ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ & ΥΠΟΛΟΓΙΣΤΙΚΗ ΦΥΣΙΚΗ

Μέρος 1ο

ΝΙΚΟΛΑΟΣ ΣΤΕΡΓΙΟΥΛΑΣ

ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΓΙΑΤΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ;

Στα μαθηματικά και στη φυσική συχνά έχουμε να κάνουμε **υπολογισμούς** που ακολουθούν μια συγκεκριμένη **διαδικασία**

ΠΑΡΑΔΕΙΓΜΑ: Εύρεση ρίζας

Έστω ότι θέλουμε να υπολογίσουμε τη $\sqrt{2}$. Ένας τρόπος είναι να σχηματίσουμε την εξίσωση

$$f(x) = x^2 - 2 = 0$$

Αν η ρίζα είναι η $x = x_0 \quad (= 1.414213562...)$

τότε το ανάπτυγμα Taylor γύρω από τη ρίζα είναι

$$f(x) = \cancel{f(x_0)}^{0} + f'(x_0)(x - x_0) + ...$$

ΓΙΑΤΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ;

Άρα

$$x_0 \simeq x - f(x)/f'(x_0)$$

και αν στην περιοχή της ρίζας $f'(x) \simeq f'(x_0)$

τότε

$$x_0 \simeq x - f(x)/f'(x)$$

και μπορούμε να δημιουργήσουμε την **αναδρομική σχέση** (μέθοδος Newton-Raphson)

$$x^{(n+1)} = [x - (f/f')]^{(n)}$$

Ξεκινάμε με μια αρχική εκτίμηση, π.χ.

$$n = 1 : x^{(1)} = 1.0$$

ΓΙΑΤΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ;

και υπολογίζουμε διαδοχικά:

n	$x^{(n)}$	$x^{(n+1)}$	f(x)
1	1.0	1.5	2.5e-01
2	1.5	1.41667	7e-03
3	1.41667	1.414216	6e-06

Η παραπάνω διαδικασία είναι ένας ΑΛΓΟΡΙΘΜΟΣ, που μας επέτρεψε να υπολογίσουμε $\sqrt{2} \simeq 1.414216$

ΑΝΑΛΥΣΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Ο αλγόριθμος Newton-Raphson για το συγκεκριμένο παράδειγμα έχει την εξής δομή:

$n = 1$ — αρχικοποίηση δείκτη

$x^{(n)} = 1.0$ — αρχικοποίηση μεταβλητής

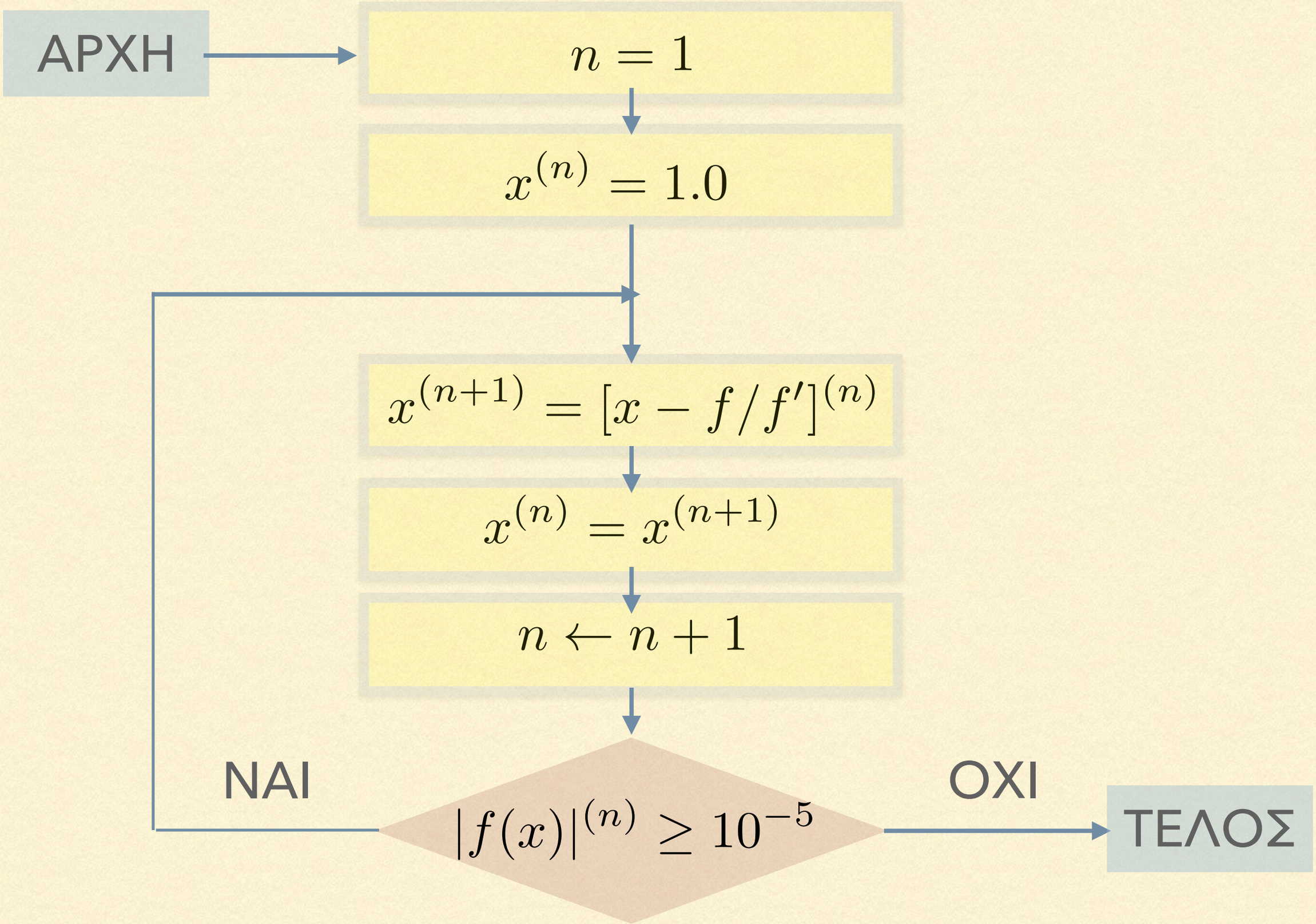
$x^{(n+1)} = [x - f/f']^{(n)}$ — υπολογισμός νέας τιμής

$x^{(n)} = x^{(n+1)}$ — αντικατάσταση παλαιάς τιμής από νέας

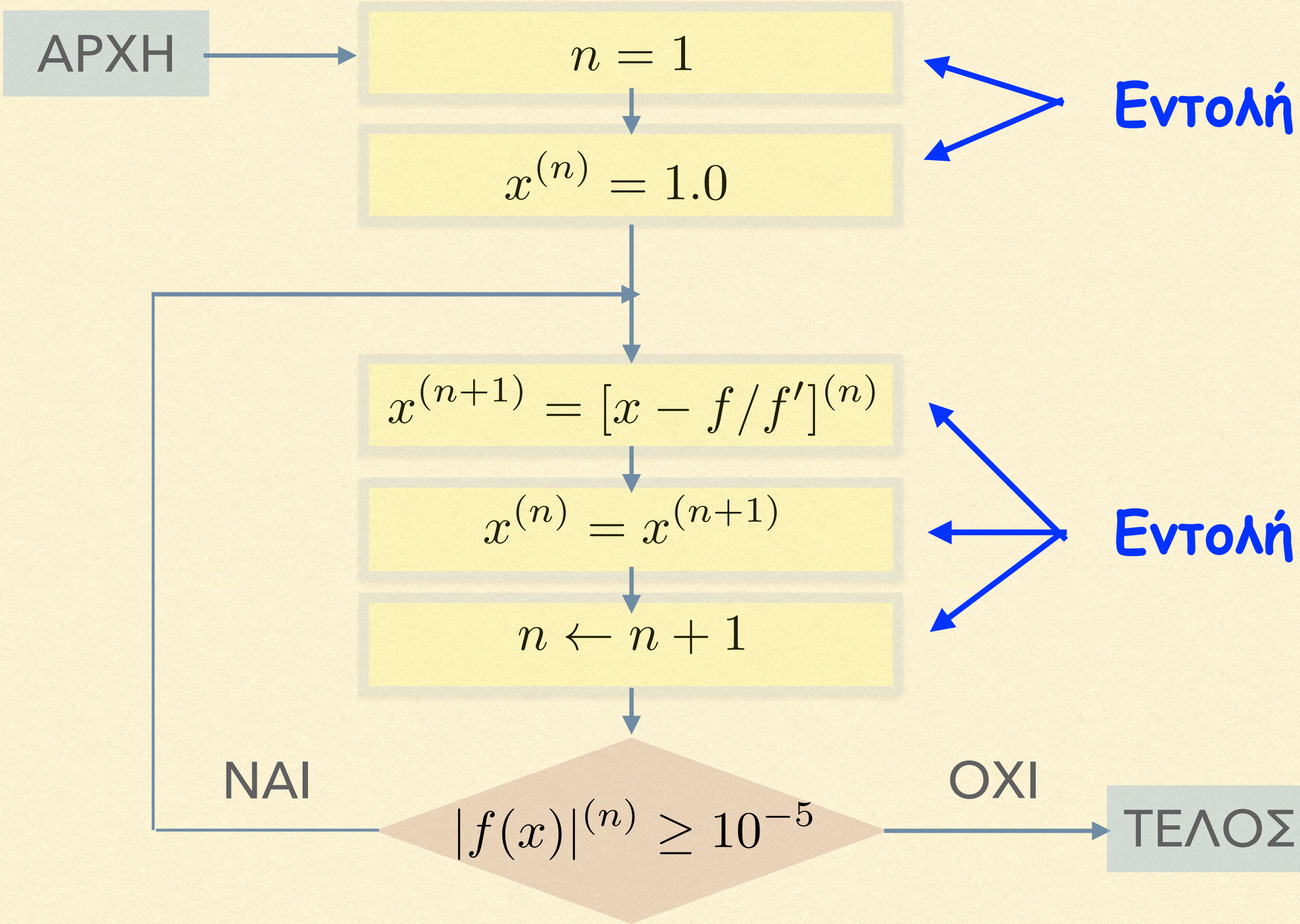
$n \leftarrow n + 1$ — αύξηση δείκτη

$|f(x)|^{(n)} < 1e - 5$ — συνθήκη τερματισμού

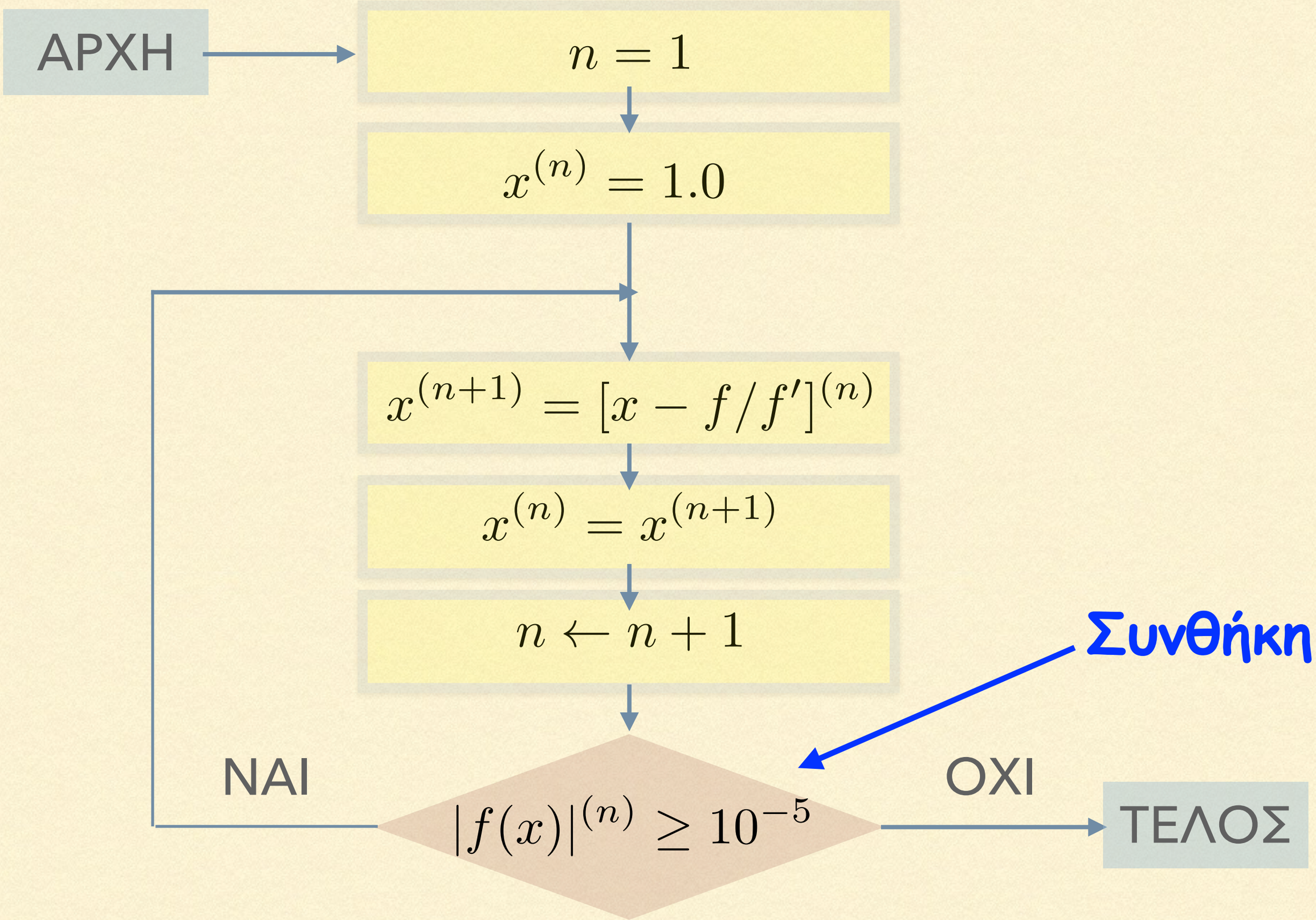
ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ



