

ΕΠΛ 033: ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ


Μάριος Belk, Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

Email: belk@cs.ucy.ac.cy





Τύποι Δεδομένων και Εγγραφές



Παράδειγμα 1: Επεξεργασία Σύνθετων Τύπων Δεδομένων

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

```
typedef struct {
```

3

```
    char title[50];
    char author[50];
    char subject[100];
    int book_id;
```

```
} Books;
```

```
int main( )
```

```
{
```

```
    int i;
```

```
    Books Book1={"C Programming", "Nuha Ali", "C Programming Tutorial", 6495407};
```

```
    /* Declare Book1 of type Book */
```

```
    Books Book2;          /* Declare Book2 of type Book */
```

```
    /* book 1 specification */
```

```
    /*strcpy( Book1.title, "C Programming");
```

```
    strcpy( Book1.author, "Nuha Ali");
```

```
    strcpy( Book1.subject, "C Programming Tutorial");
```

```
    Book1.book_id = 6495407;*/
```

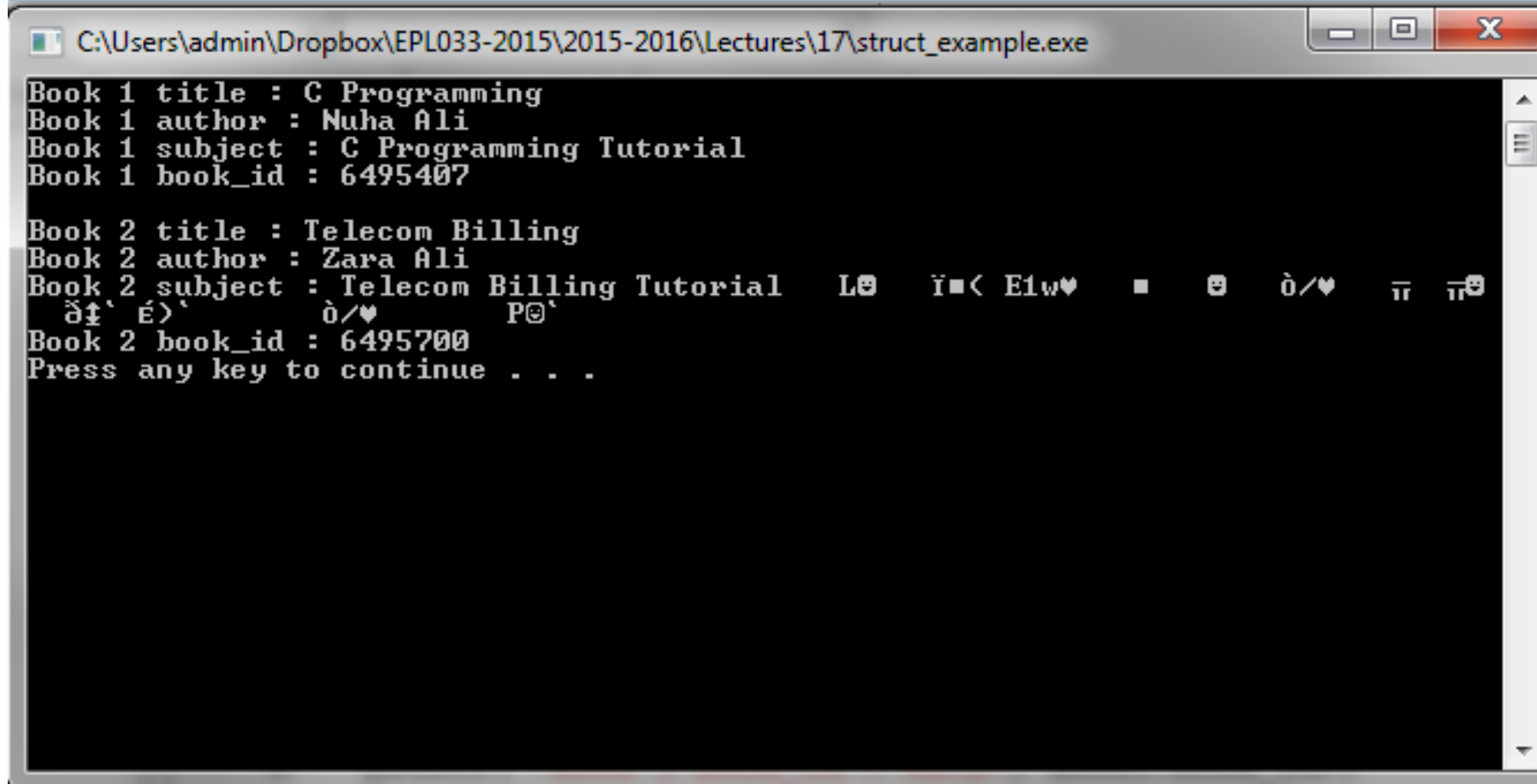
Συνέχεια στην επόμενη διαφάνεια...

```
/* book 2 specification */  
strcpy( Book2.title, "Telecom Billing");  
strcpy( Book2.author, "Zara Ali");  
strcpy( Book2.subject, "Telecom Billing Tutorial");  
Book2.book_id = 6495700;
```

```
/* print Book1 info */  
printf( "Book 1 title : %s\n", Book1.title);  
printf( "Book 1 author : %s\n", Book1.author);  
printf( "Book 1 subject : %s\n", Book1.subject);  
printf( "Book 1 book_id : %d\n", Book1.book_id);  
printf("\n");
```

```
/* print Book2 info */
//printf( "Book 2 title : %s\n", Book2.title);
printf( "Book 2 title : ");
for (i=0;i<strlen(Book2.title);i++)
    printf("%c", Book2.title[i]);
printf("\n");
printf( "Book 2 author : %s\n", Book2.author);
//printf( "Book 2 subject : %s\n", Book2.subject);
printf( "Book 2 subject : ");
for (i=0;i<100;i++)
    printf("%c", Book2.subject[i]);
printf("\n");
printf( "Book 2 book_id : %d\n", Book2.book_id);


return 0;
}
```



