
# Προγραμματισμός Υπολογιστών

## Πίνακες σαν παράμετροι σε συναρτήσεις

Νικόλαος Ζ. Ζάχαρης
Καθηγητής

Πανεπιστήμιο Δυτικής Αττικής
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

## Πίνακες σαν ορίσματα συναρτήσεων

Μπορούμε να δημιουργήσουμε συναρτήσεις οι οποίες θα δέχονται σαν ορίσματα έναν ή περισσότερους πίνακες ανάλογα με το πρόβλημα. Οι πίνακες για να χρησιμοποιηθούν σαν ορίσματα θα πρέπει να γραφούν σαν δείκτες δηλαδή χωρίς διάσταση. Για παράδειγμα για ένα πίνακα ακεραίων με το όνομα Pin θα πρέπει να τον γράψουμε

int *Pin  είτε ισοδύναμα  int Pin[]

Κατά αυτό το τρόπο δηλώνουμε ότι θα περάσουμε σαν όρισμα μόνο την αρχή ενός πίνακα ακεραίων ανεξαρτήτου διάστασης. Άρα στις συναρτήσεις που χρησιμοποιούν πίνακες θα πρέπει οπωσδήποτε ένα από τα ορίσματα να είναι το πλήθος των στοιχείων του πίνακα. Αυτό είναι λογικό από την στιγμή που χρησιμοποιούμε πίνακες διαφορετικών διαστάσεων και ξέρουμε μόνο την αρχή του πίνακα θα πρέπει να γνωρίζουμε και το πλήθος των στοιχείων του πίνακα.

**Η αρχή ενός πίνακα ακεραίων χωρίς συγκεκριμένη διάσταση. Θα μπορούσαμε να το γράψουμε και int *X**

**Το πλήθος των στοιχείων του πίνακα**

```
int sumPin(int X[], int n)
{
  int i, sum = 0;
  for(i = 0; i < n; i++)
  {
    sum = sum + X[i];
  }
  return (sum);
}
```

Από εδώ και κάτω θεωρώ ότι υπάρχει ένας πίνακας με όνομα X και n πλήθος στοιχείων. Τότε για να υπολογίσω το άθροισμα των στοιχείων θα πρέπει να κάνω μια επανάληψη από το 0 μέχρι n-1 και σε κάθε επανάληψη να προσθέτω το κάθε στοιχείο το πίνακα σε μια μεταβλητή, η οποία έχει αρχική τιμή 0. Τέλος επιστρέφω το άθροισμα

```
int main(int argc, char *argv[]) {
  int A[3] = { 4, 5, 6};
  int K[5] = { 6, 9, 2, 3, 4};
  printf("%d\n", sumPin(A, 3));
  printf("%d\n", sumPin(K, 5));
}
```

Δημιουργούμε δύο πίνακες με διαφορετικό πλήθος στοιχείων και ονόματα A και K αντίστοιχα. Καλούμε την sumPin με ορίσματα A και 3. Σε αυτή την περίπτωση το X δείχνει στην αρχή του A και το n είναι 3. Στην δεύτερη περίπτωση το X δείχνει στην αρχή του K και το n είναι 5.

Σαν είσοδο σε μια συνάρτηση μπορούμε να έχουμε απλούς τύπους π.χ. ένα ακέραιο, ένα πραγματικό κ.λπ. Ή μπορούμε να έχουμε ένα πίνακα.

```
int SmallValue(int *pinakas, int no){
int i, min;
  min = pinakas[0];
  for (i = 1; i < no; i++)
     if(pinakas[i] < min)
        min = pinakas[i];
  return min;
}
```

```
int SmallValue(int *pinakas, int no) {
int i, min;
  min = *pinakas;
  for (i = 1; i < no; i++)
     if(*(pinakas+i) < min)
        min = *(pinakas+i);
  return min;
}
```