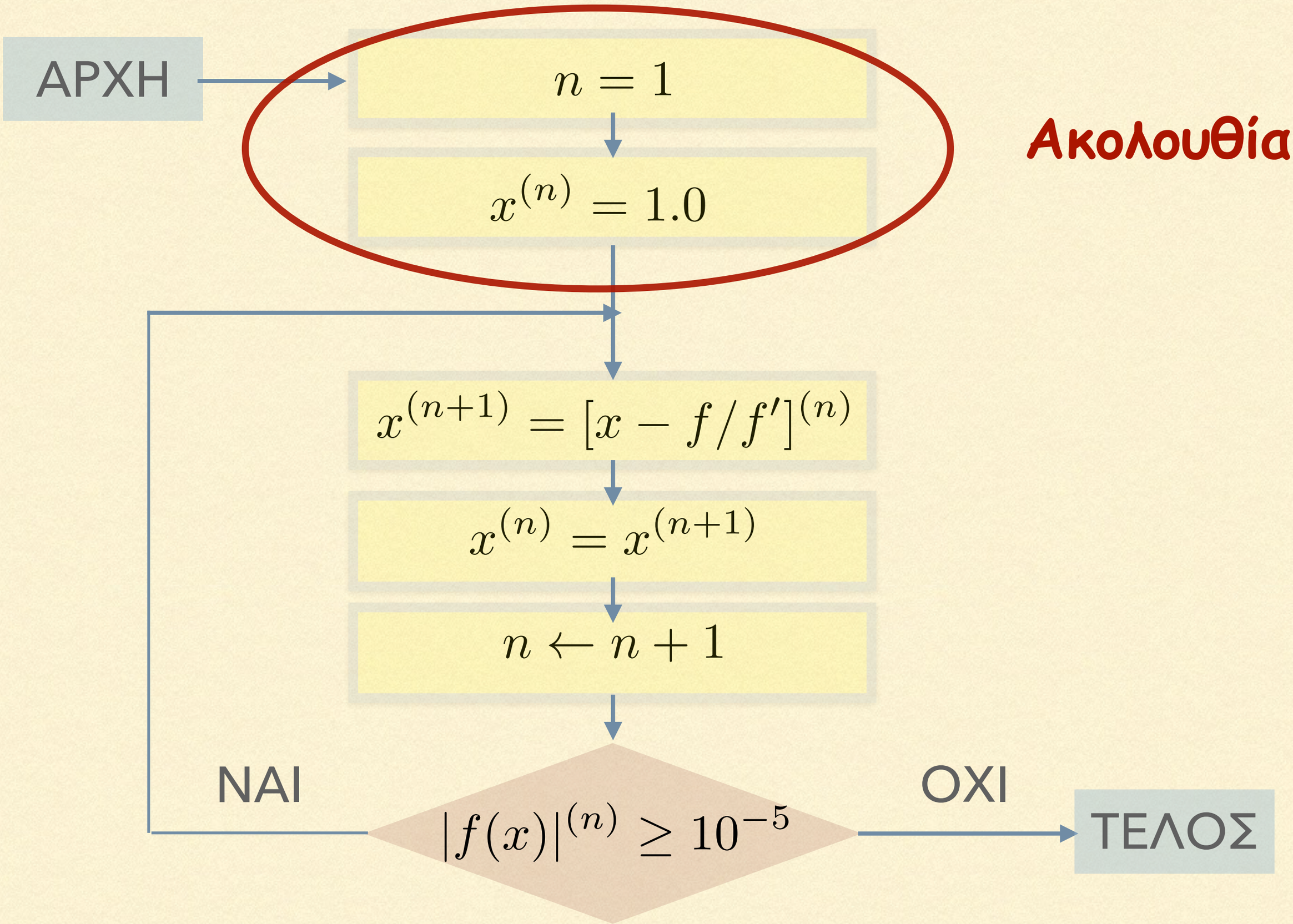
ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ



ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ

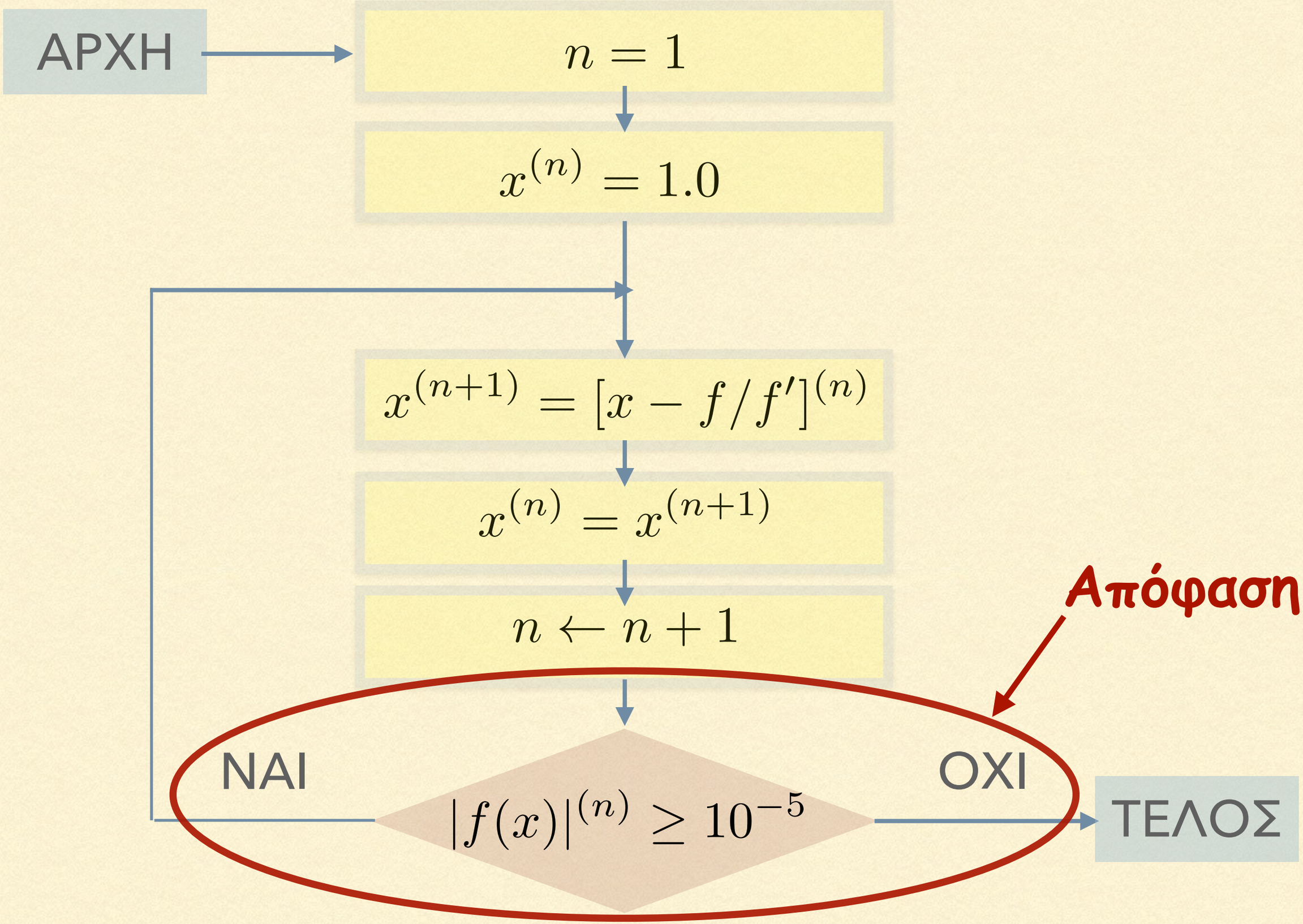


ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ

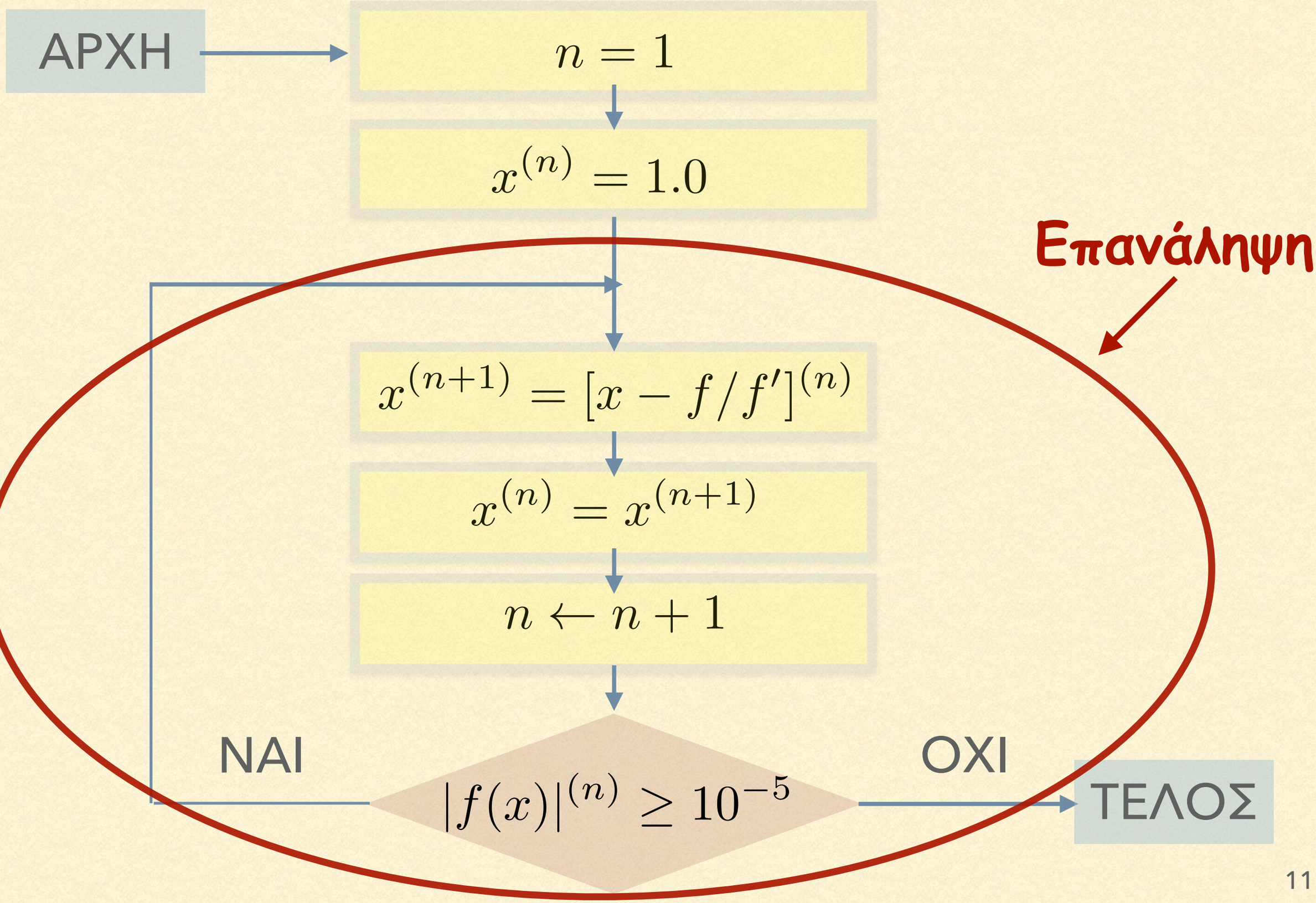


Ακολουθία

ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ



ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ



ΒΑΣΙΚΕΣ ΔΟΜΕΣ

Οι βασικές δομές του προγραμματισμού είναι:

