

Εισαγωγή στον Προγραμματισμό

Ενότητα C++ (75%) :

4^η εβδομάδα

Παναγιώτης Τζουνάκης

(βασισμένο στις σημειώσεις που ευγενικά προσέφερε ο καθηγητής κ. Γεώργιος Ραχώνης)

Φθινόπωρο 2021

Βιβλιογραφία

Βιβλιογραφία μαθήματος (Εύδοξος)

1. Βιβλίο [12550424]: Εισαγωγή στην Fortran 90/95/2003, Καραμπετάκης Νικόλαος
[Λεπτομέρειες](#)
2. Βιβλίο [22771795]: Προγραμματίζοντας με Fortran 90, Γράψα Θεοδούλα [Λεπτομέρειες](#)
3. Βιβλίο [18549112]: Σύγχρονη Fortran 95/2003, Λάζος Κωνσταντίνος Ε. [Λεπτομέρειες](#)
4. Βιβλίο [13536]: FORTRAN 77/90/95 ΚΑΙ FORTRAN 2003, ΑΛΕΞΑΝΔΡΟΣ Σ. ΚΑΡΑΚΟΣ
[Λεπτομέρειες](#)
5. Βιβλίο [86056659]: C++: Από τη Θεωρία στην Εφαρμογή, Γ. Τσελίκης [Λεπτομέρειες](#)
<http://www.cplusplus.gr>
6. Βιβλίο [12247]: C++ Βήμα προς Βήμα, Schildt Herbert [Λεπτομέρειες](#)
7. Βιβλίο [13598]: ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ ΤΗ C++, ROBERT LAFORE [Λεπτομέρειες](#)
8. Βιβλίο [50655981]: C++, 9η Έκδοση, Savitch Walter [Λεπτομέρειες](#)

Επιπρόσθετη βιβλιογραφία για μελέτη

- Walter Savitch, Problem Solving with C++, Addison-Wesley (10th edition) 2018.
- David Harel, Yishai Feldman, Algorithmics, The Spirit of Computing, Addison-Wesley (3rd edition) 2004.
- T. H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithm”, MIT Press (3rd edition) 2009.
- Bjarne Stroustrup, Η Γλώσσα Προγραμματισμού C++, 4^η εκδ., Εκδόσεις Κλειδάριθμος 2014.

Last resort: (e.g. google) search for “C++ tutorial” or “C++ tutorial greek” or “C++ εγχειρίδιο”

Ενδεικτικές διαδικτυακές πηγές για προγραμματισμό με C++

- <http://ocw.uc3m.es/ingenieria-informatica/programming-in-c-language-2013/IntroductiontoDevCIDE.pdf>
- <https://developers.google.com/edu/c++>
- <http://greenteapress.com/thinkcpp/thinkCScpp.pdf>
- <https://rooksguide.files.wordpress.com/2013/12/rooks-guide-isbn-version.pdf>
- <https://www.linuxlinks.com/excellent-free-books-learn-c-plus-plus/>
- <https://www.ece.uvic.ca/~frodo/cppbook/> (textbook-exercises, slides, SDE, videos, etc.)
- <https://www.e-booksdirectory.com/details.php?ebook=2518>
- <https://www.materials.uoc.gr/el/undergrad/courses/ETY215/notes.pdf> (στα **Ελληνικά!**)

C++

- 1980 Bjarne Stroustrup
C with Classes
- 1983 Κυκλοφορεί έξω από ερευνητικά
ιδρύματα

περισσότερα: π.χ. <https://en.wikipedia.org/wiki/C++>

Το αλφάβητο της C++

- **Γράμματα:** Κεφαλαία και μικρά Αγγλικά γράμματα

A,B,...,Z,a,b,...,z

- **Ψηφία**

0,1,2,3,4,5,6,7,8,9

- **Ειδικοί χαρακτήρες**

! # \$ % & ' () * + - / < > : ;

Κενό . = \ [] ^ { } ? “ |

Το λεξιλόγιο της C++

- Το λεξιλόγιο της C++ περιλαμβάνει 8 κατηγορίες: *ονόματα, αριθμούς, χαρακτήρες, αλυσίδες χαρακτήρων, τελεστές, διαχωριστές, λέξεις κλειδιά και σχόλια.*

Όνόματα

- Περιέχουν 1-(32) αλφαριθμητικούς χαρακτήρες (τα 26 γράμματα της αγγλικής αλφαβήτου, τα 10 ψηφία και το σύμβολο _).
- Ξεκινούν πάντα με αγγλικό γράμμα ή το χαρακτήρα υπογράμμισης.
- Κεφαλαία και πεζά γράμματα θεωρούνται διαφορετικά.
- Δεν επιτρέπεται η χρήση ονομάτων που έχουν ειδική σημασία για τη C++ (λέξεις κλειδιά).

