
        {
                a[i] = rand();
        }
}
```

Η πρόταση αυτή εκτελείται 100 φορές με τιμές του i από 0 μέχρι 99. Κάθε φορά που εκτελείται, ένας τυχαίος αριθμός καταχωρίζεται σε μια θέση μνήμης του πίνακα, την πρώτη φορά στην a[0], τη δεύτερη στην a[1] κτλ

Χειρισμός πινάκων μιας διάστασης (3)

Εμφάνιση περιεχομένων πίνακα :

```
void print_it(int a[])
{
        int i;
        for (i = 0; i<100; i++)
        {
                print("%d\n",a[i]);
        }
}
```

Η πρόταση αυτή εκτελείται 100 φορές με τιμές του i από 0 μέχρι 99. Κάθε φορά που εκτελείται, εμφανίζονται στην οθόνη τα περιεχόμενα μιας θέσης μνήμης του πίνακα, την πρώτη φορά της a[0], τη δεύτερη της a[1] κτλ

Χειρισμός πινάκων μιας διάστασης (4)

Υπολογισμός αθροίσματος περιεχομένων πίνακα:

```
int athroisma(int a[])
{
   int i,sum;
         sum=0;
         for (i=0;i<100;i++)
         {
                  sum=sum+a[i];
         }
         return sum;
}
```

Κάθε φορά που εκτελείται αυτή η πρόταση προστίθεται στη sum το περιεχόμενο μιας θέσης μνήμης του πίνακα. Την πρώτη φορά της a[0], τη δεύτερη της a[1] κτλ Στο τέλος η sum θα περιέχει το άθροισμα των περιεχομένων όλων των θέσεων μνήμης του πίνακα. Η συνάρτηση επιστρέφει ως τιμή το περιεχόμενο της sum.

Χειρισμός πινάκων μιας διάστασης (5)

Εμφάνιση ελάχιστης και μέγιστης τιμής περιεχομένων πίνακα :

```
void minmax(int a[])
{
        int min,max,i;
        min=max=a[0];
        for (i=0;i<100;i++)
        {
                if(a[i]<min) min=a[i];
                if(a[i]>max) max=a[i];
        }
        printf("O mikroteros arithmos einai: %d\n",min);
        printf("O megaliteros arithmos einai: %d\n",max);
}
```

Κάθε φορά που εκτελούνται αυτές οι προτάσεις ελέγχεται η τιμή μας θέσης μνήμης του πίνακα αν είναι μεγαλύτερη από τη μέχρι στιγμής μεγαλύτερη ή μικρότερη τιμή που έχει βρεθεί. Αν όντως βρεθεί μικρότερη ή μεγαλύτερη τιμή, τότε αυτή αντικαθιστά την υπάρχουσα τιμή στις μεταβλητές min και max αντίστοιχα. Αυτό γίνεται για κάθε θέση μνήμης και τελικά η min και η max θα περιέχουν τη μικρότερη και μεγαλύτερη τιμή αντίστοιχα από όλα τα αντίστοιχα στοιχεία του πίνακα.

Χειρισμός πινάκων μιας διάστασης (6)

Εύρεση μιας τιμής μέσα σε πίνακα :

