

## ΣΥΝΑΡΤΗΣΕΙΣ ΒΙΒΛΙΟΘΗΚΗΣ

1

## ΕΙΣΑΓΩΓΗ

- ❖ Στο κεφάλαιο αυτό, γίνεται κατηγοριοποίηση και πλήρης ανάλυση των συναρτήσεων βιβλιοθήκης, εκτός από εκείνες που αφορούν αρχεία.
- ❖ Τονίζεται ιδιαίτερα
  - ◆ αν κάποια συνάρτηση βιβλιοθήκης δεν εμφανίζεται σε κάποιο περιβάλλον (MS-Windows, Unix),
  - ◆ αν κάποια συνάρτηση διαφέρει **συντακτικά** από το ένα περιβάλλον στο άλλο (σπάνια).

2

## ΠΑΡΕΧΟΜΕΝΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

- ❖ Οι συναρτήσεις/κλήσεις C++ που υπάρχουν στη βιβλιοθήκη της, ονομάζονται συχνά **παρεχόμενες συναρτήσεις (reserved functions)**. Ομαδοποιούνται ως ακολούθως:
  - ❖ Διαχείριση buffer (buffer manipulation)
  - ❖ Διαχείριση χαρακτήρα (character classification)
  - ❖ Διαχείριση δεδομένων (data conversion)
  - ❖ Διαχείριση αρχείων (files and input/output)
  - ❖ Μαθηματικές συναρτήσεις (math-functions)
  - ❖ Διαχείριση μνήμης (memory control/ allocation)
  - ❖ Ελέγχου διαδικασίας (process control)
  - ❖ Διάταξης και έρευνας (sorting and searching)
  - ❖ Διαχείρισης string (string manipulation)
  - ❖ Διαχείρισης χρόνου (time-functions)
  - ❖ Διαχείρισης λίστας παραμέτρων (functions of arguments lists)
  - ❖ Άλλες συναρτήσεις (miscellaneous)

3

## ΣΥΝΑΡΤΗΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΧΑΡΑΚΤΗΡΑ 1/5

- ❖ Οι συναρτήσεις διαχείρισης ή προσδιορισμού χαρακτήρα (**character classification**) ελέγχουν αν ένας χαρακτήρας (1 byte) ανήκει στη μια ή στην άλλη κατηγορία Αλφάβητου ASCII. (π.χ. αν είναι μικρό γράμμα, αν είναι ψηφίο, κλπ.)
- ❖ Ορισμένες από αυτές τις συναρτήσεις είναι και **macros** και μετατρέπουν «μικρούς» σε «μεγάλους» χαρακτήρες και αντίστροφα.
- ❖ Στη συνέχεια δίνονται οι σημαντικότερες από αυτές τις κλάσεις.

4

## ΣΥΝΑΡΤΗΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΧΑΡΑΚΤΗΡΑ 2/5

### ❖ Συνάρτηση **isalnum**

❖ Επιστρέφει μηδέν αν ο χαρακτήρας δεν είναι ούτε γράμμα, ούτε ψηφίο (αλφαριθμητικός) δηλαδή από 'A'-'Z' ή 'a'-'z' ή '0'-'9'.

❖ Συντακτικός τύπος:

```
ret=isalnum(chr);
```

όπου αν **chr** είναι αλφαριθμητικός (ASCII) τότε το **ret** έχει μη μηδενική τιμή (τύπου int)

❖ Η ενσωμάτωση του σταθερού αρχείου **<ctype.h>** είναι αναγκαία, ενώ υπάρχει και σαν **macro** σε κάθε περιβάλλον.

❖ Προσοχή, θεωρείται αναγκαίο το αλφάβητο ASCII.

5

## ΣΥΝΑΡΤΗΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΧΑΡΑΚΤΗΡΑ 3/5

### ❖ Συνάρτηση **isalpha**

❖ Ελέγχει αν ένας χαρακτήρας είναι μικρό ή και κεφαλαίο λατινικό γράμμα.

❖ Τα υπόλοιπα ισχύουν όπως στη συνάρτηση **isalnum**.

### ❖ Συνάρτηση **isascii**

❖ Είναι η οδηγός (pilot) συνάρτηση όλης της κατηγορίας **isy**.

❖ Ελέγχει αν ο χαρακτήρας είναι ASCII ή όχι (π.χ. το EOF δεν είναι ASCII).

❖ Όλες οι συναρτήσεις της κατηγορίας αυτής λειτουργούν με χαρακτήρες ASCII ή με τον EOF (οριζόμενο στο **<stdio.h>**).

❖ Τα υπόλοιπα ισχύουν όπως στη συνάρτηση **isalnum**.