- Η ακολουθία εντολών

  είναι μια σειρά από εντολές που εκτελούνται διαδοχικά

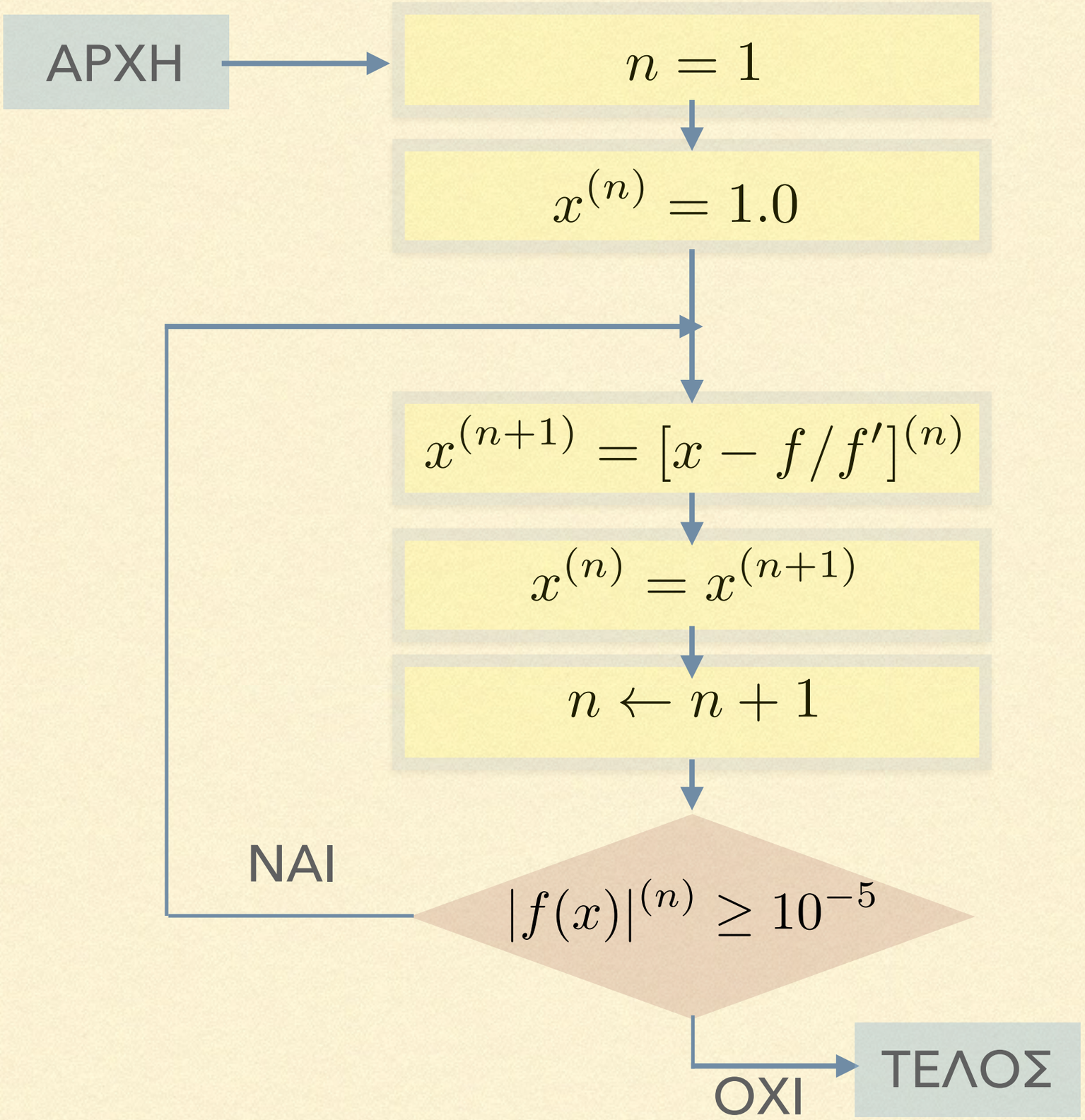
- Η απόφαση

  ελέγχεται μια συνθήκη και αναλόγως εάν είναι αληθινή ή όχι εκτελείται μια εντολή ή κάποια άλλη

- Η επανάληψη

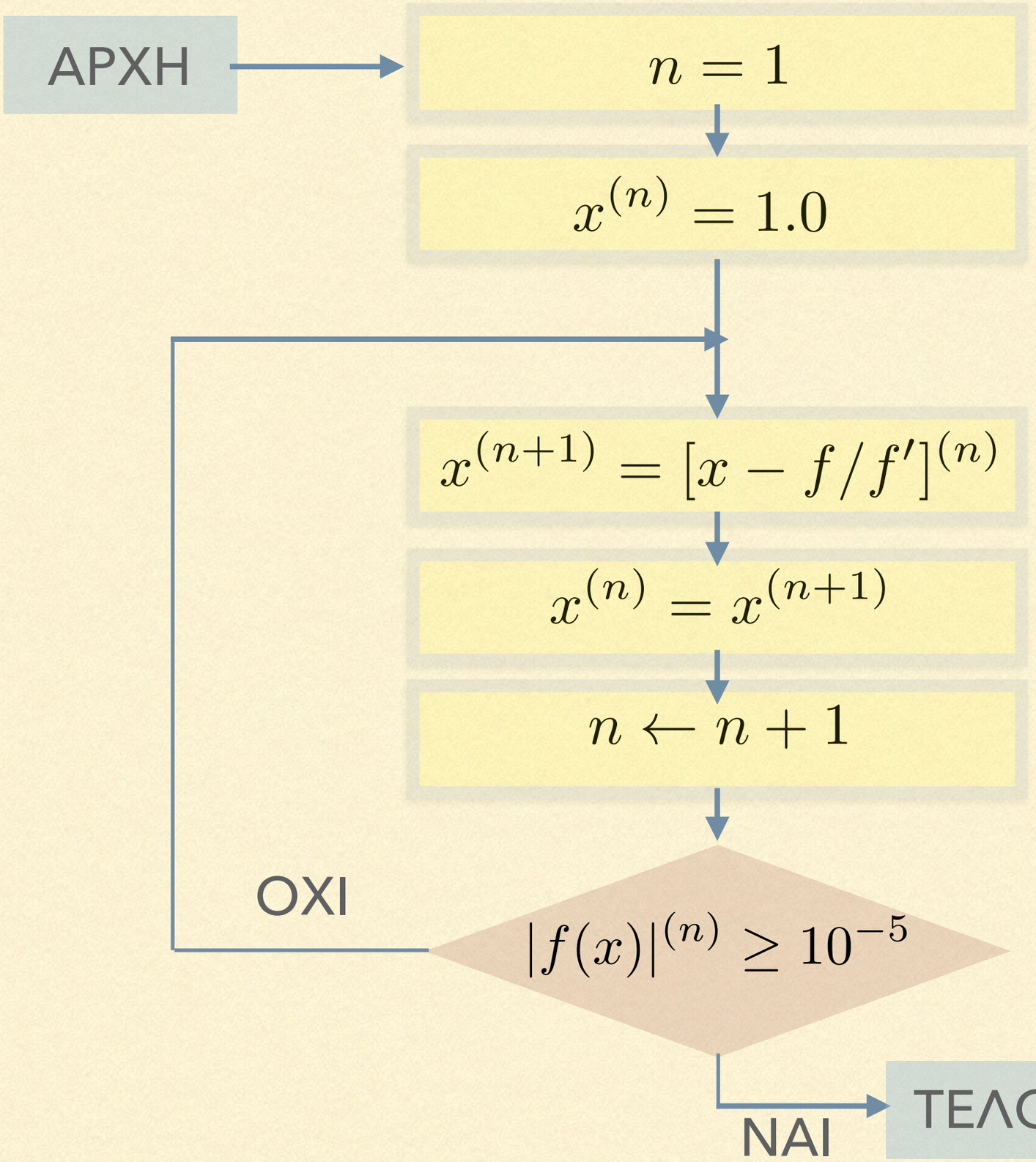
  όσο ισχύει μια συνθήκη εναναλαμβάνεται ένα μέρος του κώδικα

ΜΕΤΑΤΡΟΠΗ ΣΕ ΨΕΥΔΟΚΩΔΙΚΑ



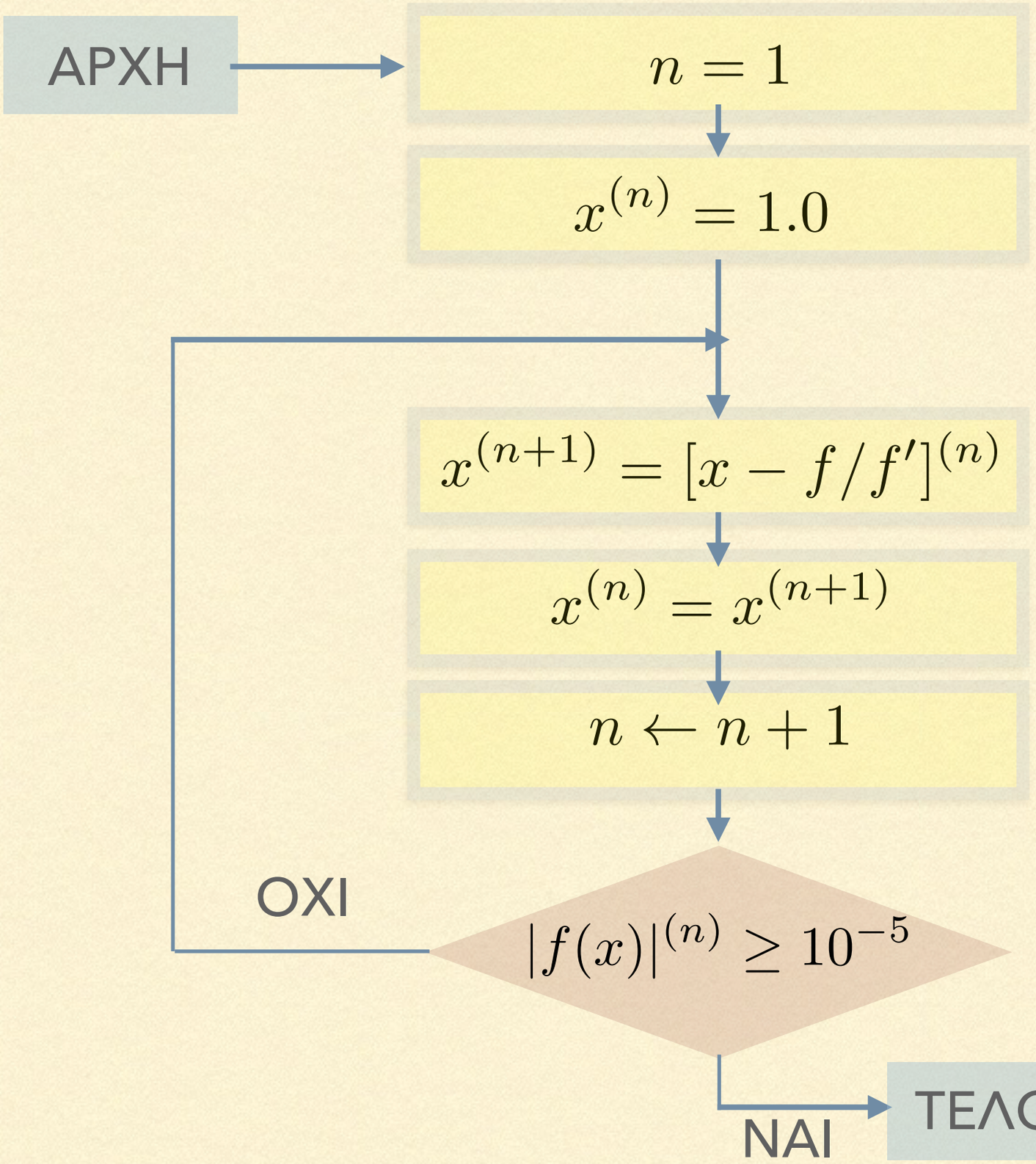

```
n = 1

xold = 1.0

xnew = xold-f/df

xold = xnew

n = n+1

if(f>=1e-5)
   repeat
else
      end
```

