Δομημένος Προγραμματισμός

Ενώσεις και επαναληπτική διάλεξη

Ελευθερία Κατσίρη

Επίκουρη Καθηγήτρια ΔΠΘ ΗΜΜΥ

Ιανουάριος 2021

Ενώσεις

Λέμε ένωση (union) μία περιοχή της μνήμης η οποία μπορεί να χρησιμοποιηθεί από διαφορετικές μεταβλητές και από διαφορετικούς τύπους μεταβλητών.

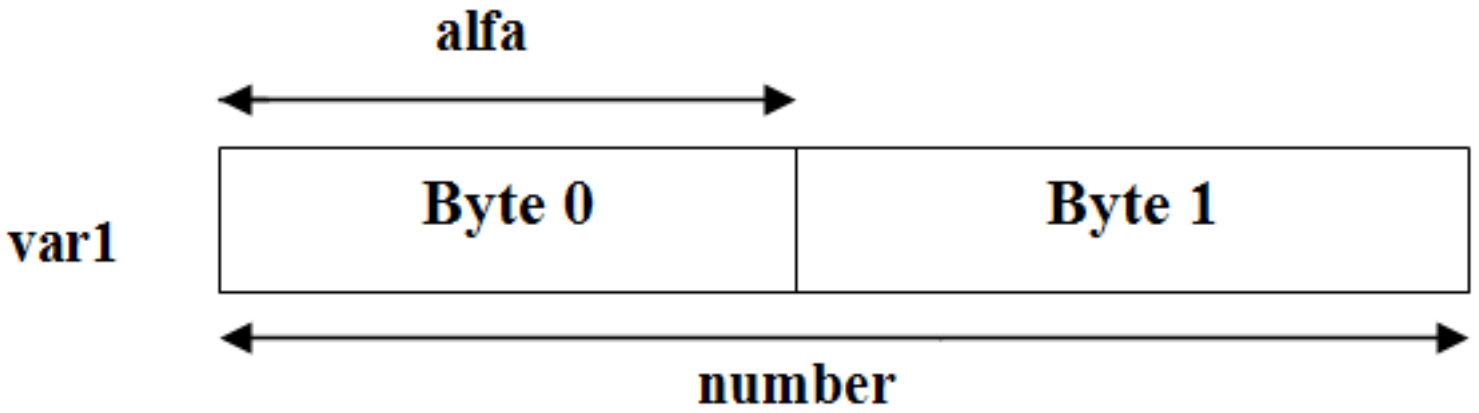
Η δήλωση μιας ένωσης, έστω first, η οποία θα αποτελείται από ένα χαρακτήρα και ένα βραχύ ακέραιο θα γράφεται :

```
union first
{
  char alfa;
  short int number;
} ;
```

Για να δηλώσουμε μια μεταβλητή, έστω var1, του τύπου της ένωσης first, θα μπορούσαμε να γράψουμε:

```
union first var1;
```

Παράσταση στη μνήμη της μεταβλητής var1 της ένωσης first



Για να προσεγγίσουμε ένα στοιχείο της ένωσης, χρησιμοποιούμε την ίδια σύνταξη που χρησιμοποιούμε για τις δομές δηλαδή,

τους τελεστές τελεία και βέλος.

Αν η μεταβλητή της ένωσης είναι γενική, χρησιμοποιούμε τον τελεστή τελεία

Αν η μεταβλητή της ένωσης περνάει σε μία συνάρτηση, τότε χρησιμοποιούμε τον τελεστή βέλος.