Τι δεν επιτρέπεται

- Δεν επιτρέπονται σύμβολα εκτός από το «_».
- Δεν επιτρέπονται τα κενά.
- Δεν επιτρέπονται οι Ελληνικοί χαρακτήρες.

Παραδείγματα

Επιτρεπτά ονόματα

AF1, DAY, Day, FATHER_NAME

Μη επιτρεπτά ονόματα

1DC, ΩΡΑ, FATHER-NAME

Αριθμοί

- *Ακέραιοι αριθμοί.*

$[-32768, 32767]$

$[-2147483648, 2147483647]$

$[0, 65535]$ (ακέραιοι χωρίς πρόσημο)

$[0, 2^{32}-1]$

- *Πραγματικοί αριθμοί.*

Απλής ακριβείας

$[-3.4 \times 10^{38}, -3.4 \times 10^{-38}]$ $[3.4 \times 10^{-38}, 3.4 \times 10^{38}]$

πλήθος σημαντικών ψηφίων: 7

Διπλής ακριβείας

$[-1.7 \times 10^{308}, -1.7 \times 10^{-308}]$ $[1.7 \times 10^{-308}, 1.7 \times 10^{308}]$

πλήθος σημαντικών ψηφίων: 15

Διπλής ακριβείας

$[-1.7 \times 10^{4932}, -1.7 \times 10^{-4932}]$ $[1.7 \times 10^{-4932}, 1.7 \times 10^{4932}]$

πλήθος σημαντικών ψηφίων: 19

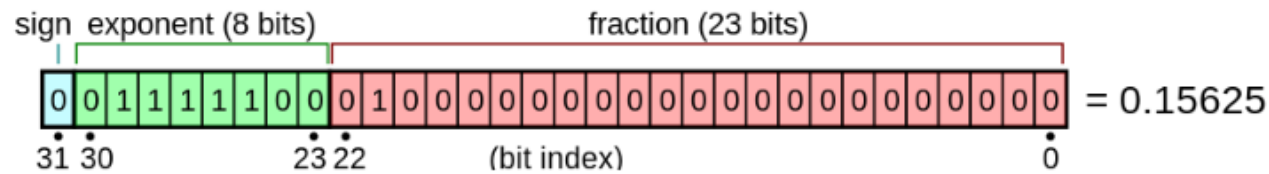
Οι πραγματικοί αριθμοί χωρίζονται σε δύο κατηγορίες :

- **Πραγματικοί αριθμοί σταθερής υποδιαστολής** (θετικοί ή αρνητικοί αριθμοί που περιέχουν υποδιαστολή) π.χ. 34.5

(https://en.wikipedia.org/wiki/Fixed-point_arithmetic)

- **Πραγματικοί αριθμοί κινητής υποδιαστολής** (θετικοί ή αρνητικοί αριθμοί που είναι εκφρασμένοι σε εκθετική μορφή $xxxxE\pm xxxx$ όπου η βάση είναι θετικός ή αρνητικός ακέραιος ή δεκαδικός αριθμός, ενώ ο εκθέτης είναι θετικός ή αρνητικός ακέραιος αριθμός) π.χ.
- Ο αριθμός $0.345E+2$ συμβολίζει τον 34.5
- Ο αριθμός $0.345E-2$ συμβολίζει τον 0.00345

IEEE 754 single-precision binary floating-point format: binary32



The real value assumed by a given 32-bit *binary32* data with a given *sign*, biased exponent e (the 8-bit unsigned integer), and a 23-bit *fraction* is

$$(-1)^{b_{31}} \times 2^{(b_{30}b_{29}\dots b_{23})_2 - 127} \times (1.b_{22}b_{21}\dots b_0)_2,$$

which yields

$$\text{value} = (-1)^{\text{sign}} \times 2^{(E-127)} \times \left(1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} \right).$$

In this example:

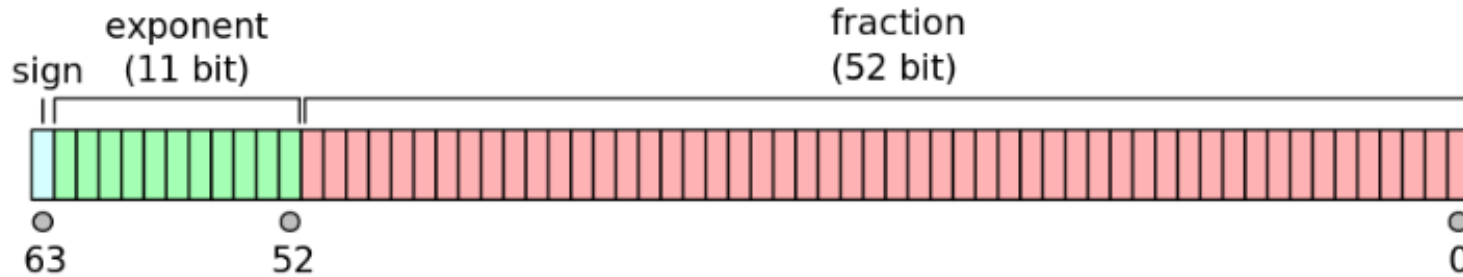
- $\text{sign} = b_{31} = 0,$
- $(-1)^{\text{sign}} = (-1)^0 = +1 \in \{-1, +1\},$
- $E = b_{30}b_{29}\dots b_{23} = \sum_{i=0}^7 b_{23+i} 2^{+i} = 124 \in \{1, \dots, (2^8 - 1) - 1\} = \{1, \dots, 254\},$
- $2^{(E-127)} = 2^{124-127} = 2^{-3} \in \{2^{-126}, \dots, 2^{127}\},$
- $1.b_{22}b_{21}\dots b_0 = 1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} = 1 + 1 \cdot 2^{-2} = 1.25 \in \{1, 1 + 2^{-23}, \dots, 2 - 2^{-23}\} \subset [1; 2 - 2^{-23}] \subset [1; 2).$

thus:

$$\text{value} = (+1) \times 2^{-3} \times 1.25 = +0.15625.$$

https://en.wikipedia.org/wiki/Single-precision_floating-point_format

IEEE 754 double-precision binary floating-point format: binary64



The real value assumed by a given 64-bit double-precision datum with a given **biased exponent** e and a 52-bit fraction is

$$(-1)^{\text{sign}} (1.b_{51}b_{50}\dots b_0)_2 \times 2^{e-1023}$$

or

$$(-1)^{\text{sign}} \left(1 + \sum_{i=1}^{52} b_{52-i} 2^{-i} \right) \times 2^{e-1023}$$

Χαρακτήρες

- Όλοι οι χαρακτήρες που υποστηρίζονται από τον Η/Υ μέσα σε αποστρόφους π.χ. ‘a’, ‘γ’. Ισχύουν τα εξής :
 - Για να δηλώσουμε το χαρακτήρα απόστροφο πρέπει να τον γράψουμε μετά από το \ δηλαδή \’.
 - Ο χαρακτήρας ‘A’ είναι διαφορετικός από το χαρακτήρα ‘a’.

- Ο χαρακτήρας ‘7’ διαφέρει από τον αριθμό 7. Ο πρώτος δε μπορεί να πάρει μέρος σε αριθμητικές πράξεις ενώ ο δεύτερος μπορεί.
- Πρέπει να δοθεί προσοχή στο διαχωρισμό του μηδέν 0 και του κεφαλαίου όμικρον O.

Αλυσίδες χαρακτήρων

Πρόκειται για ακολουθία χαρακτήρων μεταξύ διπλών αποστρόφων π.χ. “Μαθηματικά”, “123”.

Προσέξτε ότι το ‘α’ είναι διαφορετικό από το “α”.

Τελεστές

Είναι σύμβολα που δηλώνουν πράξεις μεταξύ τελεστέων, δηλαδή αριθμών, αλυσίδων χαρακτήρων κ.λ.π. .

Υπάρχουν 3 κατηγορίες τελεστών: αριθμητικοί, σύγκρισης και λογικοί :

- *Τελεστές αριθμητικοί*

Χρησιμοποιούνται για πράξεις μεταξύ αριθμών.