```
int find_it(int a[ ],int ar)
{
        int i;
        bool found=FALSE;
        for (i=0;i<100;i++)
        {
                if(a[i]==ar) found = TRUE;
        }
        return found;
}
```

Η συνάρτηση ψάχνει να βρει αν ο αριθμός ar υπάρχει στον πίνακα a.

Κάθε φορά που εκτελείται αυτή η πρόταση, ελέγχεται η τιμή μιας θέσης μνήμης του πίνακα αν είναι ίση με τον αριθμό ar. Στην περίπτωση που βρεθεί ίση τιμή, τότε καταχωρίζεται στη θέση found η τιμή TRUE.

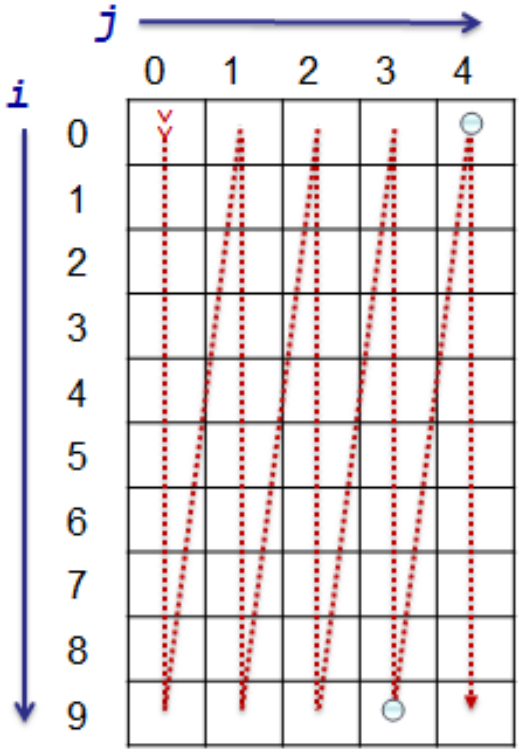
Μετά το τέλος του βρόχου for αν η τιμή της found είναι TRUE αυτό σημαίνει ότι ο ζητούμενος αριθμός βρέθηκε διαφορετικά όχι. ΠΡΟΣΟΧΗ η αρχική τιμή της found πρέπει να είναι FALSE.

Χειρισμός πινάκων δυο διαστάσεων (1)

- Στην έκδοση που ακολουθεί η επίσκεψη του πίνακα γίνεται ανά στήλη:

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

main()
{
        int i,j,a[10][5],timi;
        for(i=0;i<10;i++)
                        for(j=0;j<5;j++)
                        a[i][j]=rand();
}
```



Χειρισμός πινάκων δυο διαστάσεων (2)

Άθροισμα των στοιχείων ενός πίνακα 2Δ

Η συνάρτηση που ακολουθεί επιστρέφει ως τιμή το άθροισμα των στοιχείων ενός πίνακα ακεραίων 10Χ5

```
int sum(int p[ ][5])
{
        int i,j,ss=0;
        for(i=0;i<10;i++)
                for(j=0;j<5;j++)
                        ss=ss+p[i][j];
        return ss;
}
```

Στη συνάρτηση μεταβιβάζεται ένας πίνακας ακεραίων

Στη μεταβλητή ss αθροίζονται όλα τα στοιχεία του πίνακα

Το παρακάτω τμήμα κώδικα δείχνει τον τρόπο με τον οποίο μπορεί να χρησιμοποιηθεί η συνάρτηση sum από τη main() για τον υπολογισμό του αθροίσματος των στοιχείων του πίνακα a.

```
int a[10][5], ath;
...
ath=sum(a);
printf("Το άθροισμα των στοιχείων του a είναι%d\n",ath);
```

Χειρισμός πινάκων δυο διαστάσεων (1)

Εύρεση Μεγίστου στοιχείου

```
int mymax(int p[][5])
{
        int i,j,mx;
        mx=p[0][0];
        for(i=0;i<10;i++)
                for(j=0;j<5;j++)
                        if (p[i][j]>mx) mx=p[i][j];
        return mx;
}
```

Στην τοπική μεταβλητή mx καταχωρίζεται η τιμή της πρώτης θέσης μνήμης του πίνακα

Στην περίπτωση που βρεθεί τιμή μεγαλύτερη από το περιεχόμενο της mx, τότε αυτή καταχωρίζεται στην mx. Η mx περιέχει πάντα την μέχρι στιγμής μεγαλύτερη τιμή

Χειρισμός πινάκων δυο διαστάσεων (2)

