



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 Βιβλιοθήκες

- `<ctype.h>` (Χειρισμός χαρακτήρων)
- `<math.h>` (Μαθηματικές συναρτήσεις)
- `<stdlib.h>` (Διάφορες συναρτήσεις)
- `<time.h>` (Χειρισμός χρόνου)

# Δομή Διαφανειών

## 1 Εισαγωγή

## 2 Βιβλιοθήκες

- `<ctype.h>` (Χειρισμός χαρακτήρων)
- `<math.h>` (Μαθηματικές συναρτήσεις)
- `<stdlib.h>` (Διάφορες συναρτήσεις)
- `<time.h>` (Χειρισμός χρόνου)

Αντικείμενο της παρούσης ενότητας είναι οι συναρτήσεις βιβλιοθήκης.  
Θα αναφερθούμε στις βιβλιοθήκες

- `<ctype.h>` για χειρισμό αλφαριθμητικών
- πως τα δημιουργούμε
- πώς τα διαβάζουμε
- πώς τα ενημερώνουμε κ.α.

# Δομή Διαφανειών

## 1 Εισαγωγή

## 2 Βιβλιοθήκες

- `<ctype.h>` (Χειρισμός χαρακτήρων)
- `<math.h>` (Μαθηματικές συναρτήσεις)
- `<stdlib.h>` (Διάφορες συναρτήσεις)
- `<time.h>` (Χειρισμός χρόνου)

# Δομή Διαφανειών

## 1 Εισαγωγή

## 2 Βιβλιοθήκες

- `<ctype.h>` (Χειρισμός χαρακτήρων)
- `<math.h>` (Μαθηματικές συναρτήσεις)
- `<stdlib.h>` (Διάφορες συναρτήσεις)
- `<time.h>` (Χειρισμός χρόνου)

## <ctype.h> (Χειρισμός χαρακτήρων)

Μερικές από τις συναρτήσεις είναι

- `int isalnum(int ch)`

Επιστρέφει μη μηδενική τιμή όταν δεχθεί κάποιο αλφαριθμητικό χαρακτήρα (a-z, A-Z, 0-9), διαφορετικά επιστρέφει μηδέν.

- `int isalpha(int ch)`

Επιστρέφει μη μηδενική τιμή όταν δεχθεί κάποιο χαρακτήρα (a-z, A-Z), διαφορετικά επιστρέφει μηδέν.

- `int isdigit(int ch)`

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

- `int isspace(int ch)`

## <ctype.h> (Χειρισμός χαρακτήρων)

Επιστρέφει μη μηδενική τιμή όταν δεχθεί χαρακτήρα διαστήματος, διαφορετικά επιστρέφει μηδέν.

- `int islower(int ch)`

Επιστρέφει μη μηδενική τιμή όταν δεχθεί κάποιο πεζό χαρακτήρα (a-z), διαφορετικά επιστρέφει μηδέν.

- `int isupper(int ch)`

Επιστρέφει μη μηδενική τιμή όταν δεχθεί κάποιο κεφαλαίο χαρακτήρα (A-Z), διαφορετικά επιστρέφει μηδέν.

- `int tolower(int ch)`

Εάν δεχθεί κεφαλαίο χαρακτήρα επιστρέφει τον αντίστοιχο πεζό χαρακτήρα, διαφορετικά επιστρέφει την είσοδο.



## <ctype.h> (Χειρισμός χαρακτήρων)

- `int toupper(int ch)`

Εάν δεχθεί πεζό χαρακτήρα επιστρέφει τον αντίστοιχο κεφαλαίο χαρακτήρα, διαφορετικά επιστρέφει την είσοδο.

## <ctype.h> (Χειρισμός χαρακτήρων) -

### Παραδείγματα

Έστω το αλφαριθμητικό `**programMatismos==2==` το οποίο θέλουμε να μετατραπεί ως εξής `Programmatismos II`.