Τελεστής	Λειτουργία	Σύνταξη
+	Πρόσθεση	$X+Y$
-	Αφαίρεση	$X-Y$
*	Πολλαπλασιασμός	$X*Y$
/	Διαίρεση	X/Y
%	Υπόλοιπο ακέραιας διαίρεσης	$X\%Y$

- *Τελεστές Σύγκρισης*

Χρησιμοποιούνται για σύγκριση μεταξύ αριθμών ή αλυσίδων χαρακτήρων.

Τελεστής	Λειτουργία	Σύνταξη
==	Ισότητα	$X==Y$
!=	Ανισότητα	$X!=Y$
>	Μεγαλύτερο	$X>Y$
<	Μικρότερο	$X<Y$
>=	Μεγαλύτερο ή ίσο	$X>=Y$
<=	Μικρότερο ή ίσο	$X<=Y$

- *Τελεστές Λογικοί*

Χρησιμοποιούνται για την εκτέλεση λογικών πράξεων.

Τελεστής

Λειτουργία

!

Λογική άρνηση

&&

Λογική πρόσθεση

||

Διάζευξη

Οι πίνακες αληθείας των λογικών τελεστών είναι :

X	Y	X&&Y	X Y	!X
1	1	1	1	0
1	0	0	1	0
0	1	0	1	1
0	0	0	0	1

Διαχωριστές

Πρόκειται για χαρακτήρες με συγκεκριμένη λειτουργία στη C++. Οι κυριότεροι είναι :

= Καταχώριση

() Παρενθέσεις

; Δήλωση τέλους εντολής

Λέξεις - Κλειδιά

Πρόκειται για λέξεις με ειδική σημασία στη C++:

asm	auto	break	case
catch	char	class	const
continue	default	delete	do
double	else	enum	extern
float	for	friend	goto
if	inline	int	long
new	operator	private	protected
public	register	return	short
signed	sizeof	static	struct
switch	template	this	throw
try	typedef	union	unsigned
virtual	void	violate	while

Σχόλια

Οποιοιδήποτε χαρακτήρες ακολουθούν το σύμβολο // (μέχρι το τέλος της γραμμής) θεωρούνται σχόλια και δεν μεταφράζονται από τον compiler της C++.

Παράδειγμα: // Αυτό είναι σχόλιο

Άλλο Παράδειγμα:

/ Αυτό είναι ένα σχόλιο
που πιάνει πολλές
γραμμές! */*

Κατηγορίες δεδομένων

Μεταβλητές. Δεδομένα που η τιμή τους μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος.

Πιο συγκεκριμένα, με τον όρο **μεταβλητή** εννοούμε μια (ή παραπάνω) θέση η οποία δημιουργείται στη μνήμη του Η/Υ, για να δεχτεί ένα συγκεκριμένο τύπο δεδομένων, και η οποία έχει ένα χαρακτηριστικό όνομα που πληροί τους κανόνες που θέσαμε προηγούμενα.

Τύποι δεδομένων

Όνομα

Σύνολο τιμών

int

ακέραιοι στο διάστημα
[-32768, 32767]

long int

ακέραιοι στο διάστημα
[-2147483648, 2147483647]

unsigned int

ακέραιοι στο διάστημα
[0, 65535]

unsigned long

ακέραιοι στο διάστημα
[0, $2^{32}-1$]

float $[-3.4 \times 10^{38}, -3.4 \times 10^{-38}] [3.4 \times 10^{-38}, 3.4 \times 10^{38}]$

double $[-1.7 \times 10^{308}, -1.7 \times 10^{-308}] [1.7 \times 10^{-308}, 1.7 \times 10^{308}]$

long double

$[-1.7 \times 10^{4932}, -1.7 \times 10^{-4932}] [1.7 \times 10^{-4932}, 1.7 \times 10^{4932}]$

char χαρακτήρες ή ακέραιοι στο
διάστημα -128 έως 127

Εκφράσεις στη C++

Τύπος	Τελεστής	Σειρά προτεραιότητας για τελεστές με την ίδια προτεραιότητα
Αριθμητικός	pow(x,y) * / % + -	Αριστερά προς Δεξιά A προς Δ
Σύγκρισης	> >= < <= == !=	A προς Δ A προς Δ A προς Δ
Λογικοί	! &&	Δ προς A A προς Δ