Εύρεση Ελάχιστου στοιχείου

```
int mymim(int p[][5])
{
        int i,j,mn;
        mn=p[0][0];
        for(i=0;i<10;i++)
                for(j=0;j<5;j++)
                        if (p[i][j]<mn) mn=p[i][j];
        return mn;
}
```

Στην τοπική μεταβλητή mx καταχωρίζεται η τιμή της πρώτης θέσης μνήμης του πίνακα

Στην περίπτωση που βρεθεί τιμή μικρότερη από το περιεχόμενο της mn, τότε αυτή καταχωρίζεται στην mn. Η mn περιέχει πάντα την μέχρι στιγμής μικρότερη τιμή

Χειρισμός πινάκων δυο διαστάσεων (3)

Το παρακάτω τμήμα κώδικα δείχνει τον τρόπο με τον οποίο μπορεί να χρησιμοποιηθούν από τη main() οι παραπάνω συναρτήσεις για τον υπολογισμό της μεγαλύτερης και μικρότερης τιμής του πίνακα a.

```
. . .
int a[10][5];
printf("Η μεγαλύτερη τιμή του a είναι%d\n",mymax(a));
printf("Η μικρότερη τιμή του a είναι%d\n",mymim(a));
```

Στην περίπτωση αυτή, οι τιμές που επιστρέφουν οι συναρτήσεις δεν καταχωρίζονται σε καμία μεταβλητή αλλά εμφανίζονται κατευθείαν στην οθόνη

Χειρισμός πινάκων δυο διαστάσεων (4)

Εύρεση ενός αριθμού σε έναν πίνακα 2Δ

Η ακόλουθη συνάρτηση επιστρέφει την τιμή TRUE στην περίπτωση που ο αριθμός ar εντοπιστεί μέσα στον πίνακα. Διαφορετικά επιστρέφει την τιμή FALSE. Τόσο ο πίνακας όσο και ο αριθμός μεταβιβάζονται στη συνάρτηση ως παράμετροι.

```
bool find(int p[ ][5], int ar)
{
        int i,j;
   bool found;
        found=FALSE;
        for(i=0;i<10;i++)
        {
                for(j=0;j<5;j++)
                {
                        if (ar==p[i][j]) found=TRUE;
                        if (found) break;
                }
                if (found) break;
        }
        return found;
}
```

Στην τοπική μεταβλητή found καταχωρίζεται αρχικά η τιμή FALSE

Στην περίπτωση που βρεθεί ίση τιμή, τότε καταχωρίζεται στη θέση found η τιμή TRUE.

Στην περίπτωση που ο αριθμός έχει βρεθεί (found== TRUE) τότε διακόπτονται και οι 2 βρόχοι for.

Η συνάρτηση επιστρέφει την τιμή της found.

Χειρισμός πινάκων δυο διαστάσεων (5)

Μορφοποιημένη εμφάνιση ενός πίνακα 2Δ

Η ακόλουθη συνάρτηση εμφανίζει τα περιεχόμενα ενός πίνακα ακεραίων 10Χ5 σε μορφή γραμμών και στηλών.

```
void print_it(int p[ ][5])
{
    int i,j;
    for(i=0;i<10;i++)
    {
        for(j=0;j<5;j++)
            printf("%6d ",p[i][j]);
            printf("\n");
        }
}
```

Κάθε στοιχείο του πίνακα καταλαμβάνει έξη θέσεις στην οθόνη

Επιβάλλει αλλαγή γραμμής μετά από την εμφάνιση των στοιχείων κάθε σειράς του πίνακα

Χειρισμός πινάκων δυο διαστάσεων (6)