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\admin\Dropbox\EPL033-2015\2015-2016\Lectures\17\struct_example.exe. The window contains the following text output:

```
Book 1 title : C Programming
Book 1 author : Nuha Ali
Book 1 subject : C Programming Tutorial
Book 1 book_id : 6495407

Book 2 title : Telecom Billing
Book 2 author : Zara Ali
Book 2 subject : Telecom Billing Tutorial
Book 2 book_id : 6495700
Press any key to continue . . .
```



Παράδειγμα 2: Σύνθετοι Τύποι Δεδομένων και Δείκτες


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

8

```
#define N 2
```

```
typedef struct
{
    int id;
    float vathmoi[3];
} student_t;
```

Συνέχεια στην επόμενη διαφάνεια...

```
int main( )
```

```
{
```

```
    student_t student;
```

```
    student_t student_table[5];
```

9

```
    int i;
```

```
    /* student specification */
```

```
    student.id=123456;
```

```
    student.vathmoi[0]=1;
```

```
    student.vathmoi[1]=2;
```

```
    student.vathmoi[2]=3;
```

```
/* print Book1 info */
```

```
    printf("\nPassing the variable student of type student_t as  
function parameter...");
```

```
    printf("\nSize of student: %d", sizeof(student));
```

```
    print_student(student);
```

```
printf("\nPassing a pointer variable of type student_t as function  
parameter...");
```

```
    printf("\nSize of pointer student_s *s: %d", sizeof(&student));
```

```
    print_student_pointer(&student);
```

```
Passing the variable student of type student_t as function parameter...
```

```
Size of student: 16
```

```
Student id: 123456
```

```
Uathmos 1: 1.00
```

```
Uathmos 2: 2.00
```

```
Uathmos 3: 3.00
```

```
Passing a pointer variable of type student_t as function parameter...
```

```
Size of pointer student_s *s: 4
```

```
Student id: 123456
```

```
Uathmos 1: 1.00
```

```
Uathmos 2: 2.00
```

```
Uathmos 3: 3.00
```

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

```
{
```

```
    printf("\nPlease enter the information for student: %d\n",i+1);
```

```
    read_student(&student_table[i]);
```

```
}
```

11

C:\Users\admin\Dropbox\EPL033-2015\2015-2016\Lectures\17\struct_example_2.exe

Passing the variable student of type student_t as function parameter...

Size of student: 16

Student id: 123456

Uathmos 1: 1.00

Uathmos 2: 2.00

Uathmos 3: 3.00

Passing a pointer variable of type student_t as function parameter...