### ❖ Συνάρτηση **isctrl**

❖ Ελέγχει αν ο χαρακτήρας είναι ειδικός χαρακτήρας ελέγχου (control character), δηλ. αν είναι 0x00-0x1f ή 0x7f.

❖ Τα υπόλοιπα ισχύουν όπως στις άλλες συναρτήσεις της ίδιας κατηγορίας.

6

## ΣΥΝΑΡΤΗΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΧΑΡΑΚΤΗΡΑ 4/5

### ❖ Συνάρτηση **isdigit**

- ❖ Ελέγχει αν ο χαρακτήρας είναι ψηφίο ('0'-'9').
- ❖ Τα υπόλοιπα ισχύουν όπως στις άλλες συναρτήσεις της κατηγορίας **isy**.

### ❖ Συνάρτηση **isgraph**

- ❖ Ελέγχει αν ο χαρακτήρας είναι εκτυπούμενος εκτός του χαρακτήρα space (0x21-0x7e).
- ❖ Τα υπόλοιπα ισχύουν όπως στη συνάρτηση **isalnum**.

### ❖ Συνάρτηση **islower**

- ❖ Ελέγχει αν ο χαρακτήρας είναι μικρό λατινικό γράμμα ('a'-'z').
- ❖ Τα υπόλοιπα ισχύουν όπως στις άλλες συναρτήσεις της κατηγορίας αυτής.

### ❖ Συνάρτηση **isprint**

- ❖ Ελέγχει αν ο χαρακτήρας είναι εκτυπούμενος (0x20-0x7e).
- ❖ Τα υπόλοιπα ισχύουν όπως στις άλλες συναρτήσεις της κατηγορίας αυτής.

7

## ΣΥΝΑΡΤΗΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΧΑΡΑΚΤΗΡΑ 5/5

### ❖ Συνάρτηση **isspace**

- ❖ Ελέγχει για λευκούς χαρακτήρες (whitespace characters): 0x09-0x0d ή 0x20.
- ❖ Τα υπόλοιπα ισχύουν όπως στις άλλες συναρτήσεις της κατηγορίας αυτής.

### ❖ Συνάρτηση **ispunct**

- ❖ Ελέγχει αν ο χαρακτήρας είναι σημείο στίξης (punctuation character), δηλ. δεν είναι χαρακτήρας αλφαριθμητικός ή ελέγχου ή λευκός.
- ❖ Τα υπόλοιπα ισχύουν όπως στις άλλες συναρτήσεις της κατηγορίας αυτής.

### ❖ Συνάρτηση **isupper**

- ❖ Ελέγχει αν ο χαρακτήρας είναι κεφαλαίο λατινικό γράμμα ('A'-'Z').
- ❖ Τα υπόλοιπα ισχύουν όπως στις άλλες συναρτήσεις της κατηγορίας αυτής.

### ❖ Συνάρτηση **isxdigit**

- ❖ Ελέγχει για δεκαδεξαδικό ψηφίο, δηλ. από 'A'-'F' ή 'a'-'f' ή '0'-'9'.
- ❖ Τα υπόλοιπα ισχύουν όπως στις άλλες συναρτήσεις της κατηγορίας αυτής.

8

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 1/18

- ❖ Οι συναρτήσεις μετατροπής δεδομένων (**data conversion**) μετατρέπουν `string` σε αριθμητική τιμή (`int`, `long`, `float`, κλπ.) και αντίστροφα.
- ❖ Προϋποθέτουν την ενσωμάτωση του σταθερού αρχείου `<stdlib.h>`.
- ❖ Περιλαμβάνουν και ειδικές κλήσεις πολυ-χαρακτήρων σε άλλους ειδικούς χαρακτήρες, αλλά και από μικρά σε κεφαλαία και αντίστροφα.

9

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 2/18

### ❖ Συνάρτηση **atof**

❖ Μετατρέπει `string` σε πραγματικό διπλής ακρίβειας (`double`).

❖ Συντακτικός τύπος:

**ret=atof(string);**

όπου **string** (τύπου `char*`) μετατρέπεται στην τιμή **ret** (τύπου `double`).

❖ Σε περίπτωση λάθους η επιστρεφόμενη τιμή είναι μηδέν.

❖ Σε περίπτωση υπερχείλισης (`overflow`) η τιμή `ret` είναι απροσδιόριστη.

❖ Το `string` μπορεί να περιέχει τιμή σε εκθετική μορφή (με `d` ή `D` ή `e` ή `E`), λευκούς χαρακτήρες πριν ή μετά την τιμή, την τελεία και ένα τουλάχιστον στοιχείο πριν ή μετά από αυτή.