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

int main() {
    char strI[]="**programMatismos==2==", strF[20];
    int i, j=0, n;
    for(i=0; strI[i]!='\0'; i=i+1) {
        if(isalnum(strI[i])) {
            strF[j]=strI[i];
            j=j+1;
        }
    }
```

## Παραδείγματα

```
}  
n=strlen(strF);  
if(islower(strF[0])){  
    strF[0]=toupper(strF[0]);  
}  
for(i=1; i<n; i=i+1){  
    if(isupper(strF[i])){  
        strF[i]=tolower(strF[i]);  
    }  
}  
for(i=0; i<n; i=i+1){  
    if(isdigit(strF[i])){  
        strF[i]=' '  
        strF[i+1]='I';  
        strF[i+2]='I';  
    }  
}
```

# <ctype.h> (Χειρισμός χαρακτήρων) - Παραδείγματα

```
    }  
    printf("%s\n", strI);  
    printf("%s\n", strF);  
    return 0;  
}
```

θα εμφανιστεί στην κονσόλα

```
**programMatismos==2==  
Programmatismos II
```

# Δομή Διαφανειών

## 1 Εισαγωγή

## 2 Βιβλιοθήκες

- `<ctype.h>` (Χειρισμός χαρακτήρων)
- `<math.h>` (Μαθηματικές συναρτήσεις)
- `<stdlib.h>` (Διάφορες συναρτήσεις)
- `<time.h>` (Χειρισμός χρόνου)

# <math.h> (Μαθηματικές συναρτήσεις)

Μερικές από τις συναρτήσεις είναι

Συνάρτηση C	Μαθηματική συνάρτηση
<code>int abs(int a)</code>	$ \alpha $
<code>double fabs(double a)</code>	$ \alpha $
<code>double pow(double a, double b)</code>	$\alpha^b$
<code>double sqrt(double a)</code>	$\sqrt{\alpha}$
<code>double exp(double a)</code>	$e^{\alpha}$
<code>double log(double a)</code>	$\ln(\alpha)$
<code>double log10(double a)</code>	$\log(\alpha)$
<code>double sin(double a)</code>	$\sin(\alpha)$
<code>double cos(double a)</code>	$\cos(\alpha)$
<code>double tan(double a)</code>	$\tan(\alpha)$
<code>double cot(double a)</code>	$\cot(\alpha)$

# <math.h> (Μαθηματικές συναρτήσεις) - Παραδείγματα

Να υπολογιστεί η μαθηματική παράσταση

$$A = (\sin^2(x) + \cos^2(x)) + \sqrt{\ln(e^{2y})}$$

για  $x = 1$  και  $y = 2$ .

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

int main() {
    int x=1, y=2;
    float res1, res2;
    res1=pow(sin(x), 2)+pow(cos(x), 2);
    res2=sqrt(log(exp(2*y)));
```

# <math.h> (Μαθηματικές συναρτήσεις) - Παραδείγματα

```
printf("%f, %f\n", res1, res2);  
printf("%f\n", res1+res2);  
return 0;  
}
```

θα εμφανιστεί στην κονσόλα

```
1.000000, 2.000000  
3.000000
```



# Δομή Διαφανειών

## 1 Εισαγωγή

## 2 Βιβλιοθήκες

- `<ctype.h>` (Χειρισμός χαρακτήρων)
- `<math.h>` (Μαθηματικές συναρτήσεις)
- `<stdlib.h>` (Διάφορες συναρτήσεις)
- `<time.h>` (Χειρισμός χρόνου)

## <stdlib.h> (Διάφορες συναρτήσεις)

Μερικές από τις συναρτήσεις είναι

- `int rand()`

Επιστρέφει ένα τυχαίο αριθμό από το 0 έως το `RAND_MAX`.

- `void srand(unsigned int seed)`

Αρχικοποιεί την γεννήτρια τυχαίων αριθμών `int rand()`. Η τιμή που χρησιμοποιείται συνήθως για την αρχικοποίηση είναι η `time()`.

