

# C για επιστήμονες και μηχανικούς

Ελευθερία Κατσίρη

17/1/2020

# Εφαρμογές

1. Ρίψη ζαριού έξι πλευρών, 6000 φορές
2. Αναδρομική συνάρτηση υπολογισμού τετραγωνικής ρίζας
3. Επεξεργασία αρχείου κειμένου

# Παραγωγή τυχαίων αριθμών

- Παιχνίδια, προσομοιώσεις, στατιστική ανάλυση
- `stdlib.h`, `rand()`: παράγεται ακολουθία ψευδοτυχαίων ακεραίων στο διάστημα  $[0, \text{RAND\_MAX} > 32767]$
- Όμως: εκτελώντας το πρόγραμμα ξανά, θα παραχθεί η ίδια ακολουθία ακεραίων!
- `stdlib.h`, `srand()` : διασπείρει τη γεννήτρια τυχαίων αριθμών
- `seed`: σπόρος παραγωγής μιας νέας ακολουθίας
- `time.h`, `time()`

# Παράδειγμα

```
> srand(time(NULL))
```

```
> rand()
```

```
4497
```

```
> rand()
```

```
11439
```

# Ρίψη ζαριού έξι πλευρών

Ελέγχει αν η ρίψη ζαριού είναι πραγματικά τυχαία:

(Ίση συχνότητα για κάθε πλευρά)

- Αρχικοποίηση μετρητών
- Παραγωγή ψευδοτυχαίων με `srand()`, `rand()`
- Βρόχος `for` 6000 επαναλήψεων
- Αύξηση μετρητή `rollcount`
- Εντολή `switch` καταγράφει συμβάν για καθε μια από τις 6 εμφανίσεις για κάθε ρίψη

# Εύρος ψευδοτυχαίων αριθμών

## Ακέραιοι:

- $\text{rand()} \% b$  : δίνει ένα **ακέραιο** στην περιοχή  $[0, b-1]$
- $n = a + \text{rand()} \% b$  : ομοίως :  $[a, a+b-1]$
- $n = 1 + \text{rand()} \% 6$  // ρίψη ζαριού 6 πλευρών
- $n = 1 + \text{rand()} \% 2$  // ρίψη κέρματος 2 πλευρών

## Κινητής υποδιαστολής

- $x = x_1 + \text{rand()} * (x_2 - x_1) / (\text{double}) \text{RAND\_MAX}$   
δίνει **float** στην περιοχή  $[x_1, x_2]$
- $x = \text{rand()} / (\text{double}) \text{RAND\_MAX} : [0,1]$

# Εύρος ψευδοτυχαίων αριθμών

Face	Frequency
1	1003
2	994
3	996
4	1021
5	990
6	996

Σχετικές συχνότητες σχετικά ίδιες για κάθε εμφάνιση!

```
/* Αρχείο: srand.c
```

```
Τυχαία ρίψη ενός ζαριού έξι πλευρών, υπολογισμός της  
συχνότητας κάθε πλευράς */
```

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

```
#include <stdlib.h> /* για χρήση της srand() και rand()*/
```

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

```
int main() {
```

```
int face,          /* πλευρά της κάθε ρίψης */
```

```
int rollcount,    /* αριθμός μέτρησης ρίψεων */
```

```
freq1=0, freq2=0, freq3=0, /* συχνότητα κάθε ρίψεων*/
```

```
freq4=0, freq5=0, freq6=0;
```



```
/* Αρχείο: srand.c
```

```
Τυχαία ρίψη ενός ζαριού έξι πλευρών, υπολογισμός της  
συχνότητας κάθε πλευράς */
```

```
srand(time(NULL)); /* διασπορά της γεννήτριας  
τυχαίων αριθμών */
```

```
/* χρήση ενός βρόχου για τη ρίψη του ζαριού 600 φορές */
```

```
for (rollcount=1; rollcount <=6000; rollcount++){
```

```
/* παρήγαγε ένα τυχαίο αριθμό στο διάστημα [1,6] */
```

```
face = 1 + rand() % 6 ;
```

```
/* αποθήκευσε την πλευρά στη μεταβλητή συχνότητας */
```

```
switch(face) {  
    case 1: /* εάν η πλευρά είναι 1 */  
        freq1 +=1; /* αύξησε τη μεταβλητή συχνότητας για την  
                πλευρά 1 */  
        break;  
    case 2: /* εάν η πλευρά είναι 2 */  
        freq2 +=1; /* αύξησε τη μεταβλητή συχνότητας για την  
                πλευρά 2 */  
        break;  
    case 3: /* εάν η πλευρά είναι 3 */  
        freq3 +=1; /* αύξησε τη μεταβλητή συχνότητας για την  
                πλευρά 3 */  
        break;  
    .....  
} // end switch  
} // end for
```

```
/* εμφάνισε τη συχνότητα κάθε πλευράς */  
printf(" Face Frequency\n");  
printf(" 1 %d\n", freq1);  
printf(" 2 %d\n", freq2);  
printf(" 3 %d\n", freq3);  
printf(" 4 %d\n", freq4);  
printf(" 5 %d\n", freq5);  
printf(" 6 %d\n", freq6);  
return 0;  
}
```