❖ Τα πρόσημα `+` ή `-` έχουν νόημα πριν την τιμή ή μετά τον εκθετικό χαρακτήρα.

❖ Προσοχή, η τελική τιμή μπορεί να έχει στρογγυλοποιηθεί, ιδιαίτερα σε εκθετικές μορφές.

10

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 3/18

### ❖ Συνάρτηση **atof** (συνέχεια)

❖ Παράδειγμα: έστω οι εντολές

```
.....  
char *str1="13.13", *str2="-7.8912654773d210";  
double x1, x2;  
x1=atof(str1); x2=atof(str2);  
printf("x1=%f x2=%e", x1, x2);  
.....
```

τότε ανάλογα με το περιβάλλον και το υλικό, η έξοδος θα είναι:

```
x1=13.13 x2=-7.891265e+210
```

❖ Η συνάρτηση **atof** υπάρχει, εκτός του `<stdlib.h>` και στο σταθερό αρχείο `<math.h>`.

11

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 4/18

### ❖ Συνάρτηση **atoi**

❖ Μετατρέπει `string` σε ακέραιο (τύπου `int`).

❖ Συντακτικός τύπος:

```
ret=atoi(string);
```

όπου η τιμή **string** μετατρέπεται στην τιμή **ret** (τύπου `int`).

❖ Σε περίπτωση λάθους η τιμή `ret` είναι μηδέν (εκτός και αν το `string` αντιστοιχούσε στο μηδέν).

❖ Σε περίπτωση υπερχείλισης (`overflow`) η τιμή `ret` είναι απροσδιόριστη.

❖ Το `string` μπορεί να περιέχει λευκούς χαρακτήρες πρόσημο και οπωσδήποτε ένα τουλάχιστον ψηφίο.

12

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 5/18

### ❖ Συνάρτηση **atoi** (συνέχεια)

❖ Παράδειγμα: έστω οι εντολές

```
.....  
char *str="-12345";  
int i;  
i=atoi(str);  
printf("i=%d", i);  
.....
```

τότε η έξοδος θα είναι:

```
i=-12345
```

13

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 6/18

### ❖ Συνάρτηση **atol**

❖ Μετατρέπει string σε ακέραιο τύπου long.

❖ Συντακτικός τύπος:

```
ret=atol(string);
```

όπου **string** μετατρέπεται στην τιμή **ret** (τύπου long).

❖ Σε περίπτωση λάθους το ret είναι 0.

❖ Σε περίπτωση υπερχείλισης (overflow) η τιμή ret είναι απροσδιόριστη.

❖ Τα υπόλοιπα ισχύουν όπως στη συνάρτηση **atoi**.

❖ Παράδειγμα: έστω οι εντολές

```
.....  
char *str="100121";  
printf("%ld", atol(str));  
.....
```

τότε στην οθόνη θα εμφανισθεί η τιμή: 100121

14

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 7/18

### ❖ Συνάρτηση **ecvt**

- ❖ Μετατρέπει μια πραγματική δεκαδική τιμή σε string, το οποίο τελειώνει με τον κενό χαρακτήρα '\0'.
- ❖ Συντακτικός τύπος:  
**ret=ecvt(value, digits, &intpart, &sign);**  
όπου η τιμή **value** (τύπου double) μετατρέπεται σε string.
- ❖ Παίρνοντας **digits** (τύπου int) το πολύ ψηφία από το **value** και η υποδιαστολή υπάρχει μετά το **intpart** (τύπου int) ψηφίο.
- ❖ Εάν το **sign** (τύπου int) είναι μηδέν, η τιμή **value** είναι θετική, αλλιώς αρνητική.
- ❖ Το **ret** είναι τύπου char\*, δηλ. string, όπου καταχωρείται σαν συμβολοσειρά η τιμή **value**.
- ❖ Αν ισχύει digits μεγαλύτερο του πλήθους ψηφίων του value, τότε το string γεμίζει από δεξιά με μηδενικά.  
Προσοχή, σε αντίθετη περίπτωση γίνεται στρογγυλοποίηση του τελευταίου ψηφίου μέσα στο string.