Δημιουργία διαστημάτων

- Για να δημιουργήσουμε τυχαίους ακέραιους αριθμούς σε ένα διάστημα  $[\alpha, b]$  με  $b \leq \text{RAND\_MAX} = \text{max}$  μπορούμε να χρησιμοποιήσουμε τον τύπο

$$[\alpha, b] = \text{mod}([0, \text{max}], (b - \alpha + 1)) + \alpha$$

## <stdlib.h> (Διάφορες συναρτήσεις)

- Για να δημιουργήσουμε τυχαίους πραγματικούς αριθμούς σε ένα διάστημα  $[\alpha, b]$  με  $b \leq RAND\_MAX$  μπορούμε να χρησιμοποιήσουμε τον τύπο

$$[\alpha, b] = \frac{[0, max]}{max} \cdot (b - \alpha) + \alpha$$

- Γενικά, με τον πολλαπλασιασμό μεγαλώνουμε το εύρος του διαστήματος και με την πρόσθεση μετατοπίζουμε το διάστημα

## <stdlib.h> (Διάφορες συναρτήσεις) -

### Παραδείγματα

Να δημιουργηθούν 10 τυχαίοι ακέραιοι αριθμοί στο διάστημα  $[10, 20]$  και να υπολογιστεί μέσω συνάρτησης ο μέσος όρος και να εμφανιστεί.

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

float fun(int arr[]);

int main() {
    int arr[10], i;
    float mo;
    for(i=0; i<10; i=i+1) {
        arr[i]=rand()%(20-10+1)+10;
        printf("%d\n", arr[i]);
    }
```

## <stdlib.h> (Διάφορες συναρτήσεις) - Παραδείγματα

```
    }  
    mo=fun(arr);  
    printf("%f\n", mo);  
    return 0;  
}  
  
float fun(int arr[]){  
    int sum=0, i;  
    for(i=0; i<10; i=i+1){  
        sum=sum+arr[i];  
    }  
    return ((float) sum/10);  
}
```

θα εμφανιστεί στην κονσόλα

# <stdlib.h> (Διάφορες συναρτήσεις) - Παραδείγματα

```
20  
11  
10  
16  
18  
13  
12  
10  
19  
14  
14.300000
```

## <stdlib.h> (Διάφορες συναρτήσεις) -

### Παραδείγματα

Να δημιουργηθούν 10 τυχαίοι πραγματικοί αριθμοί στο διάστημα  $[-10, 10]$  και να υπολογιστούν μέσω συνάρτησης το πλήθος των θετικών, των αρνητικών και των μηδενικών και να εμφανιστούν.

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

void fun(float arr[], int *ppos, int *pneg, int *pzer);

int main() {
    int i, pos=0, neg=0, zer=0;
    float arr[10];
    for(i=0; i<10; i=i+1) {
        arr[i]=((float) rand() / RAND_MAX) * (10 - (-10)) + (-10)
        ;
    }
}
```

## <stdlib.h> (Διάφορες συναρτήσεις) -

### Παραδείγματα

```
        printf("%f\n", arr[i]);
    }
    fun(arr, &pos, &neg, &zer);
    printf("Positive=%d\n", pos);
    printf("Negative=%d\n", neg);
    printf("Zeros=%d\n", zer);
    return 0;
}

void fun(float arr[], int *ppos, int *pneg, int *pzer) {
    int i;
    for(i=0; i<10; i=i+1) {
        if(arr[i]>0) {
            *ppos=*ppos+1;
        }
        else if(arr[i]<0) {
```



## <stdlib.h> (Διάφορες συναρτήσεις) - Παραδείγματα

```
        *pneg=*pneg+1;
    }
    else{
        *pzer=*pzer+1;
    }
}
```

θα εμφανιστεί στην κονσόλα

```
-9.999844
-7.369244
5.112106
-0.826997
0.655345
-5.620816
```

# <stdlib.h> (Διάφορες συναρτήσεις) - Παραδείγματα

```
-9.059108
```

```
3.577294
```

```
3.585929
```

```
8.693859
```

```
Positive=5
```

```
Negative=5
```

```
Zeros=0
```

# Δομή Διαφανειών

## 1 Εισαγωγή

## 2 Βιβλιοθήκες

- `<ctype.h>` (Χειρισμός χαρακτήρων)
- `<math.h>` (Μαθηματικές συναρτήσεις)
- `<stdlib.h>` (Διάφορες συναρτήσεις)
- `<time.h>` (Χειρισμός χρόνου)

## <time.h> (Χειρισμός χρόνου)

Μερικές από τις συναρτήσεις είναι

- `clock_t clock(void)`

Επιστρέφει τον χρόνο που έχει περάσει από την έναρξη του προγράμματος σε κύκλους του επεξεργαστή. Για να μετατραπεί σε δευτερόλεπτα διαιρούμε με την σταθερά `CLOCKS_PER_SEC`.