# Υπολογισμός τετραγωνικής ρίζας

- Μέθοδος Newton
- Μαθηματική διαδικασία παραγωγής

$$x_{i+1} = \frac{1}{2} \left( x_i + \frac{a}{x_i} \right) \quad (1)$$

- Αρχική εκασία  $x_0$  για τη ρίζα
- Υπολογισμός διαδοχικών προσεγγιστικών ριζών
- $x_1, x_2, x_3, \dots, x_i, x_{i+1}, \dots$
- Κριτήριο σύγκλισης:  $|x_{i+1} - x_i| < FLT\_EPSILON$

- Συνθήκη τερματισμού επανάληψης
- Μικρό έψιλον μηχανής
- Το πρόγραμμα σταματά επίσης αν ο αριθμός των επαναλήψεων είναι μικρότερος από 100.
- Βρόχος **for** χρησιμοποιείται για τον αναδρομικό υπολογισμό της τετραγωνικής ρίζας
- Εξίσωση 1 :  $(x_2 = x_1 + a/x_1)/2.0$  ; (αναδρομική)
- Εάν το κριτήριο σύγκλισης ικανοποιηθεί η ροή θα εγκαταλείψει το βρόχο μέσω της εντολής **break**
- Ο αλγόριθμος μπορεί και να αποκλίνει (δε βρίσκεται λύση, **N=100**)

- Εκτυπώνονται:
  - Η υπολογιζόμενη τετραγωνική ρίζα
  - Η τετραγωνική ρίζα από τη μαθηματική συνάρτηση `sqrt()` και
  - ο αριθμός των επαναλήψεων.
- Έξοδος:

$$\text{sqrtx}(3.00) = 1.732051$$

$$\text{sqrt}(3.00) = 1.732051$$

$$\text{Number of iterations} = 5$$

```
/* Υπολογισμός της τετραγωνικής ρίζας sqrt(a) για
α=3.0 χρησιμοποιώντας τη μέθοδο του Newton */
#include <stdio.h>
#include <math.h> /* για fabs */
#include <float.h> /* για FLT_EPSILON */
define N 100 /* μέγιστος αριθμός επαναλήψεων */
int main(){
int ι;
double α, x0, x1, x2;
α=3.0; /* sqrt(a) με α=0.3 */
x0=α; /* η αρχική ικασία για το x0 */
    x1=x0; /* ορίσε το x1 στο x0 */
```

```
for (i=1; i <= N; i++) {  
    x2=(x1+a/x1)/2.0;  
    /* ο αναδρομικός τύπος του Newton */  
    if (fabs(x2-x1) < FLT_EPSILON ) break;  
    x1 = x2;  
    /* ενημέρωσε την τιμή x1 για την επόμενη  
    επανάληψη */  
}  
if(i < N) {  
    /* ο αριθμός των επαναλήψεων είναι μικρότερος  
    του N */
```



```
printf ( "sqrtx(%.2f) = %f\n", a, x2);
printf ( "sqrt(%.2f) = %f\n", a, sqrt(a));
printf ( "Number of iterations = %d\n", i);
}
else{
/* ο αριθμός των επαναλήψεων είναι ίσος με N */
    printf {"sqrtx failed to converge\n"};
}
return 0;
}
```

# Επεξεργασία αρχείου κειμένου

- Μέτρηση χαρακτήρων, γραμμών, λέξεων
- nc, nl, nw
- **Υπόθεση 1:** γραμμή είναι κάθε ακολουθία χαρακτήρων που τερματίζει με το χαρακτήρα νέας γραμμής '\n'
- **Υπόθεση 2:** λέξη είναι κάθε ακολουθία χαρακτήρων που δεν περιέχει 'κενό', 'στηλοθέτη' ή 'νέα γραμμή'

$(c \neq ' ' \ || \ c \neq '\n' \ || \ c \neq '\t')$

```
#include <stdio.h>

#define IN 1 /* inside a word */
#define OUT 0 /* outside a word */

/* count lines, words, and characters in input */
main()
{
int c, nl, nw, nc, state;
state = OUT;
nl = nw = nc = 0;
while ((c = getchar()) != EOF) {
```

```
++nc;
if (c == '\n')
++nl;
if (c == ' ' || c == '\n' || c == '\t')
state = OUT;
else if (state == OUT) {
state = IN;
++nw;
}
}
printf("%d %d %d\n", nl, nw, nc);
}
```

# 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
- Αλφαριθμητικά
- Δείκτες

# Παρουσίαση του περιεχομένου

*Τα κεφάλαια του βιβλίου που προτείνεται, είναι :*

## ❶ ΓΝΩΡΙΜΙΑ ΜΕ ΤΗ ΓΛΩΣΣΑ C

(Βασικές έννοιες και η δομή της γλώσσας)

## ❷ ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ C

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

## ❸ ΕΝΤΟΛΕΣ ΕΛΕΓΧΟΥ

(Περιγραφή των εντολών ελέγχου)

*Το πιο σημαντικό κεφάλαιο για ένα προγραμματιστή*

## ❹ ΣΥΝΑΡΤΗΣΕΙΣ

(Περιγραφή των συναρτήσεων, functions)

*Εδώ ανακαλύπτουμε τον πλούτο και την ευελιξία του προγραμματισμού με τη γλώσσα C*

# 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