Εμφάνιση του αθροίσματος κάθε γραμμής ενός πίνακα 2Δ

Η ακόλουθη συνάρτηση εμφανίζει το άθροισμα των στοιχείων κάθε γραμμής ενός πίνακα ακεραίων 10Χ5. Δεδομένου ότι υπολογίζονται συγκεντρωτικά στοιχεία για κάθε γραμμή, η επίσκεψη του πίνακα πρέπει να γίνει ανά γραμμή.

```
void sum_line(int p[][5])
{
        int i,j,ath;
        for(i=0;i<10;i++)
        {
                ath=0;
                for(j=0;j<5;j++)
                        ath=ath+p[i][j];
                printf("Γραμμή %d = %d\n",i,ath);
        }
}
```

Η μεταβλητή ath μηδενίζεται πριν από το «πέρασμα» κάθε γραμμής

Στη μεταβλητή ath αθροίζονται όλα τα στοιχεία της i γραμμής

Χειρισμός πινάκων δυο διαστάσεων (7)

Εμφάνιση του αθροίσματος κάθε γραμμής ενός πίνακα 2Δ

Η ακόλουθη συνάρτηση εμφανίζει τη μέγιστη τιμή κάθε στήλης ενός πίνακα ακεραίων 10Χ5. Δεδομένου ότι υπολογίζονται συγκεντρωτικά στοιχεία για κάθε στήλη χωριστά, η επίσκεψη του πίνακα πρέπει να γίνει ανά στήλη.

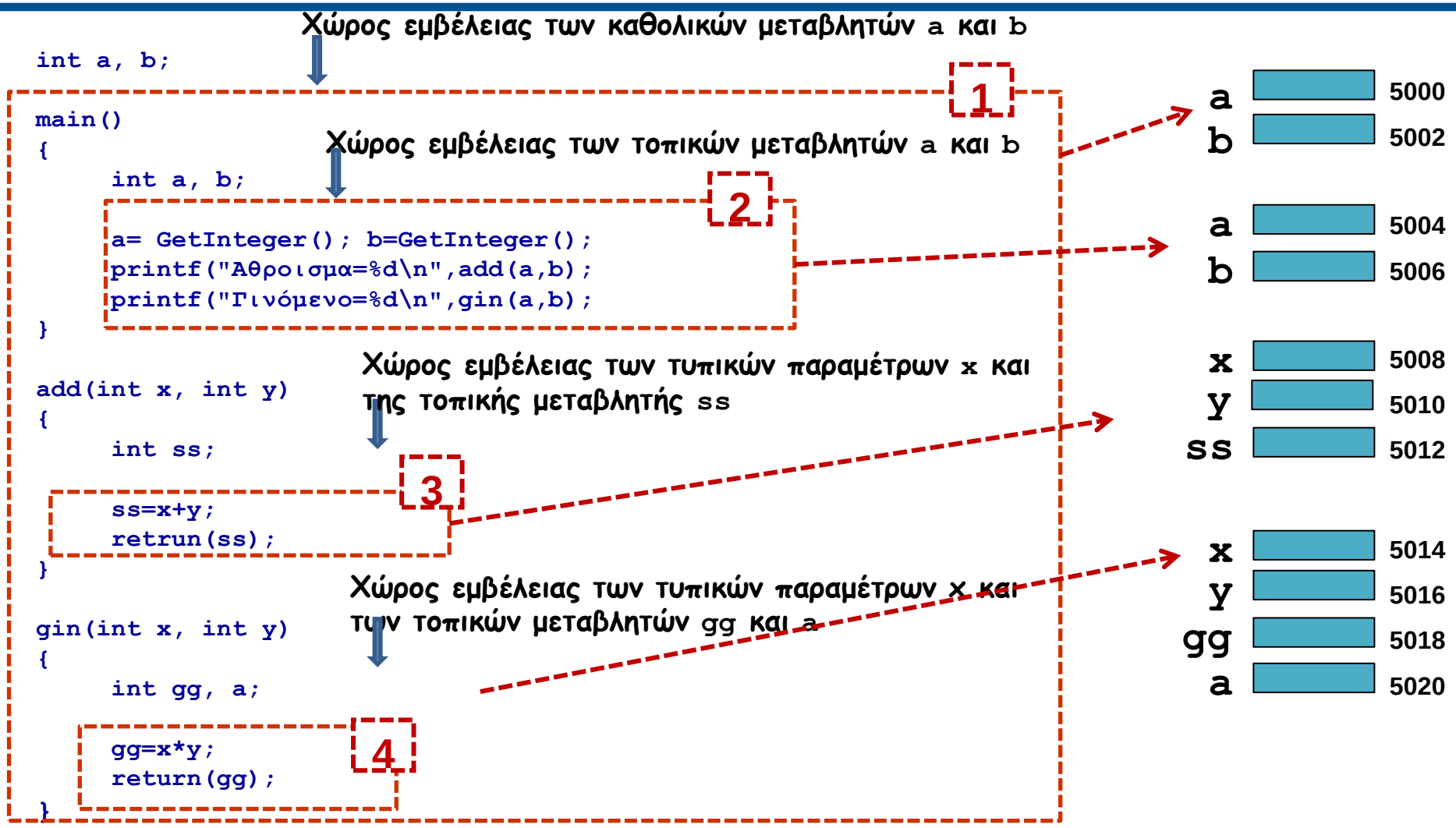
```
void max_col(int p[ ][5])
{
    int i,j,mx;
    for(j=0;j<5;j++)
    {
        mx=p[0][j];
        for(i=0;i<10;i++)
            if (p[i][j]>mx) mx=p[i][j];
            printf("Μέγιστος στήλης %d = %d\n",j,mx);
    }
}
```