15

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 8/18

### ❖ Συνάρτηση **ecvt** (συνέχεια)

- ❖ Σε περίπτωση χρήσης συναρτήσεων, η συνάρτηση **ecvt** απαιτεί τη ενσωμάτωση του σταθερού αρχείου **<stdlib.h>**.
- ❖ Παράδειγμα: έστω οι εντολές

```
.....  
char *ret;  
int digits, intpart, sign;  
digits=7;  
ret=ecvt(1234.5678, digits, &intpart, &sign);  
.....
```

ΤΟΤΕ

- ❖ μέσα στο string ret υπάρχει «1234568»
- ❖ το intpart ισούται με 4 (δηλ. το ακέραιο μέρος σταματά στα 4 πρώτα ψηφία)
- ❖ το sign είναι μηδέν (η τιμή είναι θετική)

16



## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 9/18

### ❖ Συνάρτηση **ecvt** (συνέχεια)

- ❖ Δεν προσφέρει έτοιμη τη μετατροπή πραγματικής τιμής σε string.
- ❖ Ο χρήστης πρέπει αλγοριθμικά να σχηματίσει ένα δεύτερο string που να περιλαμβάνει και την τελεία (decimal point) και το αρνητικό πρόσημα, αν υπάρχει.
- ❖ Προσοχή, στις περιπτώσεις που η τελεία των δεκαδικών είναι αριστερά των ψηφίων, δηλ. όταν το ακέραιο μέρος της τιμής είναι μηδέν, η συνάρτηση **ecvt** είναι ακόμη πιο πολύπλοκη. Τότε η τιμή του **intpart** είναι μηδέν ή αρνητική.
- ❖ Με τις συναρτήσεις **sprintf** και **sscanf** η μετατροπή της τιμής σε συμβολοσειρά γίνεται με αυτόματο τρόπο.
  - ◆ Βλ. Κεφάλαιο 5.

17

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 10/18

### ❖ Συνάρτηση **fcvt**

- ❖ Μετατρέπει μια πραγματική τιμή σε string, το οποίο τελειώνει με τον κενό χαρακτήρα '\0'.
- ❖ Συντακτικός τύπος:  
**ret=fcvt(value, nafter, &intpart, &sign);**  
όπου τα πάντα ισχύουν όπως στη συνάρτηση **ecvt** με τη μόνη διαφορά ότι το **nafter** (τύπου **int**) δηλώνει το μέγιστο επιθυμητό πλήθος ψηφίων μετά την υποδιαστολή.
- ❖ Σε περίπτωση χρήσης συναρτήσεων, απαιτείται τη ενσωμάτωση του σταθερού αρχείου **<stdlib.h>**.

18

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 11/18

### ❖ Συνάρτηση **fcvt** (συνέχεια)

❖ Παράδειγμα: έστω οι εντολές

```
.....  
char *ret;  
int nafter, intpart, sign;  
nafter=9;  
ret=fcvt(123.123456789, nafter, &intpart, &sign);  
.....
```

τότε

- ◆ μέσα στο string ret υπάρχει «123123456789»
- ◆ το intpart ισούται με 3 (δηλ. η τελεία των δεκαδικών (υποδιαστολή) υπάρχει μετά τον τρίτο χαρακτήρα)
- ◆ το sign είναι μηδέν
- ◆ η τιμή είναι θετική

19

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 12/18

### ❖ Συνάρτηση **gcvt**

- ❖ Μετατρέπει μια πραγματική τιμή σε string, το οποίο τελειώνει με τον κενό χαρακτήρα '\0'.
- ❖ Η **gcvt** σε αντίθεση με τις συναρτήσεις **ecvt** και **fcvt** δίνει το τελικό string σε πλήρη-τελική μορφή.
- ❖ Συντακτικός τύπος:

**ret=gcvt(value, digits, buf);**

όπου

- ◆ η τιμή **value** (τύπου double) μετατρέπεται σε τελική μορφή στο **string** που καταχωρείται στο buffer **buf**
- ◆ το μέγιστο πλήθος ψηφίων που καταχωρείται ισούται με την τιμή του **digits** (τύπου int)
- ❖ Η καταχώρηση γίνεται στο FORTRAN format: xxx...x.yyy...y και αν αυτή δεν είναι δυνατή, ακολουθείται το FORTRAN E format, δηλ. η εκθετική μορφή καταχώρησης.

20

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 13/18

### ❖ Συνάρτηση **gcvt** (συνέχεια)

- ❖ Το **ret** είναι δείκτης (τύπου `char`) που δείχνει στο παραγόμενο `string`. Στην πράξη όμως δεν χρησιμοποιείται.
- ❖ Σε περίπτωση χρήσης συναρτήσεων, απαιτείται τη ενσωμάτωση του σταθερού αρχείου `<stdlib.h>`.
- ❖ Παράδειγμα: έστω οι εντολές

```
.....  
char buf[40];  
int digits=8;  
gcvt(-1.12345e4, digits, buf);  
.....
```

τότε το `buf` έχει το περιεχόμενο «-11234.5», δηλ. αν και η τιμή ήταν σε εκθετική μορφή η μετατροπή έγινε κανονικά σε F-δεκαδική μορφή.