Παράδειγμα

Αν η μεταβλητή var1 της ένωσης unit είναι γενική, για να αντικαταστήσουμε τον ακέραιο 123 στο στοιχείο price, πρέπει να γράψουμε:

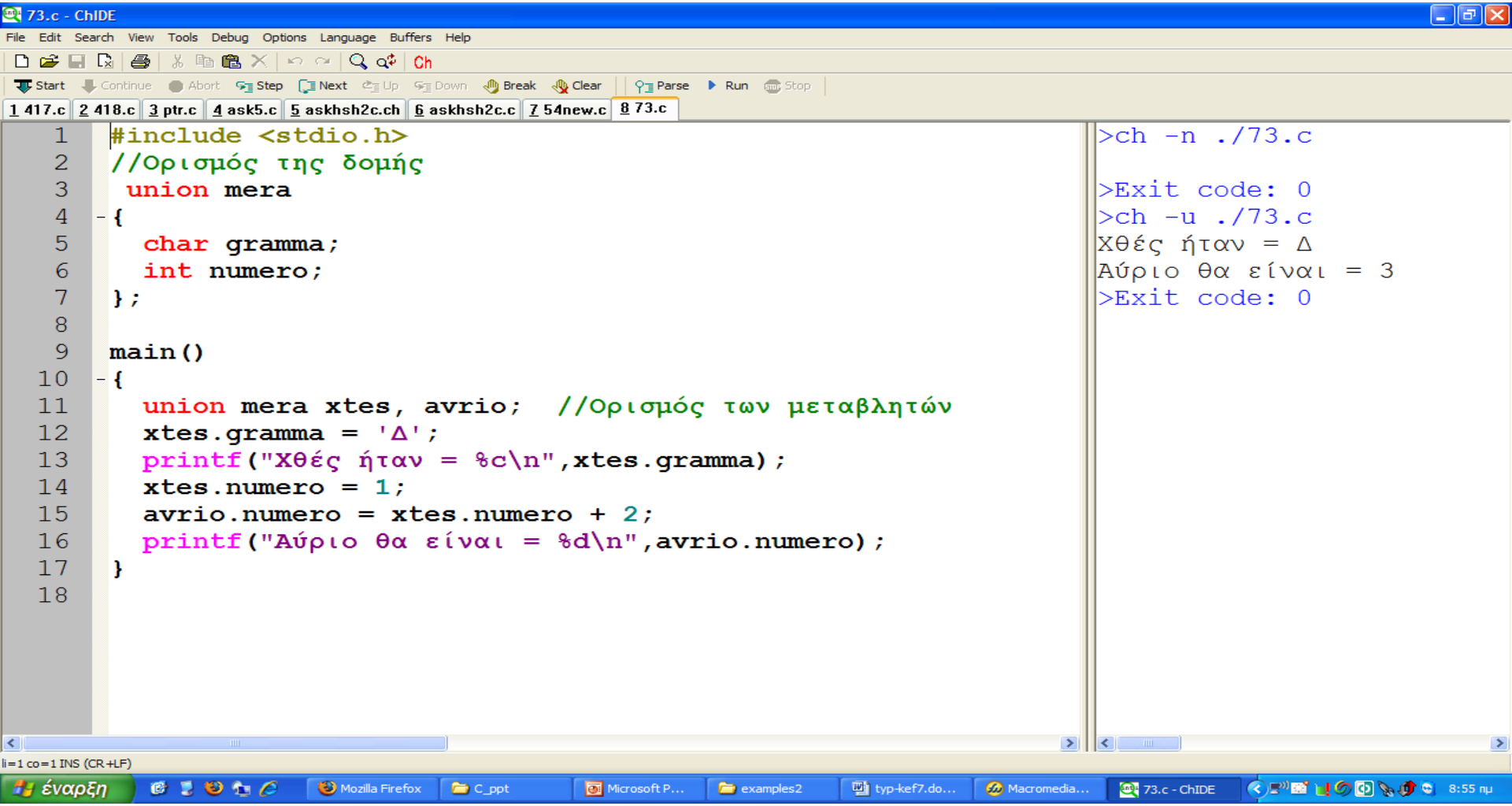
```
var1.price = 123;
```

Αν η unit περνούσε σε μία συνάρτηση, θα έπρεπε να χρησιμοποιήσουμε τον τελεστή - > :

```
function_test(xz)
   union unit xz;
{
    xz - > price = 123;
}
```