Size of pointer student_s *s: 4

Student id: 123456

Uathmos 1: 1.00

Uathmos 2: 2.00

Uathmos 3: 3.00

Please enter the information for student: 1

234432

23

34

45

Please enter the information for student: 2

543211

45

56

99

```
printf("\nPassing the table student_table as function parameter...");
printf("\nSize of student_table is: %d\n", sizeof(student_table));
print_student_table(student_table);
```

12

```
printf("\nPassing a pointer variable of type student_t as function parameter...");
printf("\n(We only need the address of 1st element to process the elements of a table)");
printf("\nSize of &student_table[0] is: %d\n", sizeof(&student_table[0]));
```

```
print_student_table(&student_table[0]);
```

```
return 0;
```

```
}
```

Passing the table student_table as function parameter...
Size of student_table is: 80

Student id: 234432
Uathmos 1: 23.00
Uathmos 2: 34.00
Uathmos 3: 45.00

Student id: 543211
Uathmos 1: 45.00
Uathmos 2: 56.00
Uathmos 3: 99.00

Passing a pointer variable of type student_t as function parameter...
(We only need the address of 1st element to process the elements of a table)
Size of &student_table[0] is: 4

Student id: 234432
Uathmos 1: 23.00
Uathmos 2: 34.00
Uathmos 3: 45.00

Student id: 543211
Uathmos 1: 45.00
Uathmos 2: 56.00
Uathmos 3: 99.00

Press any key to continue . . .

```
void print_student(student_t s)
```

```
{  
    printf("\n");  
    printf( "Student id: %d\n", s.id);  
    printf( "Vathmos 1: %.2f\n", s.vathmoi[0]);  
    printf( "Vathmos 2: %.2f\n", s.vathmoi[1]);  
    printf( "Vathmos 3: %.2f\n", s.vathmoi[2]);  
}
```

14

```
void print_student_pointer(student_t *s)
```

```
{  
    printf("\n");  
    printf( "Student id: %d\n", s->id);  
    printf( "Vathmos 1: %.2f\n", s->vathmoi[0]);  
    printf( "Vathmos 2: %.2f\n", s->vathmoi[1]);  
    printf( "Vathmos 3: %.2f\n", s->vathmoi[2]);  
}
```

```
void print_student_table(student_t * s) {
```

```
    int i;
```

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

```
    {
```

```
        printf("\n");
```

```
        printf( "Student id: %d\n", s[i].id);
```

```
        printf( "Vathmos 1: %.2f\n", s[i].vathmoi[0]);
```

```
        printf( "Vathmos 2: %.2f\n", s[i].vathmoi[1]);
```

```
        printf( "Vathmos 3: %.2f\n", s[i].vathmoi[2]);
```

```
    }
```

```
} //The function below prints student_table using pointer arithmetics
```

```
/*void print_student_table(student_t * s) {
```

```
    int i;
```

```
    for(i=0;i<N;i++) {
```

```
        printf("\n");
```

```
        printf( "Student id: %d\n", s->id);
```

```
        printf( "Vathmos 1: %.2f\n", s->vathmoi[0]);
```

```
        printf( "Vathmos 2: %.2f\n", s->vathmoi[1]);
```

```
        printf( "Vathmos 3: %.2f\n", s->vathmoi[2]);
```

```
        s++;
```

```
    }
```

```
} */
```



```
/*void read_student(student_t *s){  
scanf("%d%f%f%f",  
&(*s).id, &(*s).vathmoi[0],  
16 &(*s).vathmoi[1], &(*s).vathmoi[2]);  
}*/
```

/* The function above and the function below both read data from the user using pointer variable s to type student_t * i.e. &(*s).id and &s->id are the same! */

```
void read_student(student_t *s){  
  
scanf("%d%f%f%f",  
&s->id, &s->vathmoi[0],  
&s->vathmoi[1], &s->vathmoi[2]);  
}
```

```
Passing the variable student of type student_t as function parameter...
Size of student: 16
Student id: 123456
Uathmos 1: 1.00
Uathmos 2: 2.00
Uathmos 3: 3.00

Passing a pointer variable of type student_t as function parameter...
Size of pointer student_s *s: 4
Student id: 123456
Uathmos 1: 1.00
Uathmos 2: 2.00
Uathmos 3: 3.00

Please enter the information for student: 1
234432
23
34
45

Please enter the information for student: 2
543211
45
56
99

Passing the table student_table as function parameter...
Size of student_table is: 80


Student id: 234432
Uathmos 1: 23.00
Uathmos 2: 34.00
Uathmos 3: 45.00

Student id: 543211
Uathmos 1: 45.00
Uathmos 2: 56.00
Uathmos 3: 99.00

Passing a pointer variable of type student_t as function parameter...
(We only need the address of 1st element to process the elements of a table)
Size of &student_table[0] is: 4

Student id: 234432
Uathmos 1: 23.00
Uathmos 2: 34.00
Uathmos 3: 45.00

Student id: 543211
Uathmos 1: 45.00
Uathmos 2: 56.00
Uathmos 3: 99.00
Press any key to continue . . .
```