21

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 14/18

### ❖ Συναρτήσεις **strtod** και **strtoul**

- ❖ Μετατρέπουν ένα `string` σε πραγματικό αριθμό διπλής ακρίβειας ή σε ακέραιο τύπου `long` αντίστοιχα.
- ❖ Συντακτικός τύπος της **strtod**:

**dvalue=strtod(str, &until);**

όπου

- ◆ το περιεχόμενο του `string str` μετατρέπεται στον αριθμό `dvalue` (τύπου `double`)
- ◆ το `string until` (τύπου `char*`) δείχνει το υπόλοιπο του `str` που δεν έλαβε μέρος στο μετασχηματισμό.
- ◆ το `until` μπορεί να αρχίζει με ένα οποιονδήποτε χαρακτήρα που δεν έχει θέση και νόημα σε πραγματική τιμή (πχ. με τον κενό χαρακτήρα `'\0'`, ένα γράμμα, κλπ.)

22

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 15/18

### ❖ Συναρτήσεις **strtod** και **strtol** (συνέχεια)

❖ Συντακτικός τύπος της **strtol**:

**lvalue=strtol(str, &until, base);**

όπου

- ◆ το **lvalue** που προκύπτει είναι τύπου long
- ◆ η μετατροπή σταματά αν διαβαστεί χαρακτήρας-ψηφίο μεγαλύτερο από το **base** (τύπου int)
- ◆ η τιμή **lvalue** είναι εκφρασμένη με βάση αρίθμησης την τιμή του **base**:
  - αν το **base** είναι από 2 έως 36 τότε χρησιμοποιείται ως βάση αρίθμησης,
  - αν το **base** είναι μηδέν, ο πρώτος χαρακτήρας του string **str** είναι μηδέν και ο δεύτερος 1 έως 7, τότε το **lvalue** προκύπτει με βάση το 8-δικό σύστημα,
  - αν το **base** είναι μηδέν, ο πρώτος χαρακτήρας του string **str** είναι μηδέν και ο δεύτερος 'x' ή 'X' τότε το **lvalue** προκύπτει με βάση το 16-δικό σύστημα,
  - αν το **base** είναι μηδέν, ο πρώτος χαρακτήρας του string **str** είναι από 1 έως 9 τότε το **lvalue** προκύπτει κανονικά με βάση το 10-δικό σύστημα.

23

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 16/18

### ❖ Συναρτήσεις **strtod** και **strtol** (συνέχεια)

- ❖ Οι δύο συναρτήσεις αναμένουν **str** με περιεχόμενο σε F-μορφή ή σε ακέραια μορφή (δεκαδική ή οκταδική ή δεκαεξαδική).
- ❖ Και οι δύο απαιτούν την ενσωμάτωση του σταθερού αρχείου **<stdlib.h>**.
- ❖ Παράδειγμα: έστω οι εντολές

```
.....  
char *str,*until,*stralso,*untill,*until2;  
double dvalue;  
long lvalue, lval2;  
int base=0;  
str="1234.56789STOP";  
stralso="10110134";  
dvalue=strtod(str, &until);  
lvalue=strtol(stralso, &untill, base);  
lval2=strtol(stralso, &until2, 2);  
.....
```

24

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 17/18

### ❖ Συναρτήσεις **strtod** και **strtol** (συνέχεια)

#### ❖ Παράδειγμα (συνέχεια):

τότε

- ◆ το dvalue θα ισούται με 1234.56789
- ◆ το string until θα έχει την τιμή STOP
- ◆ το lvalue θα έχει την τιμή 10110134
- ◆ το string until1 θα είναι κενό
- ◆ το lval2 θα έχει την τιμή 45
- ◆ το until2 θα έχει την τιμή 34

- ❖ Σε περίπτωση λάθους και οι δύο συναρτήσεις θέτουν το **errno** ίσο με τη συμβολική τιμή **ERANGE**.

25

## ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΤΑΤΡΟΠΗΣ ΔΕΔΟΜΕΝΩΝ 18/18

### ❖ Συνάρτηση (Macro) **tolower**

- ❖ Μετατρέπει έναν κεφαλαίο χαρακτήρα σε μικρό χαρακτήρα, διαφορετικά ο χαρακτήρας μένει αμετάβλητος.
- ❖ Αν γνωρίζουμε ότι ο χαρακτήρας είναι ήδη κεφαλαίος, τότε μπορεί να χρησιμοποιηθεί ισοδύναμα και η Macro **\_tolower**.
- ❖ Προσοχή, αν ο χαρακτήρας δεν είναι κεφαλαίος, τότε το αποτέλεσμα της συνάρτησης μένει απροσδιόριστο.