ΜΕΤΑΤΡΟΠΗ ΣΕ ΨΕΥΔΟΚΩΔΙΚΑ




```
n = 1
xold = 1.0

while (|f|>=1e-5)
   f = xold^2-2.0
   df = 2*xold
   xnew = xold-f/df
   xold = xnew
   n = n+1
else
   exit
```

ΜΕΤΑΤΡΟΠΗ ΣΕ ΨΕΥΔΟΚΩΔΙΚΑ




```
n = 1
xold = 1.0
f = 1.0

while (|f|>=1e-5)
   f = xold^2-2.0
   df = 2*xold
   xnew = xold-f/df
   xold = xnew
   n = n+1
else
   exit
```

απαραίτητη αρχικοποίηση μεταβλητής

ΜΕΤΑΓΡΑΦΗ ΣΤΗ ΓΛΩΣΣΑ C

```
n = 1
xold = 1.0
f = 1.0

while (|f|>=1e-5)

   f = xold^2-2.0
   df = 2*xold
   xnew = xold-f/df
   xold = xnew
   n = n+1

else
   exit
```

```
n = 1;
xold = 1.0;
f = 1.0;

while( fabs(f) >= 1e-5)
{
        f = xold*xold-2.0;
        df = 2.0*xold;
        xnew = xold-f/df;
        xold = xnew;
        n = n+1;
}

return 0;
```

άγκιστρα

ΜΕΤΑΓΡΑΦΗ ΣΤΗ ΓΛΩΣΣΑ C

```
n = 1
xold = 1.0
f = 1.0

while (|f|>=1e-5)

   f = xold^2-2.0
   df = 2*xold
   xnew = xold-f/df
   xold = xnew
   n = n+1

else
   exit
```

```
n = 1;
xold = 1.0;
f = 1.0;

while( fabs(f) >= 1e-5)
{
        f = xold*xold-2.0;
        df = 2.0*xold;
        xnew = xold-f/df;
        xold = xnew;
        n = n+1;
}

return 0;
```

συνάρτηση απόλυτης τιμής

ΜΕΤΑΓΡΑΦΗ ΣΤΗ ΓΛΩΣΣΑ C

```
n = 1
xold = 1.0
f = 1.0

while (|f|>=1e-5)

   f = xold^2-2.0
   df = 2*xold
   xnew = xold-f/df
   xold = xnew
   n = n+1

else
   exit
```

```
n = 1;
xold = 1.0;
f = 1.0;

while( fabs(f) >= 1e-5)
{
        f = xold*xold-2.0;
        df = 2.0*xold;
        xnew = xold-f/df;
        xold = xnew;
        n = n+1;
}

return 0;
```

τερματισμός προγράμματος

ΜΕΤΑΓΡΑΦΗ ΣΤΗ ΓΛΩΣΣΑ C

δήλωση τύπου μεταβλητών

```
n = 1
xold = 1.0
f = 1.0

while (|f|>=1e-5)

   f = xold^2-2.0
   df = 2*xold
   xnew = xold-f/df
   xold = xnew
   n = n+1

else
   exit
```

```c
int n;
double xold, xnew, f, df;

n = 1;
xold = 1.0;
f = 1.0;

while( fabs(f) >= 1e-5)
{
        f = xold*xold-2.0;
        df = 2.0*xold;
        xnew = xold-f/df;
        xold = xnew;
        n = n+1;
}

return 0;
```

ΠΡΟΣΘΗΚΗ ΕΝΤΟΛΗΣ ΕΚΤΥΠΩΣΗΣ

```
int n;
double xold, xnew, f, df;

n=1;
xold=1.0;
f=1.0e-5;
while(fabs(f) >= 1.0e-5)
{
        f=xold*xold-2.0;
        df=2.0*xold;
        xnew=xold-f/df;
        f=xnew*xnew-2.0;
        xold=xnew;
        n=n+1;
}
printf("%6.5e\n",xnew);
return 0;
```

εντολή εκτύπωσης αποτελέσματος

ΠΡΟΣΘΗΚΗ HEADERS

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

int main()
{ int n;
  double xold, xnew, f, df;

  n=1;
  xold=1.0;
  f=1.0e-5;
  while(fabs(f) >= 1.0e-5)
  {
        f=xold*xold-2.0;
        df=2.0*xold;
        xnew=xold-f/df;
        f=xnew*xnew-2.0;
        xold=xnew;
        n=n+1;
  }
  printf("%6.5e\n",xnew);
  return 0;
}
```

Απαραίτητη δήλωση συναρτήσεων της C

ΠΡΟΣΘΗΚΗ ΚΥΡΙΑΣ ΣΥΝΑΡΤΗΣΗΣ

Απαραίτητη δήλωση κύριας συνάρτησης

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

int main()
{ int n;
  double xold, xnew, f, df;

  n=1;
  xold=1.0;
  f=1.0e-5;
  while(fabs(f) >= 1.0e-5)
  {
        f=xold*xold-2.0;
        df=2.0*xold;
        xnew=xold-f/df;
        f=xnew*xnew-2.0;
        xold=xnew;
        n=n+1;
  }
  printf("%6.5e\n",xnew);
  return 0;
}
```

ΟΛΟΚΛΗΡΩΜΕΝΟ ΠΡΟΓΡΑΜΜΑ C

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

int main()
{ int n;
  double xold, xnew, f, df;

  n=1;
  xold=1.0;
  f=1.0e-5;

  while(fabs(f) >= 1.0e-5)
  {
        f=xold*xold-2.0;
        df=2.0*xold;
        xnew=xold-f/df;
        f=xnew*xnew-2.0;
        xold=xnew;
        n=n+1;
  }
  printf("%6.5e\n",xnew);
  return 0;
}
```

ΜΕΤΑΓΛΩΤΤΙΣΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Ένας κώδικας για να "τρέξει" και να δώσει το αποτέλεσμα των πράξεων απαιτείται* πρώτα να μεταγλωττιστεί σε γλώσσα μηχανής (machine language).

Η μεταγώττιση γίνεται μέσω ειδικών προγραμμάτων μεταγλώττισης (compilers).

Για κάθε γλώσσα προγραμματισμού υπάρχουν διάφοροι compilers και για διάφορα λειτουργικά συστήματα.

*υπάρχουν και γλώσσες προγραμματισμού που είναι interpreted (μεταγλώττιση/εκτέλεση μία εντολή τη φορά)

http://rextester.com/runcode

run code

Language: C (gcc) Editor: EditArea

10 pt

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

int main()
{ int n;
  double xold, xnew, f, df;

  n=1;
  xold=1.0;
  f=1.0e-5;

  while(fabs(f) >= 1.0e-5)
  {
        f=xold*xold-2.0;
        df=2.0*xold;
        xnew=xold-f/df;
        f=xnew*xnew-2.0;
        xold=xnew;
        n=n+1;
  }
  printf("%7.6e\n",xnew);
  return 0;
}
```

Position: Ln 1, Ch 1 Total: Ln 23, Ch 300

Run it Save it Show compiler warnings [+] Compiler args [+] Show input

Compilation time: 0.12 sec, absolute running time: 0.14 sec, cpu time: 0 sec, memory peak: 3 Mb, absolute service time: 0.27 sec

1.414216e+00

http://www.tutorialspoint.com/compile_c_online.php

codingground SIMPLY EASY CODING (GNU GCC v4.8.3) </> System File Project Edit View Tutorials Help

New Project

root
main.c

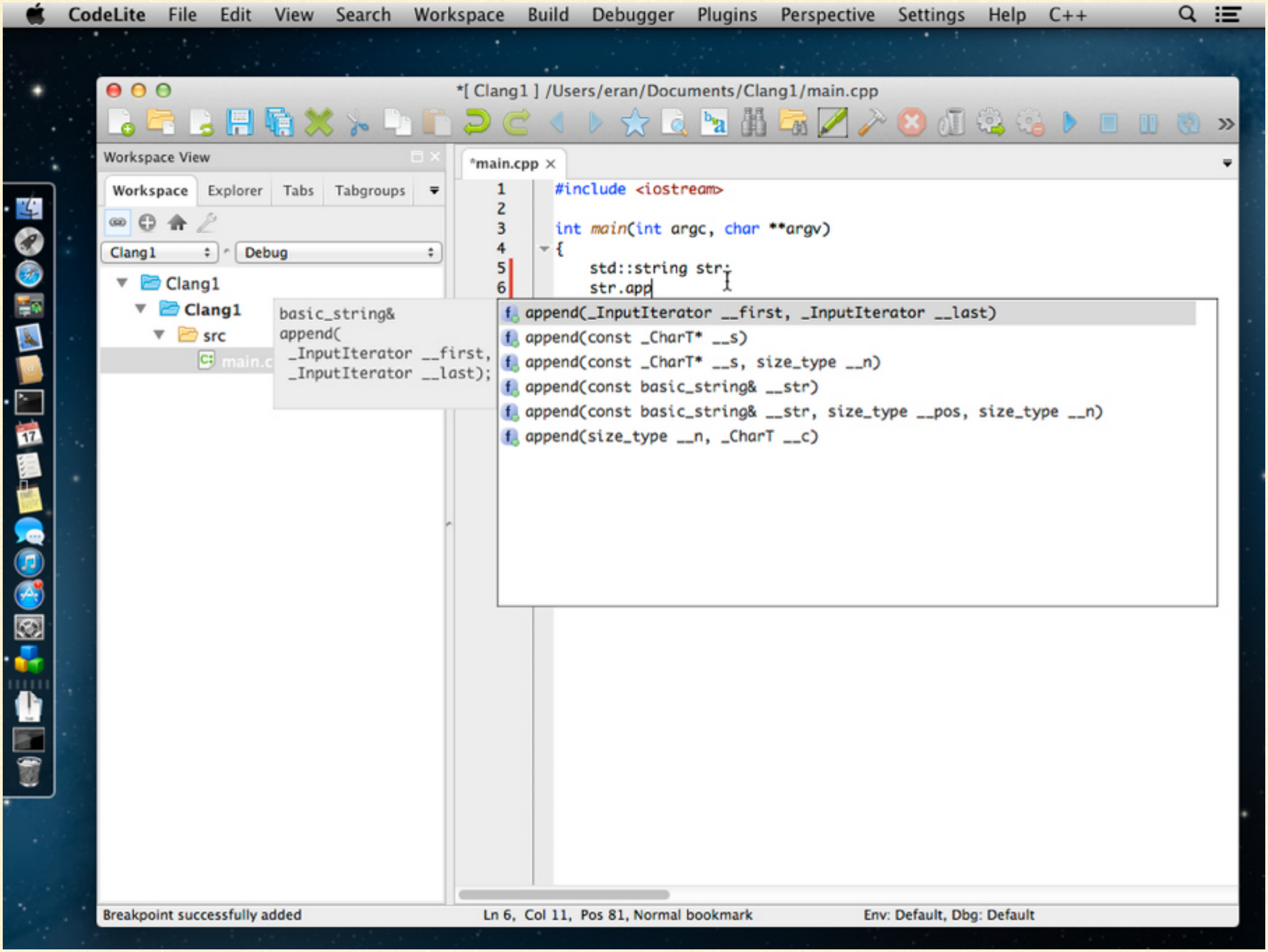
Compile | Execute main.c ×

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

int main()
{ int n;
  double xold, xnew, f, df;

  n=1;
  xold=1.0;
  f=1.0e-5;

  while(fabs(f) >= 1.0e-5)
  {
        f=xold*xold-2.0;
        df=2.0*xold;
        xnew=xold-f/df;
        f=xnew*xnew-2.0;
        xold=xnew;
        n=n+1;
  }
  printf("%7.6e\n",xnew);
  return 0;
}
```

Terminal