Παράδειγμα 3: Φωλιασμένοι Σύνθετοι Τύποι Δεδομένων

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

19

```
typedef struct
{
    int college_id;
    char college_name[50];
} student_college_detail;
typedef struct
{
    int id;
    char name[20];
    float percentage;
    // structure within structure
    student_college_detail clg_data;
}student_detail;
```

```
int main()
```

```
{
```

```
    student_detail stu_data, *stu_data_ptr;
```

```
    //struct student_detail stu_data = {1, "Raju", 90.5, 71145,
```

```
    //                                     "Anna University"};
```

```
    stu_data_ptr = &stu_data;
```

```
    printf("Student id:");
```

```
    scanf("%d",&stu_data.id);
```

```
    printf("Student name:");
```

```
    scanf("%s",stu_data.name);
```

```
    printf("Percentage:");
```

```
    scanf("%f",&stu_data.percentage);
```

```
    printf("College id:");
```

```
    scanf("%d",&stu_data.clg_data.college_id);
```

```
    printf("College name:");
```

```
    scanf("%s",stu_data.clg_data.college_name);
```

```
printf("Student id:");
scanf("%d",&stu_data_ptr->id);
printf("Student name:");
scanf("%s",stu_data_ptr->name);
printf("Percentage:");
scanf("%f",&stu_data_ptr->percentage);
printf("College id:");
scanf("%d",&stu_data_ptr->clg_data.college_id);
printf("College name:");
scanf("%s",stu_data_ptr->clg_data.college_name);

printf("\nId is: %d \n", stu_data.id);
printf("Name is: %s \n", stu_data.name);
printf("Percentage is: %f \n\n", stu_data.percentage);

printf("College Id is: %d \n", stu_data.clg_data.college_id);
printf("College Name is: %s \n", stu_data.clg_data.college_name);

return 0;
}
```

C:\Users\admin\Dropbox\EPL033-2015\2015-2016\Lectures\17\struct_example_3.exe

```
Student id:123456  
Student name:Andreas  
Percentage:55
```

```
College id:333  
College name:UCY
```

```
Student id:654321  
Student name:Kostas  
Percentage:89
```

```
College id:444  
College name:CUT
```

```
Id is: 654321  
Name is: Kostas  
Percentage is: 89.000000
```

```
College Id is: 444  
College Name is: CUT  
Press any key to continue . . . _
```



Επιπλέον Ασκήσεις

Άσκηση 1

Δημιουργήστε ένα απλό πρόγραμμα που ορίζει μία δομή δεδομένων ενός αυτοκινήτου με τα εξής στοιχεία: κωδικός, μάρκα, μοντέλο, έτος παραγωγής και τιμή πώλησης, αρχικοποιεί και τυπώνει ένα παράδειγμα αυτοκινήτου.

Άσκηση 1

25

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

typedef struct{
    int code;
    char brand[10];
    char model[10];
    int year;
    float price;
} Car;
```

```
int main(){
    Car c;
    c.code = 1111;
    strcpy(c.brand, "BMW");
    strcpy(c.model, "X3");
    c.year = 2011;
    c.price = 48.500;

    printf("Car code: %d\n", c.code);
    printf("Car brand: %s\n", c.brand);
    printf("Car model: %s\n", c.model);
    printf("Car year: %d\n", c.year);
    printf("Car price: %.3f\n", c.price);
}
```

Άσκηση 2

26

Σας δίνεται το εξής πρόγραμμα (όπως φαίνεται στα αριστερά) και ζητείται το αποτέλεσμα της εκτέλεσης. Στη συνέχεια καλείστε να υλοποιήσετε ένα αντίστοιχο πρόγραμμα με πανομοιότυπη λειτουργία κάνοντας όμως χρήση δεικτών.

```
#include <stdio.h>

int function1(int a)
{
    a = 5;
    return a;
}

int main( ) {
    int a = 1;
    a = function1(a);
    printf("%d", a);
}
```

Αποτέλεσμα εκτέλεσης/εκτύπωσης: 5

```
#include <stdio.h>

void function1(int *a) {
    (*a) = 5;
}

int main( ) {
    int a = 1;

    function1(&a);

    printf("%d", a);
}
```

Άσκηση 3

27

Γράψτε μία συνάρτηση που θα δέχεται σαν ορίσματα έναν πίνακα ακεραίων, το μέγεθος του πίνακα και έναν ακέραιο αριθμό. Η συνάρτηση θα πρέπει να επιστρέφει πόσες φορές εμφανίζεται ο αριθμός αυτός μέσα στον πίνακα αριθμών.

Η συνάρτηση θα είναι της μορφής **int countInt(int arrayInt[], int size, int a)**

```
int countInt(int arrayInt[], int size, int a){
    int i, count=0;
    for (i=0;i<size;i++){
        if(arrayInt[i]==a)
            count++;
    }
    return count;
}
```

Άσκηση 4 (Λύση α)

28

Υλοποιήστε τη συνάρτηση `double getAverage()` η οποία καλείται από το πιο κάτω πρόγραμμα και υπολογίζει τον μέσο όρο των στοιχείων του πίνακα `balance`.