Μερικές παρατηρήσεις για τις παραστάσεις στην C++

- Προσοχή στις πράξεις που έχουν ίδια προτεραιότητα, π.χ. η παράσταση $a/b/c/d$ είναι ισοδύναμη με την $((a/b)/c)/d$.
- Δεν μπορούμε να βάλουμε δύο αριθμητικούς τελεστές τον ένα δίπλα στον άλλο, π.χ. $3*-4$, πρέπει να χρησιμοποιούμε παρενθέσεις $3*(-4)$.
- Οπουδήποτε έχετε αμφιβολία για την σειρά εκτέλεσης των πράξεων **χρησιμοποιείτε παρενθέσεις**

Μαθηματικές συναρτήσεις στη C++

Όνομα	Περιγραφή
<code>fabs(x)</code>	Απόλυτη τιμή του x
<code>sqrt(x)</code>	Τετραγωνική ρίζα του x
<code>sin(x)</code>	Ημίτονο του x σε ακτίνια
<code>asin(x)</code>	Τόξο ημίτονου του x
<code>cos(x)</code>	Συνημίτονο του x σε ακτίνια
<code>acos(x)</code>	Τόξο συνημίτονου του x
<code>tan(x)</code>	Εφαπτομένη του x σε ακτίνια
<code>atan(x)</code>	Τόξο εφαπτομένης του x
<code>exp(x)</code>	e^x

log(x)	Φυσικός λογάριθμος του x
log10(x)	Δεκαδικός λογάριθμος του x
sinh(x)	Υπερβολικό ημίτονο του x
cosh(x)	Υπερβολικό συνημίτονο του x
tanh(x)	Υπερβολική εφαπτομένη του x
pow(x,y)	x^y

“Hello World (again!)”

```
// my first C++ program
```

```
#include <iostream>
```

```
int main() {
```

```
    std::cout << "Hello World!";
```

```
    std::cout << "\nJust printed a(n  
invisible) newline character!";
```

```
    return 0;
```

```
}
```

Ασκήσεις

Να μετατρέψετε τις παρακάτω αριθμητικές εκφράσεις, σε μορφή αποδεκτή από τη γλώσσα C++.

Ελέγξτε την ορθότητα της έκφρασης ενθέτοντας την μέσα στο πρόγραμμα “Hello World” (με ένα ‘;’ στο τέλος της γραμμής) πριν δείτε τη λύση...

$$x - \eta \mu^2 (x)$$

$$\varepsilon \phi^2 (x - 1) + 1$$

$$x - \eta \mu^2 (x)$$

$$\varepsilon \phi^2 (x - 1) + 1$$

$$(x - \text{pow}(\sin(x), 2)) / (\text{pow}(\tan(x - 1), 2) + 1)$$

“Hello World (again!)” E1

```
// my first C++ program with an expression
```

```
#include <iostream>
```

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

```
#define x 0.5
```

```
int main() {
```

```
    (x-pow(sin(x),2))/(pow(tan(x-1),2)+1);
```

```
    std::cout << "Hello World!\n";
```

```
    return 0;
```

```
}
```

$$\left| \frac{1 + \sqrt{x - 2}}{x^2 - 1} \right|$$

$$\left| \frac{1 + \sqrt{x - 2}}{x^2 - 1} \right|$$

`fabs((1+sqrt(x-2))/(pow(x,2)-1))`

“Hello World (again!)” E2

```
// my first C++ program with an expression
#include <iostream>
#include <math.h>
#define x 0.5
int main() {
    fabs((1+sqrt(x-2))/(pow(x,2)-1));
    std::cout << "Hello World!\n";
    return 0;
}
```

$$e^{\log(x)} - \frac{y}{x^3}$$

$$e^{\log(x)} - \frac{y}{x^3}$$

`exp(log10(x))-y/pow(x,3)`

“Hello World (again!)” E3

```
// my first C++ program with an expression
#include <iostream>
#include <math.h>
#define x 0.5
#define y 0.6
int main() {
    exp(log10(x))-y/pow(x,3); //useless?
    std::cout << "Hello World!\n";
    return 0;
}
```

$$|x - y| \leq 1$$

$$|x - y| \leq 1$$

`fabs(x-y)<=1`

$$-1 \leq \varepsilon \phi(x) \leq 3$$

$$-1 \leq \varepsilon \phi(x) \leq 3$$

$$(\tan(x) \geq -1) \&\& (\tan(x) \leq 3)$$

Δομή προγράμματος στη C++

```
#include <header file>
using namespace std;
Πρωτότυπο συνάρτησης
int main( )
{
    Δηλώσεις μεταβλητών
    Εκτελέσιμες εντολές
    return 0;
}
Δήλωση συνάρτησης
{
    Δηλώσεις μεταβλητών
    Εκτελέσιμες εντολές
}
```