```
sh-4.3# gcc -o main *.c
sh-4.3# main
1.414216e+00
sh-4.3#
```

http://codelite.org/



http://en.wikipedia.org/wiki/List_of_compilers#C_compilers

Υπάρχουν πάνω από 50 compilers για τη γλώσσα C.

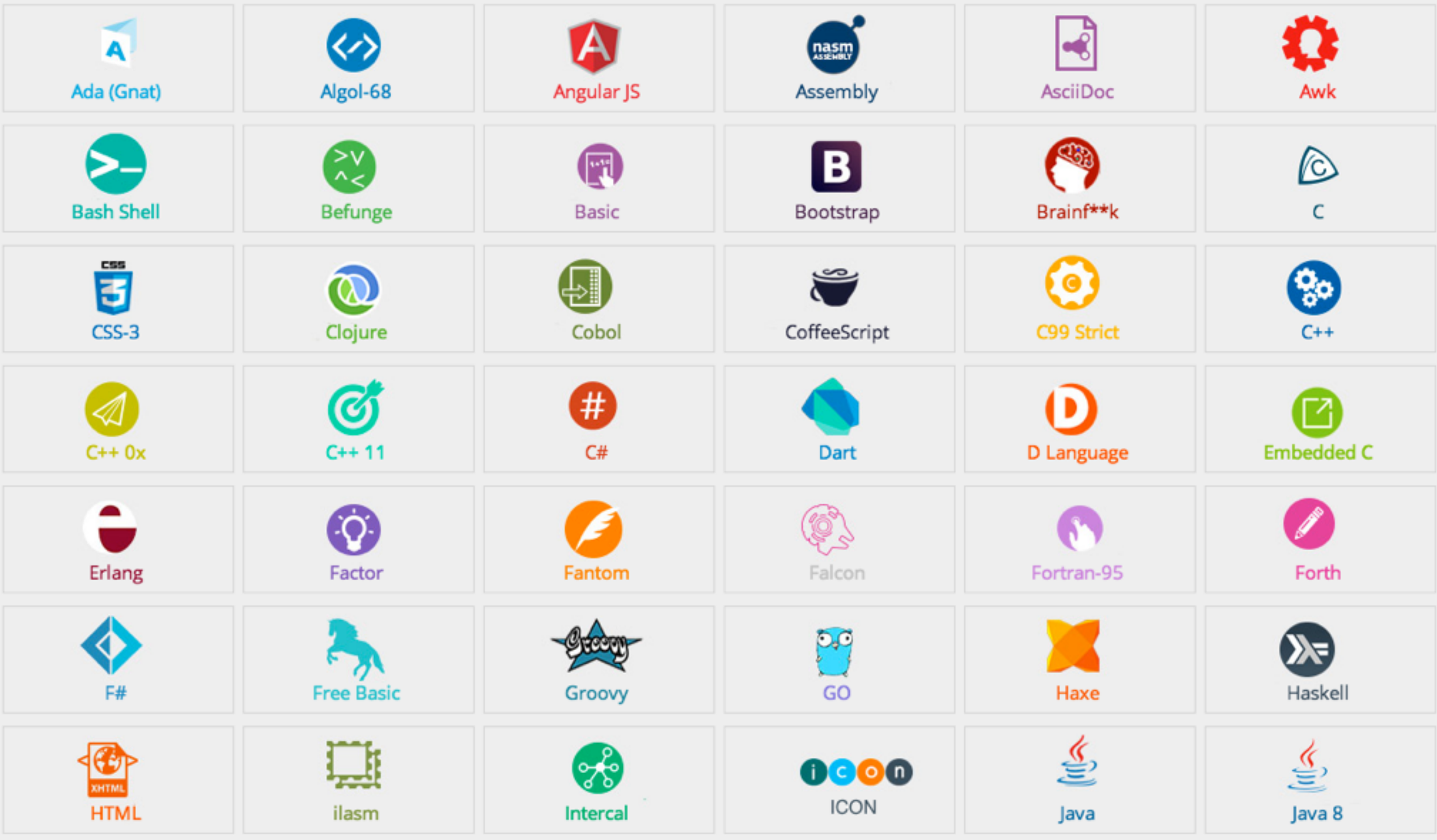
C compilers [edit]

This list is incomplete; you can help by expanding it.

Compiler	Author	Microsoft Windows	Unix-like	Other OSs	License type
AMPC	Axiomatic Solutions Sdn Bhd	Yes	Yes	Yes	Proprietary
Aztec C	Manx Software Systems	No	No	CP/M, CP/M-86, MS-DOS, Classic Mac OS	Proprietary
Amsterdam Compiler Kit	Andrew Tanenbaum and Ceriel Jacobs	No	Yes	Yes	BSD
CCS C Compiler	CCS, Inc.	Yes	Yes	Yes	Proprietary
Ch	SoftIntegration, Inc	Yes	Mac OS X, FreeBSD, Linux, Solaris, HP-UX, AIX, Qnx	Yes	Freeware
Clang	LLVM Project	Yes	Yes	Yes	BSD
CodeWarrior	Metrowerks	Yes	Yes	Classic Mac OS	Proprietary
CParser/libFirm	Matthias Braun, Christoph Mallon and Michael Beck	Yes	Yes	Yes	GPL
DeSmet-C	C-Ware Corporation	No	No	MS-DOS	GPL
Digital Mars	Digital Mars	Yes	No	No	Proprietary
Dignus Systems/C	Dignus, L.L.C	Yes (host)	Yes (host)	Z/Architecture	Proprietary
GCC C	GNU Project	MinGW, Cygwin	Yes	IBM mainframe, AmigaOS, VAX/VMS, RTEMS	GPL
Hippo-C	Hippopotamus Software, Haba Systems	No	No	Classic Mac OS, Atari ST	Proprietary
IAR C/C++ Compilers	IAR Systems	Yes	No	No	Proprietary
Interactive C	KISS Institute for Practical Robotics	Yes	Unix, Mac OS X, Linux, IRIX, Solaris, SunOS	No	Freeware
LabWindows/CVI	National Instruments	Yes	Yes	Yes	Proprietary
Lattice C	Lifeboat Associates	No	Yes	MS-DOS, OS/2, Commodore, Amiga, Atari ST, Sinclair QL	Proprietary
lcc	Chris Fraser and David Hanson	Yes	Yes	Yes	Freeware (source code available for non-commercial use)
Mac C	Consulair	No	No	Classic Mac OS	Proprietary
Mark Williams C	Mark Williams Company	Yes	Coherent	Yes	Proprietary
Micro-C Compiler (mcc)	Dunfield Development Services	No	No	MS-DOS	Freeware (source code available)
Micro C Compiler (moc)	Roshan Singh	Yes	Yes	Yes	Freeware (source code available for non-commercial use)
MikroC Compiler	Mikroelektronika	Yes	Yes	Yes	Proprietary
MPW C	Apple	No	No	Classic Mac OS	Proprietary
Neatcc	Ali Gholami Rudi	No	Yes	No	BSD
Nwcc	Nils Weller	No	Yes	No	BSD
Open64	AMD SGI Google HP Intel Nvidia PathScale Tsinghua University and others	No	Yes	Yes	GPL
Open Watcom	Sybase and SciTech Software	Yes	Linux	OS/2, MS-DOS	Sybase Open Watcom Public License
Pelles C	Pelle Orinius	Yes	No	No	Freeware
PGCC	The Portland Group	Yes	Yes	Unknown	Proprietary
Portable C Compiler	Stephen C. Johnson, Anders Magnusson and others	Yes	Yes	Yes	BSD
Power C	Mix Software	No	No	Yes	Proprietary
QuickC	Microsoft	Yes	No	No	Proprietary
RCC (RCOR C Compiler)	Rodrigo Caetano Rocha (rcor)	Yes	Yes	No	GPL
Ritchie C Compiler (PDP-11)	Dennis Ritchie and John Reiser; converted to cross-compiler by Doug Gwyn	Yes	Yes	Yes	Freeware
SAS/C	SAS Institute	Yes	Yes	Yes IBM mainframe, AmigaOS, 68K, 88K	Proprietary
SCORE C (tcc)	ODC-I	Yes	Yes	Yes	Proprietary
Small-C	Ron Caine, James E. Hendrix, Byte magazine	Yes	Yes	CP/M, MS-DOS	Public Domain
Smaller C	Alexey Frunze	Yes	Linux, RetroBSD	MS-DOS	BSD
Small Device C Compiler	Sandeep Dutta and others	Yes	Yes	Unknown	GPL
SubC	Nils M Holm	MinGW	FreeBSD, NetBSD, Linux	MS-DOS	Public Domain
THINK C, Lightspeed C	THINK Technologies	No	No	Classic Mac OS	Proprietary
Tiny C Compiler	Fabrice Bellard	Yes	Yes	No	LGPL
(Borland) Turbo C	Embarcadero	Yes	No	Yes	Proprietary - V 2.01 freely available
ups debugger (includes C interpreter)	Tom Hughes, Ian Edwards, and others	No	Yes	Solaris, SunOS	GPL
VBCC	Volker Barthelmann	Yes	Yes	Yes	Freeware (source code available, modification not allowed)