```
#include <stdio.h>
/* Δήλωση Συνάρτησης */

int main ()
{
    /* an int array with 5 elements */
    int balance[5] = {1000, 2, 3, 17, 50};
    double avg;
    /* pass pointer to the array as an argument */
    avg = getAverage( balance, 5 );
    /* output the returned value */
    printf("Average value is: %.2f\n", avg );
    return 0;
}
```

```
double getAverage(int *arr, int size) {
    int i, sum = 0;
    double avg;
    for (i = 0; i < size; ++i)
    {
        sum += *arr[i];
    }
    avg = (double)sum / size;
    return avg;
}
```

Σχόλια για την λύση α της Άσκησης 4

29

Στην κύρια συνάρτηση στην κλήση της `getAverage()` περνούμε σαν παράμετρο το `balance` που είναι ισοδύναμο με το `&balance[0]`. Οπότε στην `getAverage` πρέπει να έχουμε `int *arr` και όχι `int *arr[]`, για να μπορούν να συμφωνούν οι τύποι των δεδομένων στην κλήση και στον ορισμό της συνάρτησης.

Μετά μπορούμε να χρησιμοποιήσουμε κανονικά την εντολή `sum += *arr[i];` όπως φαίνεται στην πιο πάνω λύση για να μας μεταφέρει στην εκάστοτε θέση μνήμης που αντιστοιχεί σε κάθε στοιχείο του πίνακα `balance`, έστω κι αν η μεταβλητή `*arr` δεν είναι πίνακας!

Άσκηση 4 (Λύση β – χωρίς δείκτες)

30

Υλοποιήστε τη συνάρτηση `double getAverage()` η οποία καλείται από το πιο κάτω πρόγραμμα και υπολογίζει τον μέσο όρο των στοιχείων του πίνακα `balance`.

```
#include <stdio.h>
/* Δήλωση Συνάρτησης */

int main ()
{
    /* an int array with 5 elements */
    int balance[5] = {1000, 2, 3, 17, 50};
    double avg;
    /* pass pointer to the array as an argument */
    avg = getAverage( balance, 5 );
    /* output the returned value */
    printf("Average value is: %.2f\n", avg );
    return 0;
}
```

```
double getAverage(int arr[], int size) {
    int i, sum = 0;
    double avg;
    for (i = 0; i < size; ++i)
    {
        sum += arr[i];
    }
    avg = (double)sum / size;
    return avg;
}
```

Άσκηση 5

31

Να υλοποιήσετε ένα πρόγραμμα που να υπάρχει η δυνατότητα να καταχωρηθούν ακέραιες τιμές σ' έναν μονοδιάστατο πίνακα 10 θέσεων και να βρεθούν (α) η μέγιστη τιμή του πίνακα, (β) η ελάχιστη τιμή του πίνακα και (γ) η διαφορά τους (μέγιστη τιμή μείον ελάχιστη τιμή). Το πρόγραμμα θα πρέπει να τυπώνει τα αποτελέσματα στην οθόνη.

Άσκηση 5 (συνέχεια)

32

```
#include <stdio.h>

Int main()
{
int i, a[10], max, min;
/* καταχώρηση τιμών */
for (i=0; i<10; i++)
{
printf("\nΔώσε το %do στοιχείο του πίνακα : ", i+1);
scanf("%d", &a[i]);
} /* end of for */

max = a[0];
min = a[0];
```

```
for (i=1; i<10; i++)
{
    if (a[i] > max)
        max = a[i];
    if (a[i] < min)
        min = a[i];
} /* end of for */
printf("\nΗ μέγιστη τιμή του πίνακα είναι :
%d", max);
printf("\nΗ ελάχιστη τιμή του πίνακα είναι :
%d", min);
printf("\nΗ διαφορά μέγιστης και ελάχιστης
τιμής είναι : %d", \
max-min);

return 0;
} /* end of main */
```