### ❖ Συνάρτηση (Macro) **toupper**

- ❖ Προκαλεί ακριβώς τα αντίθετα από την **tolower**.
- ❖ Αν γνωρίζουμε ότι ο χαρακτήρας είναι ήδη μικρός μπορεί να χρησιμοποιηθεί η Macro **\_toupper**.

26

## ΣΥΝΑΡΤΗΣΕΙΣ ΕΛΕΓΧΟΥ ΚΑΤΑΛΟΓΟΥ 1/5

- ❖ Οι συναρτήσεις ελέγχου καταλόγου (directory control) αφορούν το status του παρόντος καταλόγου εργασίας, την αλλαγή του ή την δημιουργία/ διαγραφή του.
- ❖ Απαιτούν την ενσωμάτωση του σταθερού αρχείου <direct.h> σε DOS/LAN περιβάλλον ή κάποιου άλλου αντίστοιχου αρχείου σε περιβάλλον UNIX (ή και καμία ενσωμάτωση).
- ❖ Ορισμένες από αυτές υπάρχουν μόνο σε επίπεδο MS-Visual C++ και μάλιστα σαν macros.
- ❖ Στη συνέχεια αναλύονται οι σημαντικότερες από αυτές.

27

## ΣΥΝΑΡΤΗΣΕΙΣ ΕΛΕΓΧΟΥ ΚΑΤΑΛΟΓΟΥ 2/5

- ❖ Συνάρτηση **chdir**
  - ◆ Τη συναντάμε σε κάθε περιβάλλον
  - ◆ Αλλάζει τον τρέχοντα κατάλογο εργασίας σε κάποιον άλλο
  - ◆ Έχει αναλυθεί πλήρως στο κεφ. 5
  - ◆ Υπάρχει σαν **\_chdir** σε MS-Visual C++

28

## ΣΥΝΑΡΤΗΣΕΙΣ ΕΛΕΓΧΟΥ ΚΑΤΑΛΟΓΟΥ 3/5

### ❖ Συνάρτηση **getcwd**

- ❖ Τη συναντάμε σε κάθε περιβάλλον
- ❖ Δείχνει το πλήρες όνομα του παρόντος καταλόγου εργασίας
- ❖ Συντακτικός τύπος:

**ret=getcwd(buf, nbytes);**

όπου

- ◆ το όνομα του παρόντος καταλόγου αποθηκεύεται στο string **buf** (το οποίο συνιστάται να είναι μήτρα χαρακτήρων)
- ◆ το **ret** είναι string και ταυτίζεται με το **buf**, εκτός αν έχει σημειωθεί λάθος, οπότε έχει την τιμή NULL
- ◆ το μέγιστο πλήθος των bytes που αποθηκεύονται στο **buf** δίνονται από την τιμή **nbytes** (τύπου int)

29

## ΣΥΝΑΡΤΗΣΕΙΣ ΕΛΕΓΧΟΥ ΚΑΤΑΛΟΓΟΥ 4/5

### ❖ Συνάρτηση **getcwd** (συνέχεια)

- ❖ Παράδειγμα: έστω οι εντολές

.....

```
char buf[41];  
char *pointd;  
pointd=getcwd(buf, 40);
```

.....

τότε

- ◆ Το πλήρες όνομα του παρόντος καταλόγου εργασίας θα υπάρχει στο string **pointd** το οποίο θα έχει το πολύ 40 bytes.
- ◆ Σε περίπτωση λάθους τίθεται μια από τις ακόλουθες συμβολικές τιμές στο **errno**:
  - ENOMEM : δεν υπάρχει διαθέσιμη μνήμη
  - ERANGE : το όνομα του καταλόγου είναι μεγαλύτερο από **nbytes** bytes.

30

## ΣΥΝΑΡΤΗΣΕΙΣ ΕΛΕΓΧΟΥ ΚΑΤΑΛΟΓΟΥ 5/5

### ❖ Συναρτήσεις **mkdir** και **rmdir**

- ❖ Υπάρχουν αποκλειστικά σε περιβάλλον Microsoft.
- ❖ Χρησιμοποιούνται για τη δημιουργία νέου καταλόγου ή για τη διαγραφή υπάρχοντος καταλόγου αντίστοιχα.

31

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 1/14

- ❖ Η C++ διαθέτει μια ικανοποιητική ποικιλία μαθηματικών τύπων, και θεωρείται σχεδόν ισοδύναμη με την κλασσική FORTRAN.
- ❖ Πολλές από τις μαθηματικές συναρτήσεις απαιτούν τον εφοδιασμό του κώδικα με βοηθητικό σε floating point στη διάρκεια του compilation.
- ❖ Όλες οι μαθηματικές συναρτήσεις απαιτούν την ενσωμάτωση του σταθερού αρχείου **<math.h>**.