Η εντολή `#include` ζητάει από τον μεταγλωττιστή να ενσωματώσει ένα αρχείο (το header file) από την βιβλιοθήκη της C++ στο πρόγραμμά μας, π.χ.

```
#include <iostream>
```

Η εντολή εκτύπωσης `cout`

Γενική μορφή:

```
cout<<a;
```

```
cout<<a<<b<<c<<...<<x;
```

όπου `a,b,c,...,x` μπορεί να είναι σταθερές (αριθμητικές, χαρακτήρων ή συμβολοσειρές), μεταβλητές ή παραστάσεις της C++.

Παράδειγμα

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++. ”<<“It is
    really simple!”;
    return 0;
}
```

Αποτέλεσμα: Θα εκτυπωθεί το μήνυμα:

This is my first program in C++. It is really simple!

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++. ”;
    cout <<“It is really simple!”;
    return 0;
}
```

Αποτέλεσμα: Πάλι θα εκτυπωθεί το μήνυμα:

This is my first program in C++. It is really simple!

Πως θα τυπώσουμε τις δύο προτάσεις σε ξεχωριστές σειρές;

Το σύμβολο `\n` και η λέξη `endl`

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.\nIt is
    really simple!”;
    return 0;
}
```

Αποτέλεσμα: Θα εκτυπωθεί το μήνυμα:

This is my first program in C++.

It is really simple!

Την ίδια εκτύπωση θα δώσει και η εκτέλεση των παρακάτω προγραμμάτων:

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.”;
    cout <<“\nIt is really simple!”;
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.”
        <<“\n”<<“It is really simple”;
    return 0;
}
```

Παρατήρηση: Η C++ αντιμετωπίζει το \n σαν ένα χαρακτήρα

Η χρήση της λέξης endl

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.”<<endl;
    cout <<“It is really simple!”;
    return 0;
}
```

ή

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.”<<endl<<“It is
    really simple!”;
return 0;
}
```

Τι αποτέλεσμα θα είχε το πρόγραμμα;

```
#include <iostream>
using namespace std;
int main( )
{
    cout <<“This is my first program in C++.”<<endl;
    cout <<“\nIt is really simple!”;
    return 0;
}
```

Αποτέλεσμα:

This is my first program in C++.

It is really simple!

Μπορούμε να έχουμε διευθετημένη εκτύπωση με χρήση του χειριστή εξόδου `setw(k)`, όπου το k μπορεί να είναι ακέραιος θετικός, ή μεταβλητή ή παράσταση που παίρνουν τιμές ακέραιες θετικές. Το k καθορίζει το πλάτος (πλήθος χαρακτήρων) της επόμενης τιμής εξόδου και η τιμή αυτή θα είναι τοποθετημένη στις επόμενες k θέσεις στα δεξιά, π.χ.


```
#include <iostream>
#include <iomanip> //header file απαραίτητο για το setw(k)
using namespace std;
int main ( )
{
    cout <<setw(14)<<“THESSALONIKI”<<setw(7)<<7<<endl;
    cout <<setw(14)<<“ATHENS”<<setw(7)<<8<<endl;
    cout <<setw(14)<<“LARISA”<<setw(7)<<9;
    return 0;
}
```

Αποτέλεσμα:

THESSALONIKI	7
ATHENS	8
LARISA	9

Με το επόμενο πρόγραμμα «υπολογίζουμε» τις λύσεις της εξίσωσης δευτέρου βαθμού:

```
#include <iostream>
#include <math.h> //header file απαραίτητο για τις
//μαθηματικές συναρτήσεις. Προσοχή στην επέκταση h του
//ονόματος!
using namespace std;
int main ( )
{
    float a,b,c;
    float x1,x2;
    a=1;
    b=-5;
    c=6;
    x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
    x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
    cout <<“The solutions are:”<<“\nx1=”
        <<x1<<“\nx2= ”<<x2;
    return 0;
}
```

```

#include <iostream>
#include <math.h>
using namespace std;
int main ( )
{
    float a=2;           //Είναι δυνατό να καταχωρούμε τιμές
    float b,c;           //σε μεταβλητές ταυτόχρονα με τη
                        //δήλωσή τους

    float x1,x2;
    b=-5;
    c=6;
    x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
    x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
    cout <<"The solutions are:"<<"\nx1="
         <<x1<<"\nx2= " <<x2;

    return 0;
}