- `time_t time(time_t *ptr)`

Επιστρέφει τον αριθμό δευτερολέπτων που έχουν περάσει από τον δηλωθέντα χρόνο.

- `double difftime(time_t t1, time_t t2)`

Επιστρέφει την διαφορά των χρόνων `t1` και `t2` σε δευτερόλεπτα.

## <time.h> (Χειρισμός χρόνου) - Παραδείγματα

Να μετρήσετε και να εμφανίσετε τον χρόνο εκτέλεσης της αναδρομικής συνάρτησης Fibonacci για τον υπολογισμό της τιμής  $F_{45}$ .

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

long int fib_rec(int n);

int main() {
    long int res1, res2;
    int num=45;
    clock_t t1, t2;
    time_t tm1, tm2;
    double time_spent1, time_spent2;
    t1=clock();
    res1=fib_rec(num);
```

## <time.h> (Χειρισμός χρόνου) - Παραδείγματα

```
t2=clock();
time_spent1 = (double)(t2 - t1) / CLOCKS_PER_SEC;
tm1=time(0);
res2=fib_rec(num);
tm2=time(0);
time_spent2 = difftime(tm2,tm1);
printf("F(%d)= %ld\n",num,res1);
printf("F(%d)= %ld\n",num,res2);
printf("Time = %f \n",time_spent1);
printf("Time = %f \n",time_spent2);
return 0;
}

long int fib_rec(int n){
    if(n==1){
        return 0;
    }
}
```

## <time.h> (Χειρισμός χρόνου) - Παραδείγματα

```
    else if(n==2) {  
        return 1;  
    }  
    else{  
        return(fib_rec(n-1) + fib_rec(n-2));  
    }  
}
```

θα εμφανιστεί στην κονσόλα

```
F(45)= 701408733  
F(45)= 701408733  
Time = 6.926346  
Time = 7.000000
```

## <time.h> (Χειρισμός χρόνου) - Παραδείγματα

Να μετρήσετε και να εμφανίσετε τον χρόνο εκτέλεσης της αναδρομικής και κανονικής συνάρτησης Fibonacci για τον υπολογισμό της τιμής  $F_{45}$ .

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

long int fib(int n);
long int fib_rec(int n);

int main() {
    long int res1, res2;
    int num=45;
    clock_t t1, t2;
    double time_spent1, time_spent2;
    t1=clock();
```



## <time.h> (Χειρισμός χρόνου) - Παραδείγματα

```
    res1=fib(num);
    t2=clock();
    time_spent1 = (double)(t2 - t1) / CLOCKS_PER_SEC;
    t1=clock();
    res2=fib_rec(num);
    t2=clock();
    time_spent2 = (double)(t2 - t1) / CLOCKS_PER_SEC;
    printf("F(%d)= %ld\n",num,res1);
    printf("F(%d)= %ld\n",num,res2);
    printf("Time = %f \n",time_spent1);
    printf("Time = %f \n",time_spent2);
    return 0;
}

long int fib(int n){
    int i;
    long int res=0, term1=1, term2=0;
```

## <time.h> (Χειρισμός χρόνου) - Παραδείγματα

```
for (i=3; i<=n; i++)  
{  
    res=term1+term2;  
    term2=term1;  
    term1=res;  
}  
return (res);  
}  
  
long int fib_rec(int n) {  
    if (n==1) {  
        return 0;  
    }  
    else if (n==2) {  
        return 1;  
    }  
    else {
```

## <time.h> (Χειρισμός χρόνου) - Παραδείγματα

```
    return(fib_rec(n-1) + fib_rec(n-2));  
}  
}
```

```
F(45) = 701408733
```

```
F(45) = 701408733
```

```
Time = 0.000002
```

```
Time = 8.203845
```