32



## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 2/14

### ❖ Συνάρτηση **acos**

- ❖ Επιστρέφει το τόξο συνημιτόνου ενός πραγματικού τύπου double
- ❖ Συντακτικός τύπος:

$$y = \text{acos}(x);$$

όπου

- ◆ **y** τύπου double παίρνει τιμές στο διάστημα από μηδέν έως και  $\pi$
- ◆ **x** μεταξύ των τιμών -1 και 1
- ◆ σε περίπτωση λάθους τίθεται το **errno** ίσο με τη συμβολική τιμή **EDOM**, τυπώνεται **DOMAIN error** στο stderr και επιστρέφεται η τιμή μηδέν
- ◆ τα **x** και **y** μπορούν να είναι απλοί τύποι float αντί για double

33

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 3/14

### ❖ Συνάρτηση **acos** (συνέχεια)

- ❖ Παράδειγμα: έστω η εντολή

```
y=acos(-1.0);
```

τότε το y ισούται με 3,141593.

Αν δοθεί όμως η εντολή

```
y=acos(5);
```

τότε το errno θα τεθεί ίσο με EDOM και θα τυπωθεί στο stderr (πχ. οθόνη):

```
acos:DOMAIN error
```

Στην περίπτωση αυτή το y θα ισούται με μηδέν και το errno θα πρέπει να τεθεί ίσο με μηδέν στη συνέχεια της διαδικασίας

34

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 4/14

### ❖ Συνάρτηση **asin**

- ❖ Επιστρέφει το τόξο ημιτόνου ενός πραγματικού τύπου `double`
- ❖ Συντακτικός τύπος:

$$y = \text{asin}(x);$$

όπου

- ◆ **y** τύπου `double` στο διάστημα  $-\pi/2$  έως  $\pi/2$
- ◆ **x** μεταξύ των τιμών  $-1$  και  $1$
- ◆ τα υπόλοιπα ισχύουν όπως στη συνάρτηση **acos**.

35

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 5/14

### ❖ Συνάρτηση **atan**

- ❖ Επιστρέφει το τόξο εφαπτομένης  $x$
- ❖ Συντακτικός τύπος:

$$y = \text{atan}(x);$$

όπου

- ◆ **y** στο διάστημα  $-\pi/2$  έως  $\pi/2$
- ◆ **x** τύπου `double`
- ◆ τα υπόλοιπα ισχύουν όπως στη συνάρτηση **acos**.

36

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 6/14

### ❖ Συναρτήσεις **cos** και **cosh**

- ❖ Επιστρέφουν το συνημίτονο και το υπερβολικό συνημίτονο κάποιας τιμής τύπου **double**, αντίστοιχα
- ❖ Συντακτικοί τύποι:

**y=cos(x);**  
**y=cosh(x);**

όπου

- ◆ **y** τύπου **double**
- ◆ εάν το **x** είναι μεγάλο μπορεί να χαθεί ένα μέρος ακρίβειας
  - Στην περίπτωση της συνάρτησης **cos** αν χαθεί λίγη ακρίβεια δημιουργείται ένα **PLOSS error** και το **errno** τίθεται ίσο με **ERANGE**.  
Αν χαθεί αρκετή ακρίβεια τυπώνεται στο **stderr** ένα **TLOSS error**.
  - Στην περίπτωση της συνάρτησης **cosh** αν το **y** είναι πολύ μεγάλο, επιστρέφεται η τιμή **HUGE** και το **errno** τίθεται ίσο με **ERANGE**.

37

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 7/14

### ❖ Συνάρτηση **exp**

- ❖ Επιστρέφει την τιμή  $e^x$  ενός αριθμού **x** τύπου **double**
- ❖ Συντακτικός τύπος:

**y=exp(x);**

όπου

- ◆ **y** τύπου **double**
- ◆ Στην περίπτωση **overflow** του **y** επιστρέφεται η τιμή **HUGE** και το **errno** τίθεται ίσο με **ERANGE**.
- ◆ Σε περίπτωση **underflow** του **y** επιστρέφεται απλά η τιμή μηδέν.