Virtual-C IDE	Dieter Pawelczak	Yes	Mac OS X	No	Freeware (for non-commercial use)
Visual C++ Express	Microsoft	Yes	No	No	Freeware
Wind River (Diab) Compiler	Wind River Systems	Yes	Yes	Yes	Proprietary
XL C	IBM	No	AIX, Linux	No	Proprietary
MCP Compiler	Unisys	No	No	MCP	Proprietary

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Οι 100 δημοφιλέστερες γλώσσες προγραμματισμού:



ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Java MySQL
Java MySQL
JS
Jquery
julia
Ksh Shell
Lisp
LOLCODE
Lua
Matlab/Octave
Malgolbe
Markdown
MathML
Mozart-OZ
Nimrod
Node.JS
Objective-C
OCaml
Pascal
PARI/GP
Pawn
Perl
Perl MySQL
PHP
PHP MySQL
Web View
Prolog
Python
Python-3
Python MySQL
Rexx
reStructure
Ruby
Rust
R Programming
Scala
Scheme
Smalltalk
Simula
SML/NJ
Script Basic
SQLite SQL
Tcl
Unlambda
VB.NET
Verilog
Whitespace

ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

OBJECTIVE C

Objective C is an object oriented programming language, based on C, used by Apple developers .

Objective C was first developed in the 1990s.

Objective C files have an .m extension.

Objective C can:

Develop mobile apps for iOS

iOS

Develop applications for OS X

Objective C Jobs

Average Salary: $70,000

Job Count: 18,849

Top Cities: New York, San Francisco, Chicago

Top Employers: Regis Corporation, SmartStyle, SmartCuts

Tidbits of Wisdom

Though considered the correct language for iOS development, it cannot be used for other platforms such as Android.

C++

C++ is an object-oriented programming language used to develop software, video games, and more.

C++ was first developed in 1983, as an enhancement to the C programming language

C++ files have a .c++ extension.

C++ can develop:

Apps for Windows and Linux

Video games

Mobile apps

C++ Jobs

Average Salary: $90,000

Job Count: 31,893

Top Cities: New York, Seattle, San Francisco

Top Employers: Amazon, CyberCoders, Microsoft

JAVA®

This is a server-side interpreted compiled language, using a virtual machine. It is not JavaScript, and is not related to it.

Java was developed in 1995, and is one of the oldest programming languages on the web.

Java lets you:

Play online games

Upload photos

Take virtual tours

Use interactive maps

Java Jobs

Average Salary: $95,000

Job Count: 66,485

Top Cities: New York, Washington D.C., San Jose

Top Employers: Amazon, IBM, eBay

Tidbits of Wisdom

Users can disable Java on their machines.

Java is the basis of Android

Slow to change, so it's easier to keep up with

PYTHON™

This is a server-side interpreted, open-source, non-compiled, scripting language. It can be used on its own, or as part of another framework, like django.

Python can:

Build websites

Provide database access

python™

Build desktop graphic user interfaces (GUIs)

Build software and games

Python Jobs

Average Salary: $83,000

Job Count: 19,627

Top Cities: Mountain View, San Francisco, New York

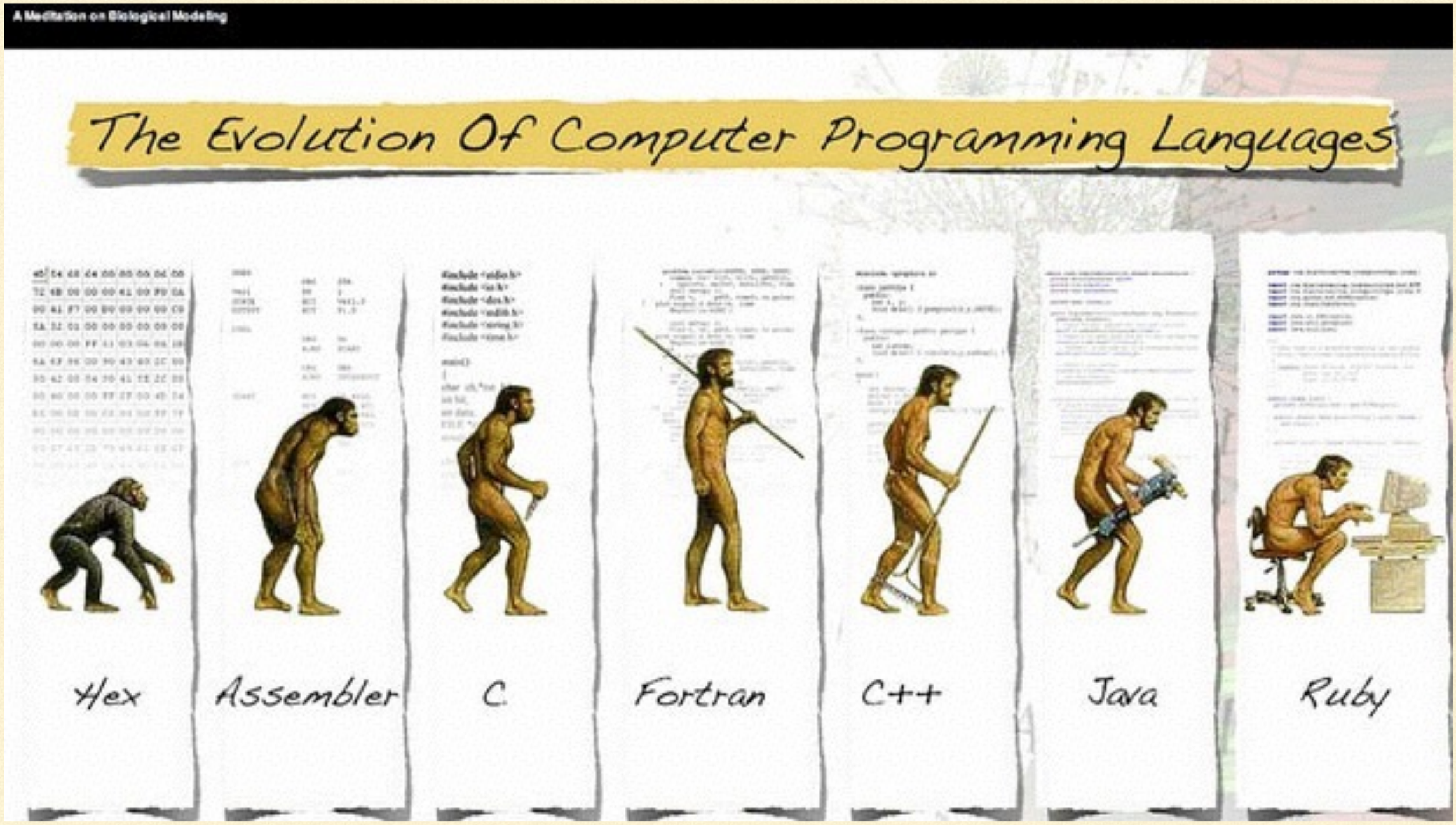
Top Employers: Amazon, Intel®, Dell

Tidbits of Wisdom

NASA's shuttle support contractor, United Space Alliance (USA) uses Python.

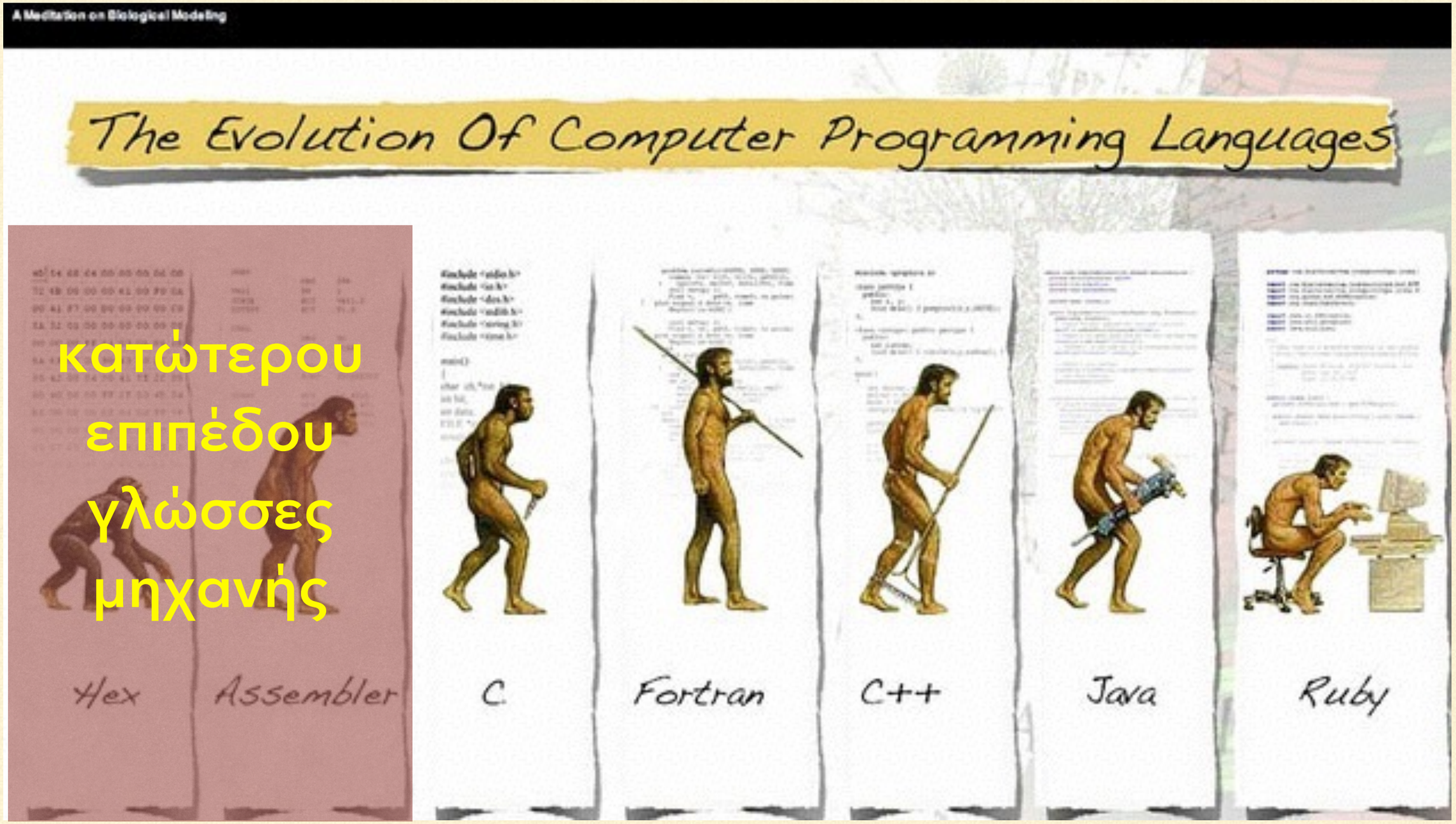
ΕΞΕΛΙΞΗ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η γλώσσα C είναι μια γλώσσα ανώτερου επιπέδου (high-level language) χωρίς την πολυπλοκότητα ή εξειδίκευση μεταγενέστερων γλωσσών.



ΕΞΕΛΙΞΗ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η γλώσσα C είναι μια γλώσσα ανωτέρου επιπέδου (high-level language) χωρίς την πολυπλοκότητα ή εξειδίκευση μεταγενέστερων γλωσσών.



ΕΞΕΛΙΞΗ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

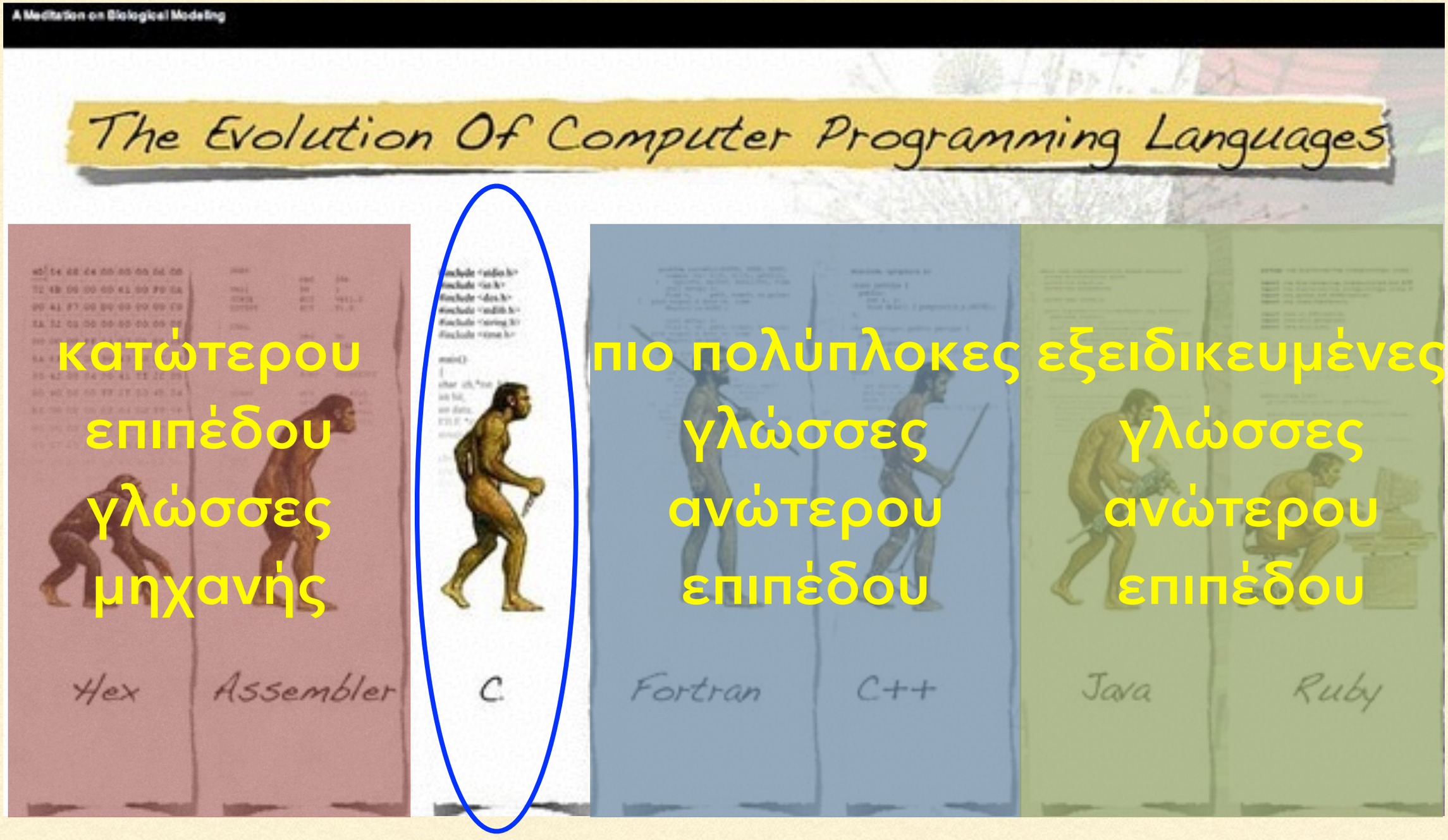
Η γλώσσα C είναι μια γλώσσα ανώτερου επιπέδου (high-level language) χωρίς την πολυπλοκότητα ή εξειδίκευση μεταγενέστερων γλωσσών.

