Πανεπιστήμιο Κύπρου - Τμήμα Πληροφορικής
ΕΠΛ 032 – Προγραμματισμός Μεθόδων Επίλυσης Προβλημάτων
Σημειώσεις Εργαστηρίων
Μαρία Σταυρινού Ιωάννου

Πίνακες παράμετροι συναρτήσεων

Μέχρι τώρα έχουμε δεί συναρτήσεις που επιστρέφουν μόνο μία τιμή με τη βοήθεια της πρότασης return. Τώρα θα δούμε ότι οι πίνακες μπορούν να χρησιμοποιηθούν για την επιστροφή πολλαπλών δεδομένων από μία συνάρτηση. Το πρόγραμμα PASSBACK.C παρουσιάζει ένα τέτοιο παράδειγμα.

Το πρώτο ενδιαφέρον σημείο βρίσκεται στη δήλωση της συνάρτησης dosome όπου διαπιστώνουμε ότι υπάρχει η δήλωση ενός πίνακα ακεραίων, του list ως τυπικής παραμέτρου. Όταν ένας πίνακας είναι τυπική παράμετρος μίας συνάρτησης τότε δεν δηλώνεται το ακριβές μέγεθός του αλλά μόνο το γεγονός ότι είναι πίνακας (οι άδειες τετράγωνες παρενθέσεις) καθώς και ο τύπος στοιχείων του (εδώ ότι τα στοιχεία είναι ακέραιοι). Άρα ουσιαστικά γνωστοποιούμε στον μεταφραστή την βάση και την απόσταση (αριθμό bytes) μεταξύ στοιχείων.

Στο κυρίως πρόγραμμα έχουμε τη δήλωση του μετρητή index και ενός πίνακα 20 ακεραίων με το όνομα matrix. Οι εκτελέσιμες προτάσεις που ακολουθούν είναι ία σειρά από επαναλήψεις for, τόσο στο κυρίως πρόγραμμα όσο και στη συνάρτηση. Προσέξτε ότι κάθε φορά εκτυπώνονται μόνο τα 5 πρώτα στοιχεία του πίνακα για λόγους συντομίας στην εκτύπωση. Αν θέλετε μπορείτε να πάρετε πλήρεις εκτυπώσεις αλλάζοντας τη συνθήκη στις επαναλήψεις εκτύπωσης.

Το πρώτο for της main θέτει κάποιες τιμές στον πίνακα matrix, ενώ το δεύτερο εμφανίζει ορισμένα από τα υπάρχοντα δεδομένα. Στη συνέχεια καλείται η συνάρτηση dosome με πραγματική παράμετρο τον πίνακα matrix ο οποίος συνδέεται με τον πίνακα - τυπική παράμετρο list. Η dosome εκτυπώνει πρώτα ορισμένα από τα στοιχεία του πίνακα που παρέλαβε. Στη συνέχεια προσθέτει 10 σε όλα τα στοιχεία του πίνακα, εκτυπώνει τον νέο αποτέλεσμα και τερματίζει. Μετά την επιστροφή στη main το τελευταίο for εκτυπώνει ξανά τα στοιχεία του πίνακα matrix. Το αποτέλεσμα δίνεται αμέσως.

```
Start  matrix[0] = 1
Start  matrix[1] = 2
Start  matrix[2] = 3
Start  matrix[3] = 4
Start  matrix[4] = 5
Before matrix[0] = 1
Before matrix[1] = 2
```

```
Before matrix[2] = 3
Before matrix[3] = 4
Before matrix[4] = 5
After  matrix[0] = 11
After  matrix[1] = 12
After  matrix[2] = 13
After  matrix[3] = 14
After  matrix[4] = 15
Back   matrix[0] = 11
Back   matrix[1] = 12
Back   matrix[2] = 13
Back   matrix[3] = 14
Back   matrix[4] = 15
```

Διαπιστώνουμε ότι με την κλήση της συνάρτησης dosome η τυπική παράμετρος list συνδέεται με την πραγματική matrix, αφού οι τιμές των στοιχείων των δύο πινάκων είναι οι ίδιες. Όμως, σε αντίθεση με τις βαθμωτές παραμέτρους, εδώ η σύνδεση δεν υλοποιείται με τη δημιουργία ενός πλήρους αντιγράφου του πίνακα στη στοίβα της συνάρτησης. Αυτό γίνεται κατανοητό από τις εκτυπώσεις του τερματισμού της συνάρτησης. Βλέπουμε ότι οι τροποποιήσεις στις τιμές των στοιχείων του πίνακα - τυπικής παραμέτρου list μεταφέρθηκαν και στον πίνακα - πραγματική παράμετρο matrix. Άρα η σύνδεση έγινε με τέτοιο τρόπο που η συνάρτηση να έχει τελικά πρόσβαση στον αρχικό πίνακα και όχι σε κάποια απομονωμένη τυπική παράμετρο.

Αυτή η σύνδεση γίνεται ως εξής. Κατά την κλήση μίας συνάρτησης που έχει ως τυπική παράμετρο ένα πίνακα στον χώρο των τυπικών παραμέτρων δεν αντιγράφεται ολόκληρος ο πίνακας - πραγματική παράμετρος αλλά αντιγράφεται απλά η βάση του. Στη περίπτωσή μας δηλαδή το όνομα matrix, που όπως είδαμε αντιστοιχεί στη διεύθυνση του μηδενικού στοιχείου του πίνακα, &matrix[0]. Επιπλέον, από τη δήλωση της συνάρτησης, ο μεταφραστής γνωρίζει την απόσταση ανάμεσα στα διαδοχικά στοιχεία του πίνακα της τυπικής (αλλά και της πραγματικής) παραμέτρου. Επομένως όλες οι τροποποιήσεις των στοιχείων του πίνακα list που συμβαίνουν στη συνάρτηση, συμβαίνουν κατ' ευθείαν επάνω στα στοιχεία του πίνακα matrix και όχι σε αντίγραφά τους. Αυτή η τεχνική περάσματος παραμέτρων λέγεται πέρασμα διεύθυνσης, σε αντίθεση με το πέρασμα τιμής που είχαμε μάθει μέχρι τώρα.

```
/* This is an example of matrices as function parameters */

#include <stdio.h>
void dosome(int list[]);

void main()
{
int index;
int matrix[20];

   for (index = 0;index < 20;index++)            /* generate data */
      matrix[index] = index + 1;

   for (index = 0;index < 5;index++)       /* print original data */
      printf("Start  matrix[%d] = %d\n",index,matrix[index]);

   dosome(matrix);            /* go to a function & modify matrix */

   for (index = 0;index < 5;index++)     /* print modified matrix */
      printf("Back   matrix[%d] = %d\n",index,matrix[index]);
}

void dosome(int list[])      /* Ts will illustrate returning data */
{
int i;

   for (i = 0;i < 5;i++)                 /* print original matrix */
      printf("Before matrix[%d] = %d\n",i,list[i]);

   for (i = 0;i < 20;i++)                 /* add 10 to all values */
      list[i] += 10;

   for (i = 0;i < 5;i++)                 /* print modified matrix */
      printf("After  matrix[%d] = %d\n",i,list[i]);
}
```

Πανεπιστήμιο Κύπρου - Τμήμα Πληροφορικής
ΕΠΛ 032 – Προγραμματισμός Μεθόδων Επίλυσης Προβλημάτων
Σημειώσεις Εργαστηρίων
Μαρία Σταυρινού Ιωάννου