Στη μεταβλητή mx καταχωρίζεται ως αρχική τιμή το περιεχόμενο της πρώτης θέσης (i == 0) κάθε στήλης

Στην περίπτωση που βρεθεί τιμή μεγαλύτερη από την mx τότε καταχωρίζεται στην mx. Η mx περιέχει πάντα την μέχρι στιγμής μεγαλύτερη τιμή της στήλης.

Εμφάνιση της μέγιστης τιμής κάθε στήλης

Εμβέλεια Μεταβλητών (1)



Εμβέλεια Μεταβλητών (2)

- Η μέθοδος μεταβίβασης τιμών σε μια συνάρτηση μέσω καθολικών μεταβλητών πρέπει γενικά να αποφεύγεται και να χρησιμοποιείται μόνο όταν πραγματικά υπάρχει ανάγκη.
- Σε μια καθολική μεταβλητή έχουν πρόσβαση όλες οι συναρτήσεις που βρίσκονται μετά από το σημείο του προγράμματος όπου δηλώθηκε. Όταν θέλουμε όλες ανεξαρτήτως οι συναρτήσεις να έχουν πρόσβαση σε μια καθολική μεταβλητή, πρέπει να δηλώσουμε τη μεταβλητή αυτή στην αρχή του προγράμματος (πριν από τη main()).
- Στην περίπτωση που μια τοπική μεταβλητή δηλωθεί με όνομα ίδιο με μια καθολική μεταβλητή, τότε στο χώρο εμβέλειας της τοπικής μεταβλητής μεταφερόμαστε στην τοπική μεταβλητή ενώ οπουδήποτε αλλού στην καθολική μεταβλητή.

Εμβέλεια Μεταβλητών (3)

- Οι μεταβλητές a και b της συνάρτησης main() του παραδείγματος χρησιμοποιούν εντελώς διαφορετικές θέσεις μνήμης από τις καθολικές μεταβλητές a και b.

Όταν γίνεται αναφορά σε μια μεταβλητή χρησιμοποιείται η μεταβλητή που έχει τον πλησιέστερο χώρο εμβέλειας.