ΕΞΕΛΙΞΗ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η γλώσσα C είναι μια γλώσσα ανώτερου επιπέδου (high-level language) χωρίς την πολυπλοκότητα ή εξειδίκευση μεταγενέστερων γλωσσών.

ΕΞΕΛΙΞΗ ΤΩΝ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Η γλώσσα C είναι μια γλώσσα ανώτερου επιπέδου (high-level language) χωρίς την πολυπλοκότητα ή εξειδίκευση μεταγενέστερων γλωσσών.



ΓΙΑΤΙ C ΩΣ ΠΡΩΤΗ ΓΛΩΣΣΑ;

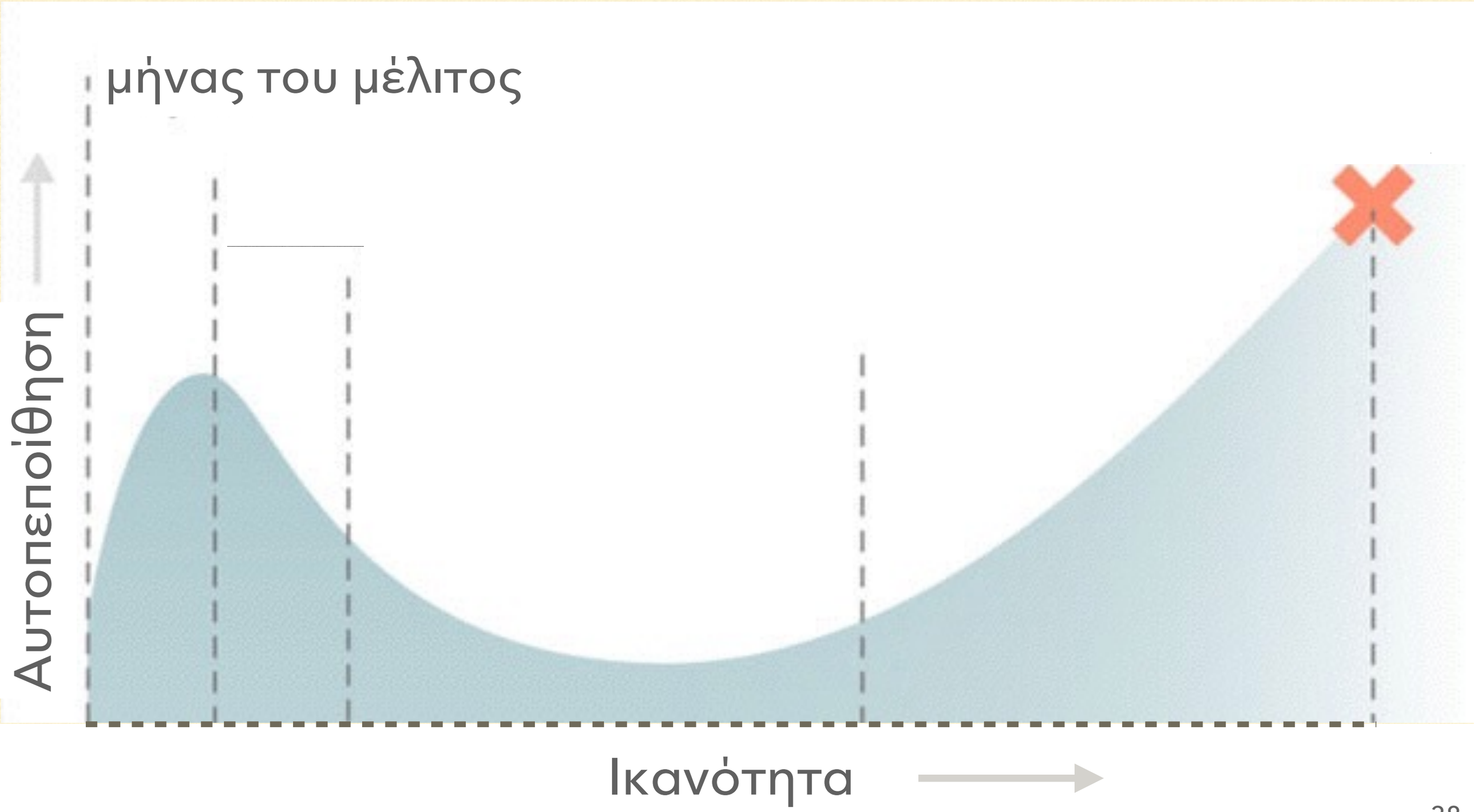
Η γλώσσα C είναι για τις άλλες γλώσσες ανώτερου επιπέδου ό,τι και τα λατινικά για τις νεότερες λατινογενείς γλώσσες (Ιταλικά, Γαλλικά, Ισπανικά).

Το να ξεκινήσει κανείς να μαθαίνει προγραμματισμό με μια πιο εξειδικευμένη γλώσσα, όπως π.χ. με την Python, είναι σα να μαθαίνει να οδηγεί με αυτόματο αυτοκίνητο... το πιο πιθανό είναι να μη μάθει ποτέ αργότερα να οδηγεί με ταχύτητες...

Για τους παραπάνω λόγους, η γλώσσα C ενδείκνυται ως μια πρώτη γλώσσα προγραμματισμού, παρόλη την αυστηρότητά της.

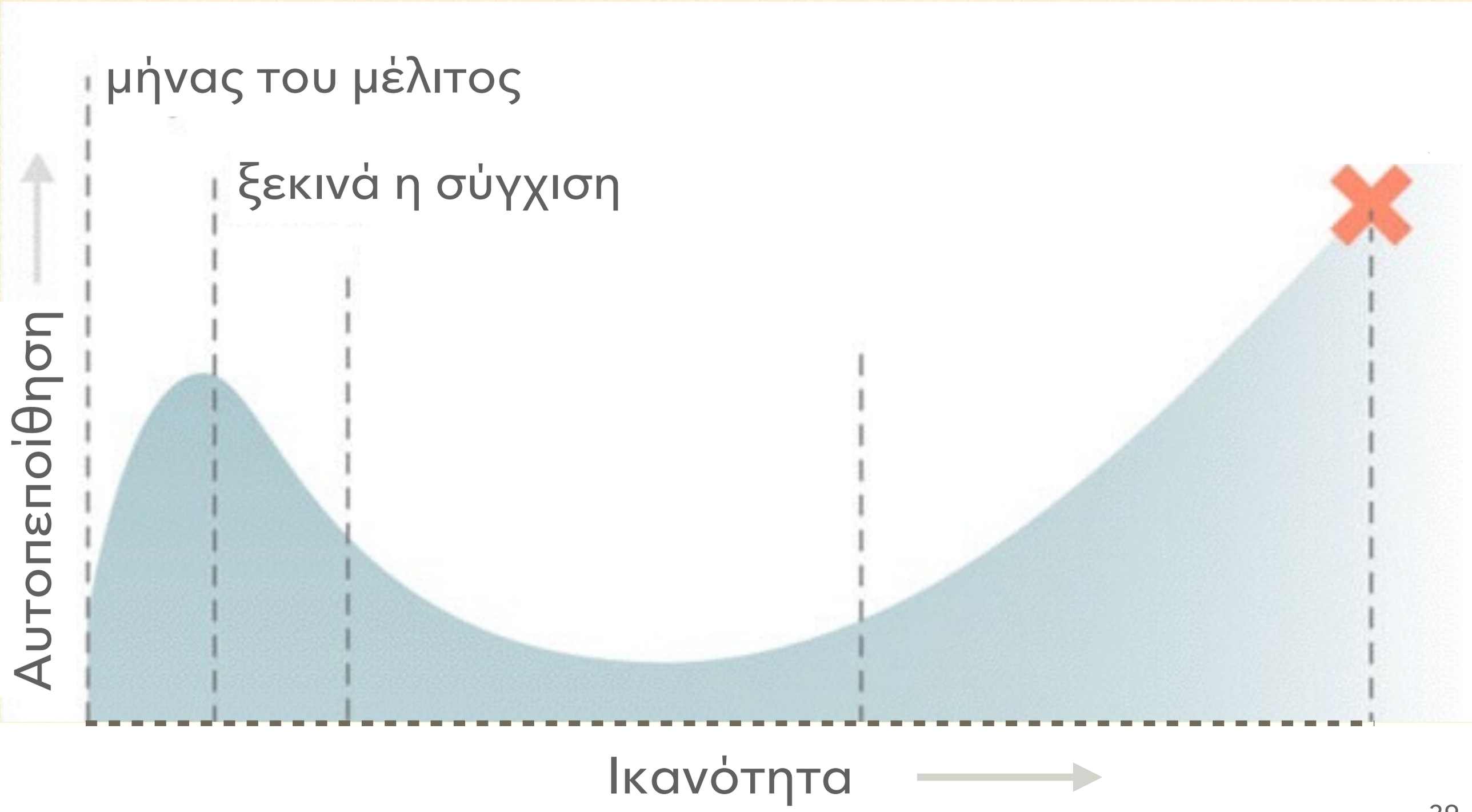
Η ΕΞΕΛΙΞΗ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ

Η εξελικτική πορεία ενός προγραμματιστή είναι μη-γραμμική:



Η ΕΞΕΛΙΞΗ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ

Η εξελικτική πορεία ενός προγραμματιστή είναι μη-γραμμική:

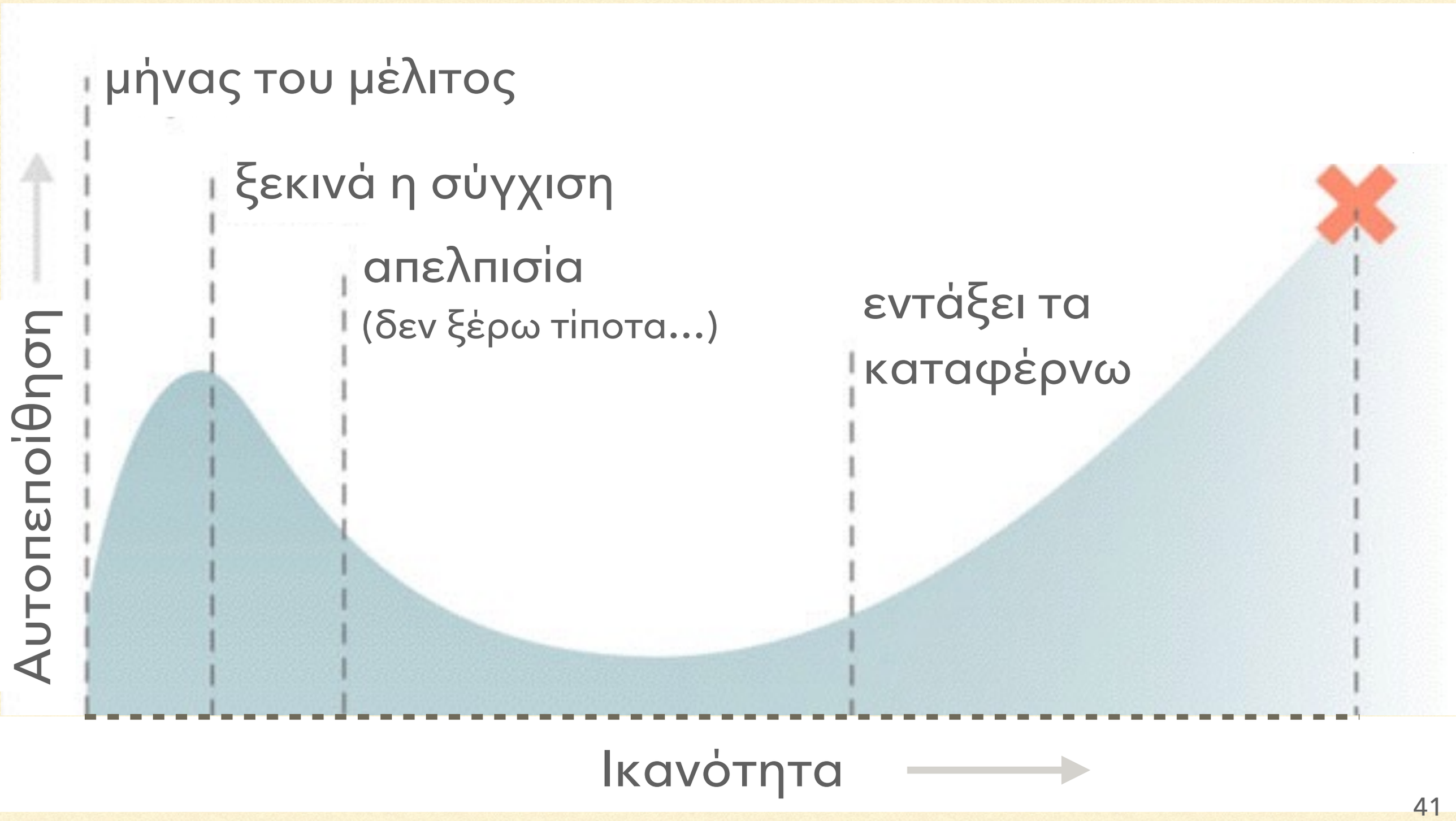


Η ΕΞΕΛΙΞΗ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ

Η εξελικτική πορεία ενός προγραμματιστή είναι μη-γραμμική:

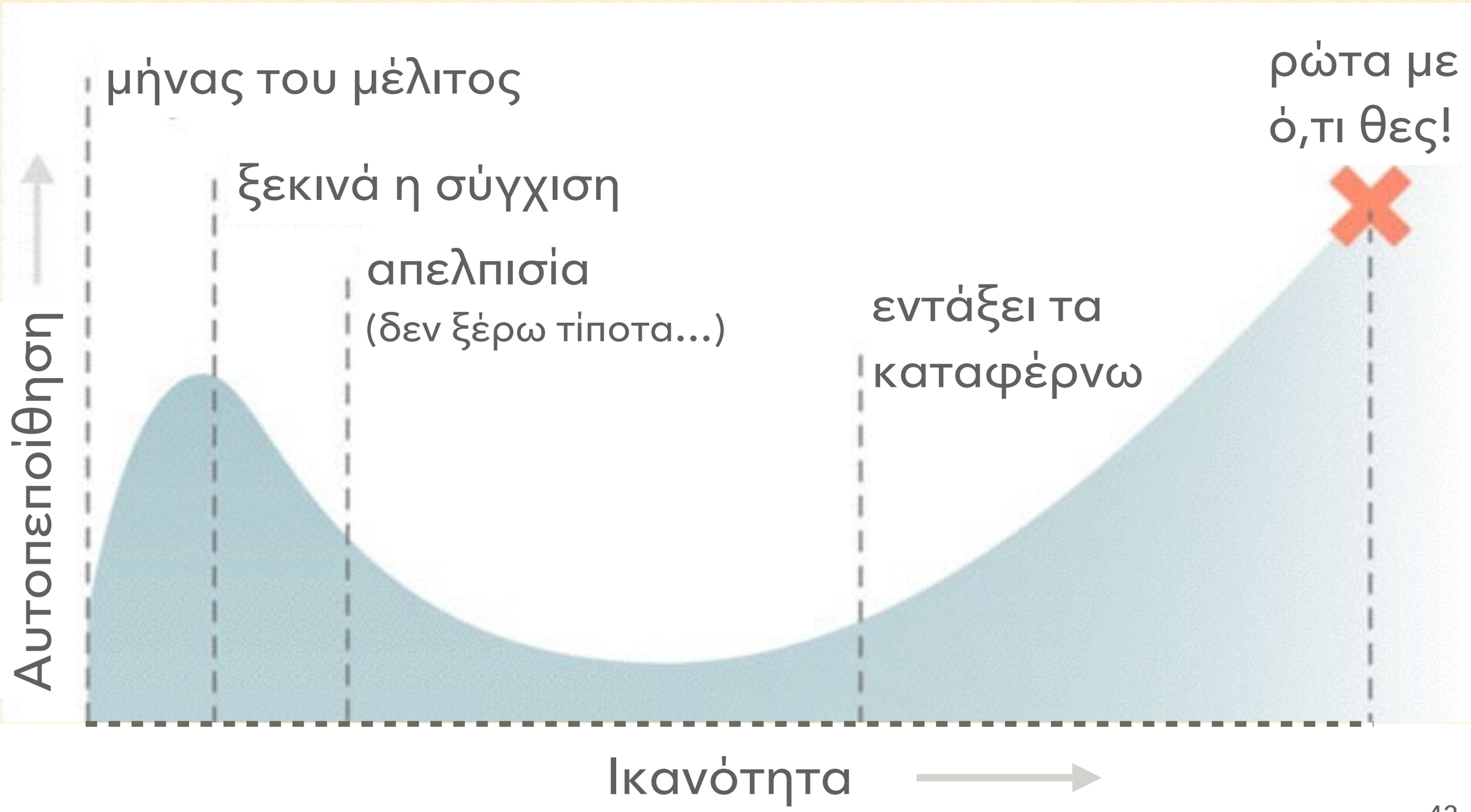
Η ΕΞΕΛΙΞΗ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ

Η εξελικτική πορεία ενός προγραμματιστή είναι μη-γραμμική:



Η ΕΞΕΛΙΞΗ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ

Η εξελικτική πορεία ενός προγραμματιστή είναι μη-γραμμική:



Η ΓΛΩΣΣΑ C - ΣΥΝΤΟΜΗ ΑΝΑΔΡΟΜΗ

Δημιουργήθηκε από τον Dennis Richie στις αρχές της δεκαετίας του 1970 στα εργαστήρια Bell Labs της εταιρείας AT&T.

Ονομάστηκε C διότι κληρονόμησε πολλά χαρακτηριστικά από μια προηγούμενη γλώσσα που ονομαζόταν "B".

Ο κύριος σκοπός της δημιουργίας της γλώσσας C ήταν η χρήση της για την ανάπτυξη του λειτουργικού συστήματος UNIX (το οποίο μετεξελίχθηκε στο Linux για υπολογιστές και στο Android για κινητά/τάμπλετ).

ANSI C

Το 1989 ο οργανισμός ANSI (American National Standard Institute) εισήγαγε το πρότυπο της ANSI C, το οποίο να καθορίζει αυστηρά μια σειρά από κανόνες, και χαρακτηριστικά της γλώσσας. Υποστηρίζεται από όλους τους μεταγλωττιστές C, άρα έχει μεγάλη φορητότητα.

Στο τέλος της δεκαετίας του ’90 δημιουργήθηκε το νεότερο πρότυπο C99. Οι περισσότεροι μεταγλωττιστές το υποστηρίζουν εν μέρει, λίγοι όμως το υποστηρίζουν εξ’ ολοκλήρου, γι’ αυτό έχει μικρή φορητότητα.

- Το πρότυπο ANSI C παραμένει το επικρατέστερο.

ΚΥΡΙΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ C

- Απλό συντακτικό και λίγες δεσμευμένες λέξεις
- Φορητότητα κώδικα (εκτελείται σε διαφορετικά λειτουργικά συστήματα)
- Υποστηρίζει υπορουτίνες και συναρτήσεις (δομημένος προγραμματισμός)
- Παρέχει έτοιμες συναρτήσεις

ΚΥΡΙΑ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΗΣ C

- Έλλειψη αυστηρών περιορισμών (ο μεταγλωττιστής δεν θα αντιληφθεί όλα τα σφάλματα)

- Ένα πρόγραμμα C μπορεί να γραφεί με αρκετά πολύπλοκο τρόπο ακόμα και αν αποτελείται από λίγες γραμμές κώδικα

- Δεν είναι αντικειμενοστρεφής γλώσσα όπως η C++.

ΚΥΚΛΟΣ ΔΗΜΙΟΥΡΓΙΑΣ ΕΝΟΣ ΚΩΔΙΚΑ C

- Συγγραφή κώδικα (edit)
- Μεταγλώττιση (compile)
- Σύνδεση με βιβλιοθήκες (Link)*
- Εκτέλεση (run/execute)

*Η σύνδεση με διάφορες βιβλιοθήκες γίνεται συνήθως αυτόματα, σε προχωρημένους κώδικες μπορεί να χρειασθεί να καθορίσουμε επιπλέον παραμέτρους

ΕΝΑ ΠΡΩΤΟ ΑΠΛΟ ΠΡΟΓΡΑΜΜΑ