Πρόγραμμα

```
#include <stdio.h>
union mera     //Ορισμός ένωσης
    {
    char gramma;
    int numero;
    };
          main()
        {
          union mera xtes, avrio;      //Ορισμός των μεταβλητών
          xtes.gramma = 'Δ';
          printf("Χθές ήταν = %c\n",xtes.gramma);
          xtes.numero = 1;
          avrio.numero = xtes.numero + 2;
          printf("Αύριο θα είναι = %d\n",avrio.numero);
        }
```


73.c - ChIDE
File Edit Search View Tools Debug Options Language Buffers Help
Start Continue Abort Step Next Up Down Break Clear Parse Run Stop
1 417.c 2 418.c 3 ptr.c 4 ask5.c 5 askhsh2c.ch 6 askhsh2c.c 7 54new.c 8 73.c
#include <stdio.h>
//Ορισμός της δομής
union mera
{
char gramma;
int numero;
};
main()
{
union mera xtes, avrio; //Ορισμός των μεταβλητών
xtes.gramma = 'Δ';
printf("Χθές ήταν = %c\n",xtes.gramma);
xtes.numero = 1;
avrio.numero = xtes.numero + 2;
printf("Αύριο θα είναι = %d\n",avrio.numero);
}
>ch -n ./73.c
>Exit code: 0
>ch -u ./73.c
Χθές ήταν = Δ
Αύριο θα είναι = 3
>Exit code: 0
li=1 co=1 INS (CR+LF)
έναρξη
Mozilla Firefox
C_ppt
Microsoft P...
examples2
typ-kef7.do...
Macromedia...
73.c - ChIDE
8:55 πμ

ΕΙΔΙΚΕΣ ΕΝΤΟΛΕΣ ΤΗΣ C

- **Η έκδοση C99 (ISO/IEC 9899:1999) αναγνωρίζεται πλέον από όλους τους διαθέσιμους μεταγλωττιστές της γλώσσας C ενώ οι νέες επεκτάσεις της ANSI C11, C17 δεν έχουν ενσωματωθεί σε όλους τους μεταγλωττιστές της γλώσσας C**
  - (π.χ. C17: >= gcc 8.1, Visual Studio 2019)
- **Φυσικά, είναι θέμα χρόνου να προσαρμοστούν στις νέες προδιαγραφές της γλώσσας**
- **Μερικές εντολές (*λέξεις κλειδιά*) και των τριών εκδόσεων δεν χρησιμοποιούνται συχνά αφού απευθύνονται σε πολύ ειδικές διαδικασίες του προγραμματισμού.**

Αλλαγή ονόματος ενός τύπου δεδομένων

- **Εκτός από τους γνωστούς και προκαθορισμένους τύπους δεδομένων της γλώσσας C, (int, float, char κτλ.)**
- **μπορούμε να ορίσουμε και νέους τύπους δεδομένων, προσαρμοσμένους στις ανάγκες των προγραμμάτων χρησιμοποιώντας τη δεσμευμένη λέξη της γλώσσας C**

typedef

Ο γενικός τύπος της εντολής typedef είναι:

typedef *τύπος όνομα;*

- ***τύπος* είναι οποιοσδήποτε αποδεκτός τύπος δεδομένων της γλώσσας C, και**
- ***όνομα* είναι το νέο όνομα το οποίο θέλουμε να ορίσουμε για αυτό τον τύπο**

Παράδειγμα

Μπορούμε να δημιουργήσουμε ένα νέο όνομα για τον τύπο δεδομένων float, χρησιμοποιώντας την εντολή :

```
typedef  float  pragmatikos;
```

Τώρα, μπορούμε να δηλώσουμε μια μεταβλητή τύπου float χρησιμοποιώντας το όνομα pragmatikos γράφοντας:

```
pragmatikos xyz;
```

Η μεταβλητή xyz θα αντιμετωπίζεται από τον υπολογιστή σαν ένα κοινός πραγματικός αριθμός με όλες τις ιδιότητες του τύπου float