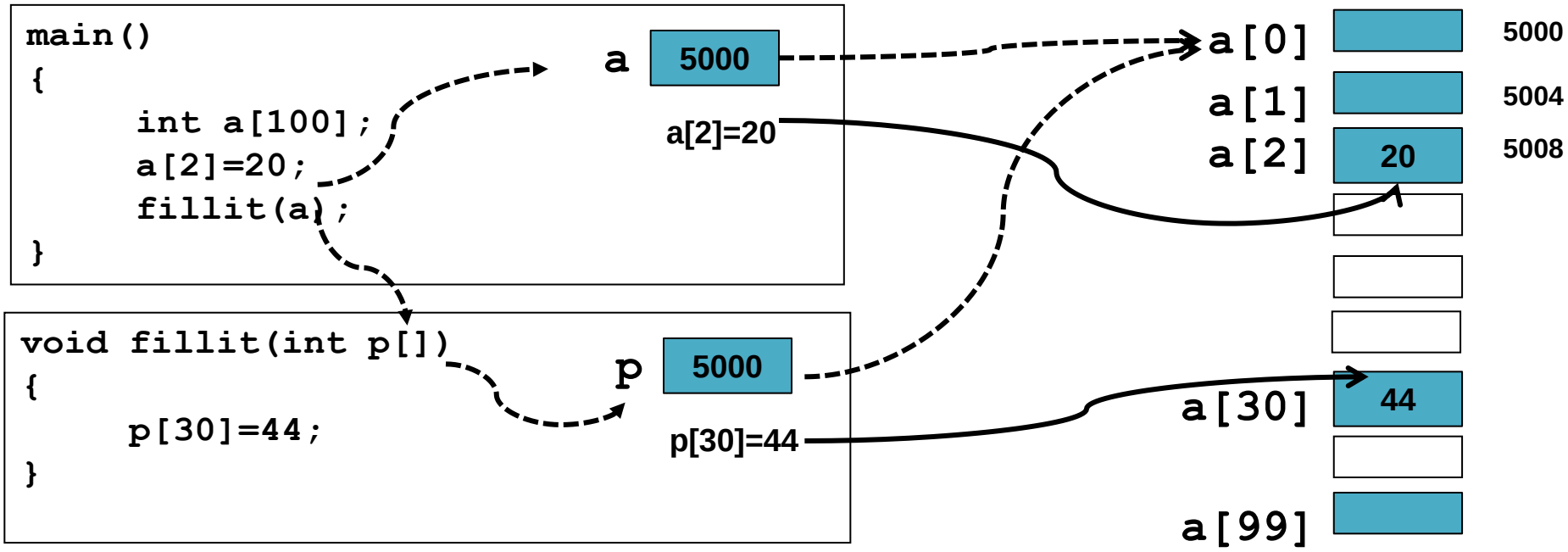
Εμβέλεια Μεταβλητών (4)

Μεταβλητή	a	b	a	b	x	y	ss	x	y	gg	a
Διεύθυνση	5000	5002	5004	5006	5008	5010	5012	5014	5016	5018	5020
1											
2											
3											
4											

Έτσι, σε ένα σημείο του προγράμματος που βρίσκεται **μόνο** στο χώρο 1 υπάρχει δυνατότητα πρόσβασης **μόνο** στις μεταβλητές **a** (5000) και **b** (5002). Ένα σημείο του προγράμματος που βρίσκεται στο χώρο 4 έχει πρόσβαση στις μεταβλητές **b** (5002), **x** (5014), **y** (5016), **gg** (5018), **a** (5020) και σε **καμία** άλλη.

Εμβέλεια Μεταβλητών (5)

Στο παρακάτω παράδειγμα ο πίνακας **a** δηλώνεται στη συνάρτηση **main()** και μεταβιβάζεται στη συνάρτηση **fillit()** μέσω της τυπικής παραμέτρου **p**.



Μέσα από τη **main()** υπάρχει πρόσβαση στον πίνακα μέσω του ονόματος **a**, ενώ μέσα από τη συνάρτηση **fillit()** η πρόσβαση στον ίδιο πίνακα **a** γίνεται μέσω του ονόματος **p**.

Τέλος Ενότητας