```c
#include <stdio.h>


int main(void)
{
    printf("Hello, World!\n");

    return 0;
}
```

Αφού γράψουμε το πρόγραμμα, το σώζουμε ως ένα αρχείο με κατάληξη .c (π.χ. hello.c).

Η ΟΔΗΓΙΑ #include

```c
#include <stdio.h>


int main(void)
{
    printf("Hello, World!\n");

    return 0;
}
```

Η οδηγία #include ενσωματώνει στον κώδικα εντολές που βρίσκονται σε κάποιο άλλο αρχείο.

ΤΟ ΑΡΧΕΙΟ #<stdio.h>

```c
#include <stdio.h>


int main(void)
{
    printf("Hello, World!\n");

    return 0;
}
```

Το αρχείο stdio.h (standard input output) περιέχει τις δηλώσεις των συναρτήσεων για εμφάνιση δεδομένων στην οθόνη (output) και εισαγωγή δεδομένων από το πληκτρολόγιο (input).

Η ΟΔΗΓΙΑ #include ΚΑΙ ΤΑ HEADER FILES

- Η οδηγία #include ξεκινάει πάντα με το σύμβολο # και δεν τελειώνει με ;

- Τα αρχεία με κατάληξη .h περιέχουν δηλώσεις συναρτήσεων και ονομάζονται header files.

- Όταν το όνομα του αρχείου είναι γραμμένο μέσα σε < >, τότε ο μεταγλωττιστής ψάχνει σε προεπιλεγμένους φακέλους του συστήματος.

- Μπορούμε να ενσωματώσουμε και δικά μας αρχεία έχοντας το όνομα γραμμένο μέσα σε " ". Ο μεταγλωττιστής ψάχνει στον ίδιο φάκελο που βρίσκεται το κυρίως πρόγραμμα.

Η ΣΥΝΑΡΤΗΣΗ main()

```c
#include <stdio.h>


int main(void)
{
    printf("Hello, World!\n");

    return 0;
}
```

Η συνάρτηση main(void) υπάρχει υποχρεωτικά σε κάθε πρόγραμμα C και είναι η κύρια συνάρτηση του προγράμματος.

Η ΣΥΝΑΡΤΗΣΗ main()

```c
#include <stdio.h>


int main(void)
{
    printf("Hello, World!\n");

    return 0;
}
```

Το void (κενό) δηλώνει πως δε χρησιμοποιούμε ορίσματα στη συνάρτηση (μελλοντικά, θα χρησιμοποιήσουμε).

Η ΣΥΝΑΡΤΗΣΗ main()

- Η εκτέλεση του προγράμματος ξεκινάει από την πρώτη εντολή που υπάρχει μέσα στη συνάρτηση main().

- Η εκτέλεση του προγράμματος τελειώνει με την τελευταία εντολή που υπάρχει μέσα στη συνάρτηση main() ή και νωρίτερα με μία εντολή εξόδου.

- Όλες οι εντολές της συνάρτησης main() αλλά και κάθε άλλης συνάρτησης πρέπει να περιέχονται μέσα σε άγκιστρα {...}.

- Μπορούμε να γράψουμε main() αντί για main(void), (δε συμφωνεί με το πρότυπο ANSI, αλλά είναι αποδεκτό).

Η ΕΝΤΟΛΗ printf()

```c
#include <stdio.h>


int main(void)
{
    printf("Hello, World!\n");

    return 0;
}
```

Η εντολή printf() είναι δηλωμένη στο <stdio.h> και τυπώνει ό,τι βρίσκεται μέσα σε " " στην οθόνη.

Η ΕΝΤΟΛΗ printf()

```c
#include <stdio.h>


int main(void)
{
    printf("Hello, World!\n");

    return 0;
}
```

Ο χαρακτήρας \n δημιουργεί στο τέλος μια νέα γραμμή (new line), όπως το πλήκτρο Enter όταν πληκτρολογούμε.

Ο ΤΥΠΟΣ ΤΗΣ MAIN

```c
#include <stdio.h>


int main(void)
{
    printf("Hello, World!\n");

    return 0;
}
```

Στο συγκεκριμένο παράδειγμα, δηλώνουμε πως ο τύπος της main() είναι int, δηλαδή στο τέλος πρέπει να λάβει ως τιμή κάποιον ακέραιο αριθμό.

Η ΕΝΤΟΛΗ return

```c
#include <stdio.h>


int main(void)
{
    printf("Hello, World!\n");

    return 0;
}
```

Η εντολή return δίνει τιμή στον ακέραιο αριθμό της main() και τερματίζεται το πρόγραμμα. Συμβατικά, χρησιμοποιούμε το 0 για να δηλώσουμε τον ομαλό τερματισμό του προγράμματος.

ΣΧΟΛΙΑ

```c
#include <stdio.h>


int main(void)
{
    /* Εκτύπωσε μια φράση */

    printf("Hello, World!\n”);

    /* Τερματισμός προγράμματος */

    return 0;
}
```

Τα σχόλια περιλαμβάνονται μέσα σε `/* */` .

(κάποιοι compilers αναγνωρίζουν και το `//` στην αρχή μόνο κάθε γραμμής)