

# Εργαστήριο Εισαγωγή στον Προγραμματισμό

Εργαστήριο 05

**Αναδρομή**

**Βασιλόπουλος Διονύσης**

**Ε.ΔΙ.Π. Τμήματος Πληροφορικής & Τηλεπικοινωνιών**

# 5<sup>ο</sup> Εργαστήριο

## Αναδρομή

Δύο βασικά σημεία:

1. Πότε σταματάει η αναδρομή (Συνθήκη τερματισμού)
2. Πότε η συνάρτηση καλεί τον εαυτό της

# 5<sup>ο</sup> Εργαστήριο

## Collatz\_rec.c

Μεταβλητή static

```
int collatz(int n){  
    static int steps=0;  
    int term;
```

```
    if (isodd(n))  
        term=n*3+1;  
    else  
        term=n/2;
```

Συνθήκη τερματισμού

```
    steps++;  
    if (n=1)  
        return(1);  
    else  
        collatz(term);  
}
```

Συνθήκη αναδρομής/επανάληψης

# 5<sup>ο</sup> Εργαστήριο

## static

A)Αν θέλουμε η εμβέλεια μίας μεταβλητής να είναι μόνο το αρχείο στο οποίο ορίζεται και να μην συγχέεται με κάποια εξωτερική μεταβλητή με το ίδιο όνομα, αλλά σε άλλο αρχείο, μπορούμε να προτάξουμε του ορισμού της (στην αρχή του file, μετά τα define) τον προσδιοριστή static, για παράδειγμα

```
static double sum;
```

B)Η εμβέλεια μίας **τοπικής** μεταβλητής που έχει ορισθεί σαν static **μέσα σε συνάρτηση**, είναι η συνάρτηση αυτή, και η τιμή της **διατηρείται** και μετά τον **τερματισμό της συνάρτησης**. Σε νέα κλήση της συνάρτησης η μεταβλητή έχει την τιμή που πήρε από την προηγούμενη κλήση της.

Σελίδα 66 σημειώσεις κ.Σταματόπουλου

# 5<sup>ο</sup> Εργαστήριο

## collatz – Αρχική τιμή 950000001

Επειδή ο 950000001 είναι περιττός από την 1<sup>η</sup> κλήση της collatz οδηγούμαστε στον όρο: 2.850.000.005

Ο τύπος int είναι όμως στο όριο -2.147.483.648 ... 2.147.483.647.

Οπότε, δοθέντος ότι οι όροι της collatz είναι θετικοί ακέραιοι η χρήση του τύπου unsigned int που είναι στο όριο 0 ... 4.294.967.295 είναι μας λύνει το πρόβλημα.

# 5<sup>ο</sup> Εργαστήριο

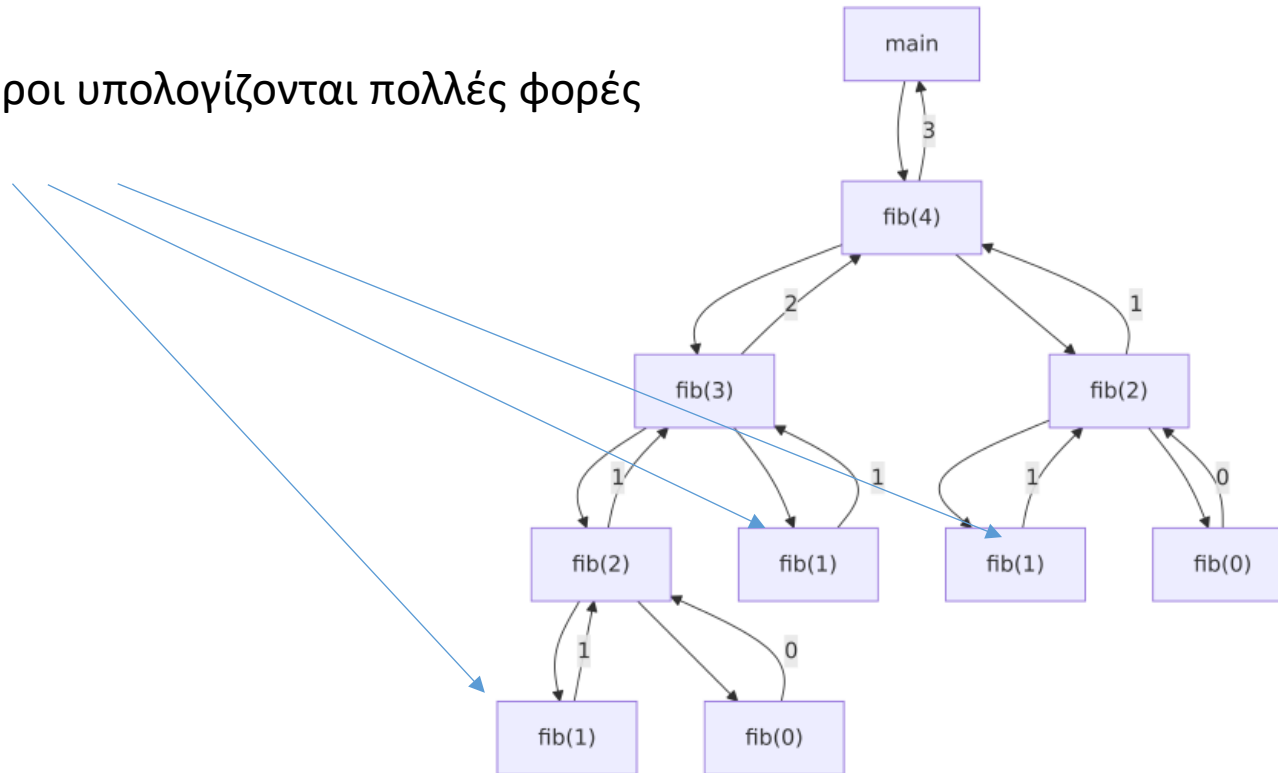
atoi

Συνάρτηση μετατροπής μιας συμβολοσειράς σε ακέραιο

# 5<sup>ο</sup> Εργαστήριο

## fibonacci

Αρκετοί όροι υπολογίζονται πολλές φορές



# 5<sup>ο</sup> Εργαστήριο

## Χρονικοί υπολογισμοί

#include <time.h> για χρήση βιβλιοθηκών χειρισμού time

```
clock_t start, end; //Ορισμός 2 μεταβλητών start, end που είναι τύπου clock_t
//αναφέρονται σε χτύπους του ρολογιού (clock) του επεξεργαστή)
```

```
start=clock();
```

```
..
```

```
.. //Κώδικας
```

```
..
```

```
end=clock();
```

start-end = χτύποι ρολογιού για την εκτέλεση του κώδικα.

CLOCKS\_PER\_SEC = χτύποι ρολογιού ανά δευτερόλεπτο

(start-end)/ CLOCKS\_PER\_SEC = δευτερόλεπτα που έχουν περάσει για τον εκτέλεση του κώδικα.