### ❖ Συνάρτηση **fabs**

- ❖ Επιστρέφει την απόλυτη τιμή του **x** τύπου **double**
- ❖ Συντακτικός τύπος:

**y=fabs(x);**

όπου **y** επίσης τύπου **double**

38

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 8/14

### ❖ Συναρτήσεις **log** και **log10**

- ❖ Υπολογίζουν φυσικό και δεκαδικό λογάριθμο, αντίστοιχα, μιας δεδομένης τιμής την τιμή  $e^x$  ενός αριθμού  $x$  τύπου `double`
- ❖ Συντακτικοί τύποι:

```
z=log(x);  
z10=log10(x);
```

όπου

- ◆ **z**, **z10** και **x** τύπου `double`
- ◆ Σε περίπτωση λάθους,
  - αν το  $x$  είναι αρνητικό, οι συναρτήσεις τυπώνουν στο `stderr` ένα **DOMAIN** error και επιστρέφουν το αρνητικό **HUGE**
  - αν το  $x$  είναι μηδέν, τυπώνουν ένα **SIGN** errorσε κάθε περίπτωση το `errno` τίθεται ίσο με **EDOM**.

39

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 9/14

### ❖ Συναρτήσεις **log** και **log10** (συνέχεια)

- ❖ Παράδειγμα: έστω οι εντολές

```
double y1,y2,x=1000.0;  
y1=log(x);  
y2=log10(x);
```

τότε το  $y1$  είναι 6.907755 και το  $y2$ , 3.0

40

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 10/14

### ❖ Συνάρτηση **modf**

- ❖ Διαχωρίζει μια δεκαδική τιμή σε δύο άλλες, όπου η πρώτη έχει μόνο το δεκαδικό μέρος και η δεύτερη μόνο το ακέραιο
- ❖ Συντακτικός τύπος:

**z=modf(x,&n);**

όπου

- ◆ **x, z, n** τύπου double
  - ◆ **z** το δεκαδικό μέρος του **x**
  - ◆ **n** το ακέραιο μέρος του **x**
  - ◆ τα **z** και **n** έχουν το ίδιο πρόσημο με το **x**
- ❖ Παράδειγμα: έστω οι εντολές

```
double z,x,n;  
x=-3123.1234;  
z=modf(x,&n);
```

τότε το **z** είναι -0.1234 και το **n** είναι -3123.0

41

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 11/14

### ❖ Συνάρτηση **pow**

- ❖ Επιστρέφει την τιμή  $x^y$
- ❖ Συντακτικός τύπος:

**z=pow(x,y);**

όπου

- ◆ **x, y, z** τύπου double
- ◆ **z** η επιστρεφόμενη τιμή  $x^y$
- ◆ Αν το **x** είναι μηδέν και **y** αρνητικό τίθεται **errno** ίσο με **ERANGE** και επιστρέφεται η τιμή **HUGE**
- ◆ Αν το **x** είναι αρνητικό και το **y** μη ακέραιο τυπώνεται **DOMAIN error** στο **stderr**, τίθεται **errno** ίσο με **EDOM** και επιστρέφεται η τιμή μηδέν
- ◆ Σε περίπτωση overflow τίθεται το **errno** ίσο με **ERANGE** και επιστρέφεται η τιμή (+ ή -) **HUGE**

42

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 12/14

### ❖ Συνάρτηση **pow** (συνέχεια)

- ❖ Παράδειγμα: έστω οι εντολές

```
double x,y,z;  
x=2.0;  
y=2.0;  
z=pow(x,y);
```

τότε το z ισούται με 4.0

### ❖ Συναρτήσεις **sin** και **sinh**

- ❖ Επιστρέφουν το ημίτονο και το υπερβολικό ημίτονο κάποιας τιμής τύπου double, αντίστοιχα
- ❖ Συντακτικοί τύποι:

```
y1=sin(x);  
y2=sinh(x);
```

όπου **y1, y2, x** τύπου double

43

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 13/14

### ❖ Συνάρτηση **sqrt**

- ❖ Υπολογίζει την τετραγωνική ρίζα δεδομένης τιμής τύπου double
- ❖ Συντακτικός τύπος:

```
z=sqrt(x);
```

όπου

- ◆ **z** τύπου double και ίσο με την τετραγωνική ρίζα του x
- ◆ Αν το x είναι αρνητικό τότε δημιουργείται **DOMAIN** error στο stderr, τίθεται **errno** ίσο με **EDOM** και επιστρέφεται η τιμή μηδέν

- ❖ Παράδειγμα: έστω οι εντολές

```
double z,x;  
x=25.0;  
z=sqrt(x);
```

τότε το z ισούται με 5.0

44

## ΜΑΘΗΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ 14/14

### ❖ Συναρτήσεις **tan** και **tanh**

❖ Υπολογίζουν την εφαπτομένη και την υπερβολική εφαπτομένη κάποιας τιμής τύπου `double`, αντίστοιχα

❖ Συντακτικοί τύποι:

**z1=tan(x);**

**z2=tanh(x);**

όπου **z1**, **z2**, **x** τύπου `double`