Αποτελέσματα

- **Χρησιμοποιώντας την εντολή typedef μπορούμε να κάνουμε ένα πρόγραμμα πιο εύκολο στην κατανόησή του τόσο από το δημιουργό του όσο και από μελλοντικούς προγραμματιστές για βελτιώσεις του αλγορίθμου.**
- **Επίσης, το πρόγραμμα αυτό γίνεται πιο αποτελεσματικό και εύχρηστο αφού μπορεί να μεταφερθεί και να λειτουργήσει χωρίς προβλήματα σε άλλους υπολογιστές ανεξαρτήτως μεγέθους, δυνατοτήτων και λειτουργικού συστήματος.**

Απαριθμήσιμοι τύποι μεταβλητών

- **Λέμε *απαριθμήσιμο* ένα νέο τύπο ο οποίος επιτρέπει στον προγραμματιστή να ορίσει αυθαίρετα στο πρόγραμμά του τις διακεκριμένες τιμές του τύπου αυτού.**
- **Με τη βοήθεια της εντολής enum ορίζουμε το γενικό όνομα και τα διακεκριμένα ονόματα των στοιχείων του απαριθμήσιμου τύπου καθώς και τις τιμές τις οποίες μπορεί να λάβει μια μεταβλητή όταν δηλωθεί ότι θα ανήκει σε αυτό τον τύπο.**
- **Ο γενικός τύπος δήλωσης ενός απαριθμήσιμου τύπου είναι:**

enum *όνομα { στοιχεία } μεταβλητές;*

Παράδειγμα

- **Αν γράψουμε:**

```
enum times
{
    a, b, c, d=12, e, f, g=12, h
} var ;
```

- **Οι τιμές των στοιχείων του απαριθμήσιμου τύπου times, θα είναι:**

  a=0, b=1, c=2, d=12, e=13, f=14, g=12, h=13

Ο τελεστής sizeof

Τόσο οι δομές όσο και οι ενώσεις μπορούν να χρησιμοποιηθούν για να δημιουργήσουν νέες μεταβλητές διαφόρων μεγεθών. Το μέγεθος των μεταβλητών αυτών εξαρτάται κυρίως από την έκδοση της γλώσσας C που χρησιμοποιεί ο υπολογιστής.

Ο μοναδιαίος τελεστής sizeof μπορεί να χρησιμοποιηθεί για να μας δώσει το ακριβές μέγεθος οποιουδήποτε τύπου μεταβλητής, συμπεριλαμβανομένων και των δομών, ενώσεων και μεταβλητών ορισμένων από το χρήστη.

- **Η γενική μορφή χρήσης της εντολής sizeof είναι:**

  Τιμή = sizeof(μεταβλητή)

- **όπου: Τιμή είναι ένας ακέραιος αριθμός ο οποίος υποδεικνύει το μέγεθος σε bytes της μεταβλητής**

Σημείωση. Ο εντολή sizeof μπορεί να βοηθήσει για να εξαλειφθούν από τα προγράμματα τα τμήματα του κώδικα που εξαρτώνται από την έκδοση της γλώσσας C.

Παράδειγμα

Αν έχουμε τη δήλωση της δομής:

```
{
   char name;
   float pound;
   int  year;
} jim;
```

τότε η εντολή:

```
X = sizeof (jim)
```

θα δώσει τιμή στη μεταβλητή X ίση με 12 επειδή οι περισσότεροι μεταγλωττιστές χρησιμοποιούν μια λέξη μνήμης (word) για κάθε μεταβλητής μιας δομής.

Τα μεγέθη των γνωστών τύπων δεδομένων της γλώσσας C

τύπος	μέγεθος σε bytes
char	1
unsigned char	1
int	4
unsigned int	4
short	2
unsigned short	2
long	4
unsigned long	4
float	4
long long	8
double	8
long double	10

Προσδιοριστές

Καλούνται *προσδιοριστές* (qualifiers) οι δεσμευμένες λέξεις οι οποίες προσδιορίζουν επί πλέον ιδιότητες εκτός από τη δήλωση του τύπου.