```

Με ποιο τρόπο θα κάνουμε το προηγούμενο πρόγραμμα πιο γενικό και χρήσιμο, ώστε ο χρήστης να δίνει τιμές στις a , b , c , και αυτό να επιστρέφει τις λύσεις της εξίσωσης;

Χρειαζόμαστε λοιπόν ένα τρόπο με τον οποίο να καθοδηγούμε το πρόγραμμα να ζητάει από το χρήστη τα δεδομένα.

Η εντολή **cin**

Γενική μορφή:

```
cin >>a;
```

```
cin >>a>>b>>...>>x;
```

όπου a,b,...,x είναι μεταβλητές.

```
#include <iostream>
#include <math.h>
using namespace std;
int main ( )
{
    float a,b,c;
    float x1,x2;
    cin >>a>>b>>c;    //δοκίμασε a=1, b=0, c= -1

    x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
    x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
    cout <<“The solutions are: ”<<“\nx1=”
         <<x1<<“\nx2= ”<<x2;
    return 0;
}
```

Μια «πιο φιλική» προς το χρήστη μορφή του προηγούμενου προγράμματος:

```
#include <iostream>
#include <math.h>
using namespace std;
int main ( )
{
    float a,b,c;
    float x1,x2;
    cout <<“Assign values to a,b,c”;
    cin >>a>>b>>c;                //δοκίμασε a=1, b=0, c= -1
    x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
    x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
    cout <<“The solutions are:”<<“\nx1=”
        <<x1<<“\nx2= ”<<x2;
    return 0;
}
```

Μια «ακόμη πιο φιλική» προς το χρήστη μορφή του προηγούμενου προγράμματος:

```
#include <iostream>
#include <math.h>
using namespace std;
int main ( )
{
    float a,b,c;
    float x1,x2;
    cout <<“Assign value to a:”;
    cin >>a;
    cout <<“Assign value to b:”;
    cin >>b;
    cout <<“Assign value to c:”;
    cin >>c;
    x1=(-b+sqrt(pow(b,2)-4*a*c))/(2*a);
    x2=(-b-sqrt(pow(b,2)-4*a*c))/(2*a);
    cout <<“The solutions are:”<<“\nx1=”
        <<x1<<“\nx2= ”<<x2;
    return 0;
}
```


Ασκήσεις W4.1

1) Θεωρούμε ότι ο κορμός ενός δένδρου είναι κυλινδρικός. Συνεπώς, ο όγκος του δένδρου V αν γνωρίζουμε την περίμετρό της βάσης του P , και το ύψος του H , θα δίνεται από τον τύπο :

$$V = \frac{P^2 \times H}{4 \times \pi}$$

όπου $\pi=3.14$.

Να γράψετε το κατάλληλο πρόγραμμα που θα υπολογίζει το V με δεδομένα τα H , P .

Ασκήσεις W4.2

2) Καταθέτουμε σε μια τράπεζα στην αρχή του κάθε έτους ένα σταθερό ποσό A . Αν η κατάθεση γίνεται με επιτόκιο $E\%$, τότε το κεφάλαιο K μετά την n -οστή κατάθεση θα είναι :

$$K = \frac{A \times (1 + E) \times \left[(1 + E)^N - 1 \right]}{E}$$

Να γραφεί πρόγραμμα υπολογισμού του K αν είναι γνωστά τα A , E και N .

Ασκήσεις W4.3

3) Έστω d_0 , d_1 οι διαμέτροι στη βάση και στο σημείο αποκορύφωσης ενός δένδρου. Αν το συνολικό μήκος του δένδρου είναι l , τότε ο όγκος V του δένδρου δίνεται από τον τύπο :

$$V = \frac{\pi}{4} \times l \times \left(\frac{d_0^2 + d_1^2}{2} \right)$$

Να γραφεί πρόγραμμα που να υπολογίζει τον όγκο του δένδρου, με δεδομένα το ύψος και τις δύο διαμέτρους.