Οι ανωτέρω συναρτήσεις είναι ισοδύναμες και δέχονται σαν όρισμα την διεύθυνση ενός πίνακα ή πιο συγκεκριμένα την διεύθυνση του πρώτου στοιχείου. Στην main καλούμε αρχικά την συνάρτηση με όρισμα την διεύθυνση του πίνακα test και έπειτα με την διεύθυνση του pinB. Για αυτό το λόγο περνάμε σαν δεύτερο όρισμα το μήκος του πίνακα, ώστε ξεκινώντας από την αρχική διεύθυνση να διασχίσουμε το συγκεκριμένο πλήθος τιμών.

```
int main(int argc, char *argv[]) {
  int test[5] = {4, 2, 6, 8, 9};
  int pinB[3] = {3, 8, 9};
  printf("%d\n",SmallValue(test, 5));
  printf("%d\n",SmallValue(pinB, 3));
 return 0;
}
```

Το όνομα ενός πίνακα η διεύθυνση του πρώτου στοιχείου και μπορούμε να την αποδώσουμε σε ένα δείκτη. Αφού ο δείκτης μπορεί να δείχνει σε διαφορετικές θέσεις κατά συνέπεια μπορούμε να περάσουμε σαν ορίσματα διαφορετικούς πίνακες.

## Παράδειγμα

Να δημιουργήσετε τέσσερις συναρτήσεις με τα ονόματα fillArray, sortArray, showArray και findPos αντίστοιχα. Οι τρεις συναρτήσεις θα δέχονται δύο ορίσματα : α) ένα δείκτη σε ένα πίνακα ακεραίων και β) έναν ακέραιο αριθμό που αντιστοιχεί στο πλήθος των στοιχείων του πίνακα. Η fillArray θα γεμίζει τις αντίστοιχες θέσεις του πίνακα με τιμές από το πληκτρολόγιο. Η sortArray θα ταξινομεί την πίνακα και η showArray θα εκτυπώνει τα στοιχεία του πίνακα. Η findPos θα δέχεται τα ίδια ορίσματα και επιπλέον ένα ακέραιο αριθμό, του οποίου θα αναζητά την θέση μέσα στο πίνακα των ακεραίων. Αν δεν υπάρχει ο ακέραιος αριθμός τότε η findPos θα επιστρέφει την τιμή -1.

Στην συνάρτηση main να δηλώσετε ένα πίνακα ακεραίων με χωρητικότητα 5 θέσεων και με την βοήθεια των τριών πρώτων συναρτήσεων να το γεμίσετε με αριθμούς που θα δώσετε από το πληκτρολόγιο, να τον ταξινομήσετε και να εκτυπώσετε τα στοιχεία του. Τέλος χρησιμοποιήσετε την findPos για να αναζητήσετε την θέση μέσα στο πίνακα κάποιου ακεραίου αριθμού που θα διαβάσετε από το πληκτρολόγιο.

```
void fillArray(int *A, int n) {
  int i;
  for(i = 0; i < n; i++)
     scanf("%d", &A[i]);
}

void showArray(int *A, int n) {
  int i;
  for(i = 0; i < n; i++)
     printf("%d\t",A[i]);

   printf("\n");
}

void swap(int *a, int *b) {
  int temp;
  temp = *a;
  *a = *b;
  *b = temp;
}
```

```
void sortArray(int *A, int n) {
   int i, j;

   for(j = 0; j < n; j++)
     for(i = 0; i < (n - 1); i++)
        if(A[i] > A[i+1])
           swap(&A[i], &A[i+1]);
}

int findPos(int *A, int n, int X) {
  int i, pos = -1;
  for(i = 0; i < n; i++)
    if(A[i] == X)
    {
       pos = i;
       break;
    }

  return (pos);
}
```

```
int main(int argc, char *argv[]) {
  int APin[3], no, pos;

  fillArray(APin, 3);
  sortArray(APin, 3);
  showArray(APin, 3);

  printf("Type an integer number : ");
  scanf("%d", &no);
  pos = findPos(APin, 3, no);

  if(pos < 0)
     printf("Not found\n");
  else
     printf("Found at position : %d\n", pos);

  return 0;
}
```