Π.χ. μπορούμε να γράφουμε:

```
volatile int j;
const long q;
const volatile unsigned long int k;
struct{
        const long int li;
         signed char sc;
       } volatile vs;
union{
        const long int li;
         signed char sc;
       } volatile vu;
```

Προσδιοριστής const

- **Με τη βοήθεια του προσδιοριστή const, δηλώνουμε μια μεταβλητή (απλή ή σύνθετη) της οποίας η τιμή θα παραμένει σταθερή σε όλη τη διάρκεια της εκτέλεσης του προγράμματος.**

Π.χ. δεν πρέπει να γράψουμε:

```
const int data = 0;   // απόδοση και τιμής
data = 1;
/* σοβαρό λάθος, εντοπίζεται από το μεταγλωττιστή*/
```

Προσδιοριστής volatile

- **Με τη βοήθεια του προσδιοριστή volatile, δηλώνουμε μια μεταβλητή (απλή ή σύνθετη) της οποίας η τιμή μπορεί να τροποποιηθεί κατά τη διάρκεια της εκτέλεσης του προγράμματος εν αγνοία του μεταγλωττιστή.**
- **Αυτή η ιδιότητα μιας μεταβλητής μπορεί να έχει πολλές εφαρμογές, όπως π.χ. οι συντεταγμένες του δείκτη ενός ποντικιού (mouse) οι οποίες χρησιμοποιούνται από μια εφαρμογή, μπορεί να μεταβάλλονται από ένα άλλο πρόγραμμα το οποίο τρέχει ταυτόχρονα.**

Προσδιοριστής restrict

- **Ο προσδιοριστής restrict, έχει προταθεί με την έκδοση C99 και εφαρμόζεται μόνο στις δηλώσεις των δεικτών.**
- **Με τη βοήθεια του προσδιοριστή restrict, ο προγραμματιστής επισημαίνει στο μεταγλωττιστή ότι ο δηλωθείς δείκτης θα είναι ο μοναδικός ο οποίος θα χρησιμοποιεί μια συγκεκριμένη περιοχή της μνήμης.**

Π.χ. μπορούμε να γράφουμε:

```
int * restrict pointer;
```

Συναρτήσεις γραμμής

- **Η δυνατότητα δήλωσης συναρτήσεων γραμμής στη γλώσσα C, έχει προταθεί με την έκδοση C99 και προέρχεται από τη γλώσσα C++.**
- **Η λέξη κλειδί για τη δήλωση μιας συνάρτησης γραμμής είναι η λέξη inline η οποία πρέπει να τοποθετηθεί ακριβώς πριν από τον τύπο επιστροφής της τιμής της συνάρτησης.**

Π.χ.

```
inline int add(int i, int j)
  {
    return i + j;
  }
```

Παρατήρηση

- **Η λέξη κλειδί inline δεν αποτελεί εντολή της γλώσσας C, απλά προειδοποιεί το μεταγλωττιστή ότι η συνάρτηση η οποία ακολουθεί δεν είναι μια απλή συνάρτηση αλλά μια συνάρτηση η οποία αντικαθιστά την κλήση της συνάρτησης με ολόκληρο το σώμα της συνάρτησης στο σημείο κλήσης της, αντί να δημιουργήσει ένα νέο σύνολο εντολών στη μνήμη.**

Άσκηση προς λύση

```
#include <stdio.h>
   main ( )
   {     char ch;    float f;    double d;
              struct client
            {   char name;
                 int  year;
            } alex;
    printf ("Client = %d \n",sizeof(Alex) );
    printf ("Char = %d \n",sizeof(ch) );
    printf ("Float = %d \n",sizeof(f));
    printf ("Double = %d \n",sizeof(d));
  }
```

ΑΣΚΗΣΗ

Να γραφτεί ένα πρόγραμμα το οποίο θα μετατρέπει ένα κείμενο το οποίο περιέχει πεζούς χαρακτήρες σε κείμενο με κεφαλαίους χαρακτήρες και αντίστροφα.

