



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κεντρικής Μακεδονίας - Σέρρες
Τμήμα Μηχανικών Πληροφορικής

Προγραμματισμός II (Θ)

Δρ. Δημήτρης Βαρσάμης
Επίκουρος Καθηγητής

Μάρτιος 2017

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ II (Θ)

- 1 Εισαγωγή
- 2 Μακροεντολές
- 3 Βιβλιοθήκες οριζόμενες από τον χρήστη

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Μακροεντολές
- 3 Βιβλιοθήκες οριζόμενες από τον χρήστη

Αντικείμενο της παρούσης ενότητας είναι οι μακροεντολές και οι οριζόμενες από τον χρήστη βιβλιοθήκες. Πιο συγκεκριμένα θα ασχοληθούμε με:

- την δημιουργία των μακροεντολών
- την χρήση των μακροεντολών
- την δημιουργία βιβλιοθηκών οριζόμενες από τον χρήστη
- την χρήση των παραπάνω βιβλιοθηκών

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Μακροεντολές
- 3 Βιβλιοθήκες οριζόμενες από τον χρήστη

Μακροεντολές

- Οι μακροεντολές ορίζονται ως εντολές στον προεπεξεργαστή με την οδηγία `#define`
- Σε μια μακροεντολή μπορεί στην πιο απλή μορφή να ανατίθεται μια τιμή, δηλαδή,

```
|| #define Pi 3.14
```

- Σε μια μακροεντολή μπορεί στην πιο απλή μορφή να ανατίθεται μια έκφραση, δηλαδή,

```
|| #define signature printf("Dimitrios Varsamis\n")
```

Επιστρέφει μη μηδενική τιμή όταν δεχθεί κάποιο αριθμητικό χαρακτήρα (0-9), διαφορετικά επιστρέφει μηδέν.

- Σε μια μακροεντολή μπορεί να ανατίθεται μια έκφραση με παραμέτρους, δηλαδή,

```
|| #define add(a,b) ((a)+(b))
```

- Οι μακροεντολές είναι ανεξάρτητες από τον τύπο των μεταβλητών που χρησιμοποιούνται
- Ο προεπεξεργαστής τοποθετεί μέσα στο πρόγραμμα τον κώδικα των μακροεντολών και έπειτα μεταγλωττίζεται το πρόγραμμα.
- Επομένως, οι μακροεντολές είναι πιο γρήγορες από τις αντίστοιχες συναρτήσεις διότι δεν καλούνται.

Μακροεντολές - Παραδείγματα

```
#include <stdio.h>
#include <stdlib.h>
#define Pi 3.14
#define signature printf("Dimitrios Varsamis\n")
#define add(a,b) ((a)+(b))

int main() {
    int num1;
    float num2;
    printf("Give two numbers :\n");
    scanf("%d",&num1);
    scanf("%f",&num2);
    printf("Num1 + Num2 = %f\n",add(num1,num2));
    printf("Pi = %f\n",Pi);
    signature;
    return 0;
}
```


Μακροεντολές - Παραδείγματα

θα εμφανιστεί στην κονσόλα

```
Give two numbers :
```

```
2
```

```
3.1
```

```
Num1 + Num2 = 5.100000
```

```
Pi = 3.140000
```

```
Dimitrios Varsamis
```

Μακροεντολές - Παραδείγματα

Να δημιουργηθεί πρόγραμμα το οποίο με την βοήθεια μακροεντολών να ελέγχει αν ένας αριθμός a είναι άρτιος ή περιττός και να επιστρέφει $\frac{a}{2}$ ή $a - 1$ αντίστοιχα.

```
#include <stdio.h>
#include <stdlib.h>
#define even(a) ((a)%2==0 ? a/2 : a-1)

int main() {
    int num;
    printf("Give an integer :");
    scanf("%d", &num);
    printf("Result = %d\n", even(num));
    return 0;
}
```

θα εμφανιστεί στην κονσόλα

Μακροεντολές - Παραδείγματα

```
|| Give an integer :23  
|| Result = 22
```

ή

```
|| Give an integer :18  
|| Result = 9
```

Δομή Διαφανειών

- 1 Εισαγωγή
- 2 Μακροεντολές
- 3 Βιβλιοθήκες οριζόμενες από τον χρήστη

Βιβλιοθήκες οριζόμενες από τον χρήστη - Παραδείγματα

Να δημιουργηθεί βιβλιοθήκη η οποία να ορίζει μια δομή διανύσματα με μέλη τις συντεταγμένες και τις συναρτήσεις άθροισμα και διαφορά διανυσμάτων.

Για παράδειγμα αν

$$\vec{a} = (x_a, y_a), \quad \vec{b} = (x_b, y_b)$$

δυο διανύσματα οι πράξεις είναι οι παρακάτω

$$\vec{a} + \vec{b} = (x_a + x_b, y_a + y_b), \quad \vec{a} - \vec{b} = (x_a - x_b, y_a - y_b)$$

Αρχικά, ορίζουμε ένα αρχείο κεφαλίδας (.h) στο οποίο περιέχονται τα πρωτότυπα των συναρτήσεων και των τύπων που θέλουμε .

mylib.h

Βιβλιοθήκες οριζόμενες από τον χρήστη - Παραδείγματα

```
#ifndef _mylib_h
#define _mylib_h

struct vector{
    int x;
    int y;
};

typedef struct vector vec;
vec add_vec(vec a, vec b);
vec dif_vec(vec a, vec b);
#endif
```

Έπειτα στο αντίστοιχο αρχείο (.c) γράφουμε τα σώματα των συναρτήσεων **mylib.c**

Βιβλιοθήκες οριζόμενες από τον χρήστη - Παραδείγματα

```
#include "mylib.h"

vec add_vec(vec a, vec b) {
    vec res;
    res.x=a.x+b.x;
    res.y=a.y+b.y;
    return(res);
}

vec dif_vec(vec a, vec b) {
    vec res;
    res.x=a.x-b.x;
    res.y=a.y-b.y;
    return(res);
}
```

Βιβλιοθήκες οριζόμενες από τον χρήστη - Παραδείγματα

Τέλος, στο πρόγραμμα μας θα ορίσουμε δυο διανύσματα και θα επιστρέψουμε τα αποτελέσματα των πράξεων

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

int main() {
    vec a, b, ad, di;
    a.x=2;
    a.y=4;
    b.x=4;
    b.y=7;
    ad=add_vec(a, b);
    di=add_vec(a, b);
```


Βιβλιοθήκες οριζόμενες από τον χρήστη - Παραδείγματα

```
printf("( %d, %d) + ( %d, %d) = ( %d, %d) \n", a.x, a.y, b.x, b  
    .y, ad.x, ad.y);  
printf("( %d, %d) + ( %d, %d) = ( %d, %d) \n", a.x, a.y, b.x, b  
    .y, ad.x, ad.y);  
return 0;  
}
```

θα εμφανιστεί στην κονσόλα

```
(2, 4) + (4, 7) = (6, 11)
```

```
(2, 4) - (4, 7) = (-2, -3)
```