## Παράδειγμα

**Να δημιουργήσετε μια συνάρτηση με το όνομα calcPins, η οποία δέχεται δύο πίνακες πραγματικών με ίσο πλήθος στοιχείων έστω X και Y, και το πλήθος των στοιχείων του πίνακα, έστω n και επιστρέφει**

$$\sum_{i=0}^{n-1} \frac{X_i + Y_i}{X_i * Y_i}$$

**Στην συνάρτηση main να δημιουργήσετε δύο πίνακες πραγματικών με χωρητικότητα 3 θέσεων.**

```
float calcPins(float X[], float Y[], int n) {
  int i;
  float sum = 0;
  for(i = 0; i < n; i++)
  {
     sum = sum + ((X[i] + Y[i]) / (X[i] * Y[i]));
  }
  return(sum);
}
```

```
int main(int argc, char *argv[]) {
  float A[3], B[3];
  int i;
  for(i=0; i < 3; i++) {
    scanf("%f", &A[i]);
    scanf("%f", &B[i]);
}
  printf("%f", calcPins(A, B, 3));
  return 0;
}
```

# Πίνακες 2-διαστάσεων σαν ορίσματα

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

void ShowValues(int rows, int cols, int pin[2][3])
{
        int i, j;
        for(i=0; i < rows; i++) {
          for(j=0; j < cols; j++) {
            printf("%d\t", pin[i][j]);
    }
    printf("\n");
  }
}

int main(int argc, char *argv[]) {
  int  A[2][3] = {{ 4, 7, 8}, {78, 6, 5}};

  ShowValues(2, 3, A);

  return 0;
}
```

```
4       7       8
78      6       5
```

Σε μια συνάρτηση μπορούμε να περάσουμε σαν όρισμα και ένα πίνακα με 2 ή περισσότερες διαστάσεις.

Η πρώτη επιλογή είναι να δηλώσουμε τις διαστάσεις του πίνακα

```
void ShowValues(int rows, int cols, int pin[2][3])
```

# Πίνακες 2-διαστάσεων σαν ορίσματα

Η δεύτερη επιλογή είναι να αγνοήσουμε τη πρώτη διάσταση του πίνακα

```
void ShowValues(int rows, int cols, int pin[][3]){
        int i, j;
        for(i=0; i < rows; i++) {
          for(j=0; j < cols; j++) {
            printf("%d\t", pin[i][j]);
    }
    printf("\n");
  }
}
```

Η τρίτη επιλογή είναι να δηλώσουμε παραμετρικά τη δεύτερη διάσταση του πίνακα

```
void ShowValues(int rows, int cols, int pin[][cols]){
        int i, j;
        for(i=0; i < rows; i++) {
          for(j=0; j < cols; j++) {
            printf("%d\t", pin[i][j]);
    }
    printf("\n");
  }   }
```

Προσοχή για αυτή τη περίπτωση θα πρέπει το όρισμα **int cols** να προηγείται από τη δήλωση του πίνακα **int pin[][cols]**

Επίσης θα πρέπει ο μεταγλωττιστής να είναι C-99 συμβατός.

# Πίνακες 3-διαστάσεων σαν ορίσματα

```
//void ShowValues3D(int rows, int cols, int height, int pin[2][3][4]) {
//void ShowValues3D(int rows, int cols, int height, int pin[][3][4]) {
void ShowValues3D(int rows, int cols, int height, int pin[][cols][height]) {
        int i, j, k;
        for(i=0; i < rows; i++) {
          for(j=0; j < cols; j++) {
            for(k=0; k < height; k++) {
              printf("%d\t", pin[i][j][k]);
    }
    printf("\n");
  }
  printf("\n");
}
}

int main(int argc, char *argv[]) {

int B[2][3][4] = {  {  {1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4} },
                    {  {1, 2, 3, 4}, {1, 2, 3, 4}, {1, 2, 3, 4} }
                 };
  ShowValues3D(2,3,4, B);
  return 0;
}
```

**Εναλλακτικά**

Προσοχή για αυτή τη περίπτωση θα πρέπει τα ορίσματα **int cols** και **int height** να προηγούνται από τη δήλωση του πίνακα **int pin[][cols][height]**