Απάντηση

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
    enum constants
        { // Ορισμός σταθερών τιμών
          ITEM= 4,
          DGR='α'-'A',
          DEN='a'-'A'
        };
  /* Ορισμός νέων τύπων δεδομένων */
 typedef char *STRING[ITEM];
 typedef char *PTR_STR;
 typedef char BITES8;
 typedef int BITES16;
void Upper (PTR_STR str1, PTR_STR str2);
void Lower (PTR_STR str1, PTR_STR str2);
```

```
main()
 {    STRING str1, str2;
      STRING tab =
    { "This example demonstrates how to set the color of the text.",
      "This example demonstrates how to set the background-color.",
      "Αυτό το παράδειγμα είναι μια επίδειξη Ελληνικών.",
      "Συνέχεια του παραδείγματος Ελληνικών."      };
       BITES16 i;
}
```

```
/*  Μετατροπή σε κεφαλαία  */
   for (i=0; i<ITEM; i++)
   {
     str1[i] = malloc((strlen(tab[i])+1) * sizeof(BITES8));
     Upper(tab[i], str1[i]);
  /*  Εμφάνιση του αρχικού κειμένου */
     printf("%s\n", tab[i]);
/*  Εμφάνιση της μετατροπής */
   for (i=0; i<ITEM; i++){
     printf("\n %s", str1[i]); }
     printf("\n");
}
```

```
/*  Μετατροπή σε πεζά */
    for (i=0; i<ITEM; i++)
    {  str2[i] = malloc((strlen(tab[i])+1) * sizeof(BITES8));
       Lower(str1[i], str2[i]);        }
  /*  Εμφάνιση της μετατροπής */
   for (i=0; i<ITEM; i++){
       printf("\n %s", str2[i]);
  /*  Απελευθέρωση της δεσμευμένης μνήμης */
       free (str1[i]);
       free (str2[i]); }
    printf("\n");
}
```

```
void Upper(PTR_STR str1, PTR_STR str2)
 {
   BITES16 i;
   for (i=0; str1[i]; i++){
    if ((str1[i] >= 'α') && (str1[i] <= 'ω'))
      str2[i] = str1[i] - DGR;
    else
       if ((str1[i] >= 'a') && (str1[i] <= 'z'))
         str2[i] = str1[i] - DEN;
       else
          str2[i] = str1[i];
    }
/* Πρόσθεση του τελικού χαρακτήρα */
   str2[i] = '\0';
}
```

```
void Lower(PTR_STR str1, PTR_STR str2)
 {
   BITES16 i;
   for (i=0; str1[i]; i++) {
     if ((str1[i] >= 'A') && (str1[i] <= 'Ω'))
       str2[i] = str1[i] + DGR;
     else
        if ((str1[i] >= 'A') && (str1[i] <= 'Z'))
          str2[i] = str1[i] + DEN;
        else
          str2[i] = str1[i];
   }
 /* Πρόσθεση του τελικού χαρακτήρα */
   str2[i] = '\0';
}
```

Αριθ. Μητρώου Περιττός

Να γράψετε μία συνάρτηση η οποία με τη βοήθεια ερωτήσεων στην οθόνη να ζητά:

όνομα παίκτη

ηλικία του (15-35)

χρησιμοποιώντας ένα πίνακα δομών

Η συνάρτησή να έχει μια παράμε-τρο, η οποία θα είναι ο δείκτης ενός στοιχείου του πίνακα.

Το κυρίως πρόγραμμα θα πρέπει να συμπληρώνει τα στοιχεία N = Α.Μ. παικτών και να εμφανίζει το όνομα του πρώτου παίκτη με ηλικία ίση με τη δική σας και τη μέγιστη τιμή της ηλικίας των παικτών της ομάδας αυτής

Αριθ. Μητρώου Άρτιος

Να γράψετε μία συνάρτηση η οποία με τη βοήθεια ερωτήσεων στην οθόνη να ζητά:

όνομα παίκτη

αριθμό φανέλας του (00-99)

χρησιμοποιώντας ένα πίνακα δομών

Η συνάρτησή να έχει μια παράμε-τρο, η οποία θα είναι ο δείκτης ενός στοιχείου του πίνακα.

Το κυρίως πρόγραμμα θα πρέπει να συμπληρώνει τα στοιχεία N=Α.Μ./100 παικτών και να εμφανίζει το όνομα του παίκτη με αριθμό φανέλας ίσο με τα 2 τελευταία ψηφία του Α.Μ. σας και τη μικρότερη τιμή του αριθμού φανέλας

Συγγράμματα

- **ΟΔΗΓΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΤΗ ΓΛΩΣΣΑ C,**

  Α. Καράκος, Εκδόσεις Καράκος, 2017, Ξάνθη, ISBN: 978-618-83086-0-2. Κωδικός βιβλίου στον Εύδοξο: 68371534

- **Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C,**

  Kernighan, Ritchie, 2η έκδοση, Κωδικός Βιβλίου στον Εύδοξο: 22701978

  ✓ Θεωρείται «Το» βιβλίο για τη C. Ο συγγραφέας είναι από τους συγγραφείς της γλώσσας.

  ✓ Αλλά: Δεν είναι για αρχαρίους, προυποθέτει κάποια εξοικοίωση με τις βασικές αρχές προγραμματισμού

Αξιολόγηση μαθήματος

- **Γραπτή εξέταση 50% βαθμού**
  - 1-2 προγράμματα,
- **Εξέταση εργαστηρίου 50% βαθμού**
  - MCQ, συμπλήρωσε τα κενά, βρες το λάθος
- **Δικαίωμα συμμετοχής στη γραπτή εξέταση:**

N-1 παρουσίες στα εργαστήρια, όπου
N=#εργαστηρίων
AND
Βαθμός εξέτασης εργαστηρίου > 5

Μαθησιακοί Στόχοι

- **Πρώτο μάθημα προγραμματισμού με χρήση της γλώσσας C**
  - Δεν απαιτείται προηγούμενη γνώση!
- **Εως το τέλος του μαθήματος θα μπορείτε:**
  - Να κατανοήσετε θεμελιώδεις έννοιες των διαδικασιακών γλωσσών προγραμματισμού (procedural languages)
  - Να μπορείτε να σχεδιάσετε αλγορίθμους για να επιλύετε απλά προβλήματα
  - Να χρησιμοποιείτε τη γλώσσα C

Ύλη μαθήματος

- Θεμελιώδεις Έννοιες
- Μεταγλωτίστε και τρέξτε το πρώτο πρόγραμμα σε C
- Μεταβλητές, Τύποι Δεδομένων και αριθμητικές εκφράσεις
- Βρόχοι επανάληψης
- Λήψη αποφάσεων
- Πίνακες
- Συναρτήσεις
- Δομές / Ενώσεις /προσδιοριστές / enums
- Αλφαριθμητικά
- Δείκτες
- Αφηρημένοι Τύποι

Παρουσίαση του περιεχομένου

Τα κεφάλαια του βιβλίου που προτείνεται, είναι :

1. **ΓΝΩΡΙΜΙΑ ΜΕ ΤΗ ΓΛΩΣΣΑ C**

   (Βασικές έννοιες και η δομή της γλώσσας)

2. **ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ C**

   (Οι μεταβλητές, οι σταθερές, οι τελεστές και οι εκφράσεις, εντολές εισόδου/εξόδου)

3. **ΕΝΤΟΛΕΣ ΕΛΕΓΧΟΥ**

   (Περιγραφή των εντολών ελέγχου)

   Το πιο σημαντικό κεφάλαιο για ένα προγραμματιστή

4. **ΣΥΝΑΡΤΗΣΕΙΣ**

   (Περιγραφή των συναρτήσεων, functions)

Παρουσίαση του περιεχομένου (συνέχεια)

5 ΔΕΙΚΤΕΣ ΚΑΙ ΠΙΝΑΚΕΣ

(Οι έννοιες των δεικτών (pointers) και εφαρμογές των πινάκων (arrays))

6 ΕΠΕΞΕΡΓΑΣΙΑ ΑΡΧΕΙΩΝ

(Περιγραφή και χρήση των αρχείων στη γλώσσα C)

7 ΔΟΜΕΣ ΚΑΙ ΕΝΩΣΕΙΣ

(Εισαγωγή στις δομές και τις ενώσεις)

ΕΦΑΡΜΟΓΕΣ της γλώσσας C