ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

ΠΛΗΡΟΦΟΡΙΚΗ Ι

Ενότητα 4: Πίνακες (arrays)

Μιχάλης Δρακόπουλος

Σχολή Θετικών επιστημών

Τμήμα Μαθηματικών

Δομές δεδομένων

Σε πολλά (κυρίως μαθηματικά) προβλήματα (π.χ. ανάλυση πειραμάτων, στατιστική επεξεργασία, γραφικές παραστάσεις, επίλυση γραμμικών συστημάτων, κ.α.):

- Ανάγκη για αποθήκευση/διαχείριση πολλών δεδομένων.
- Για την αποτελεσματικότερη (κυρίως από πλευράς ταχύτητας επεξεργασίας, αλλά και προγραμματιστικής ευκολίας), διαχείριση τους απαιτείται η οργάνωση των δεδομένων ανάλογα με:
  - τις μεταξύ τους σχέσεις·
  - τη σειρά επεξεργασίας τους·
  - τον τρόπο και το μέσον αποθήκευσης τους.
- Δομές δεδομένων.

Μονοδιάστατοι πίνακες (array)

- Η απλούστερη δομή (συλλογή) δεδομένων είναι ο 1-διάστατος πίνακας (array) που είναι:

  Διατεταγμένη Η σειρά των στοιχείων είναι σημαντική.

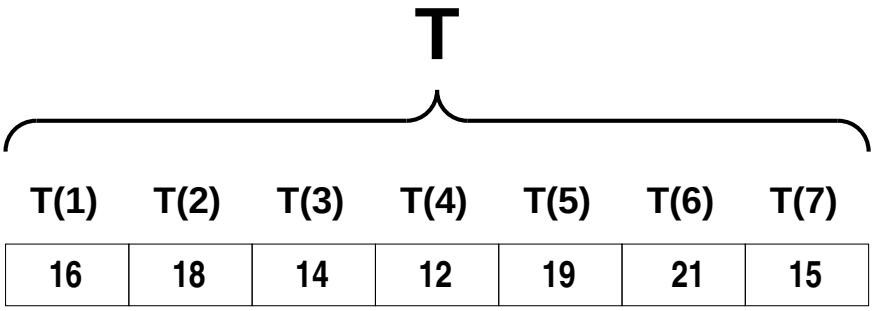
  Ομογενής Ολα τα δεδομένα είναι του ιδίου τύπου.
- Αναλογία με διανύσματα στα μαθηματικά.
- Χρησιμοποιούνται όταν θέλουμε να φυλάξουμε τα δεδομένα μας για περαιτέρω υπολογισμούς στο προγραμμά μας. Π.χ.
  - Οταν θέλω απλώς να προσθέσω 10 αριθμούς χρησιμοποιώ μια μόνο μεταβλητή για όλα τα δεδομένα μου.
  - Οταν θέλω να συγκρίνω καθένα από τους 10 αριθμούς με τον μέσο όρο τους, χρησιμοποιώ ένα array 10 θέσεων.

Παράδειγμα

Για να διαχειριστούμε τις ημερήσιες θερμοκρασίες μιας εβδομάδας

ΚΥ	ΔΕ	ΤΡ	ΤΕ	ΠΕ	ΠΑ	ΣΑ
16	18	14	12	19	21	15

Χρησιμοποιείται ο 1-διάστατος πίνακας T



όπου η θερμοκρασία της Κυριακής είναι η `T(1)`, της Δευτέρας η `T(2)`, ...
Τα στοιχεία του πίνακα αποθηκεύονται σε ...

... διαδοχικές και συνεχόμενες θέσεις μνήμης.

Arrays στο MATLAB

- Δημιουργούνται:
  - άμεσα από τον χρήστη/προγραμματιστή

    ```
    a = [ 10, 20, 30, 40 ];
    ```

  - από συναρτήσεις του MATLAB. Π.χ.

    ```
    x = rand(1,10)    % με 10 τυχαία στοιχεία
    y = zeros(1,5)    % με 5 μηδενικά στοιχεία
    z = ones(1,12)    % με 12 στοιχεία ίσα με το 1
    ```

- το στοιχείο i ενός array `a` είναι το `a(i)` και είναι στην ουσία μια μεταβλητή που μπορεί να πάρει τιμές και να συμμετέχει σε αριθμητικές εκφράσεις. Ο δείκτης `i` μπορεί να είναι σταθερά, μεταβλητή και γενικά μια αριθμητική έκφραση με ακέραια τιμή.
- Οι δείκτες ξεκινούν από το 1.
- Μήκος array: `length(a)`

Ο τελεστής : (άνω-κάτω τελεία)

Δημιουργεί arrays ισαπέχουσες τιμές με δεδομένα το πρώτο και τελευταίο στοιχείο τους και προαιρετικά το βήμα.

```
J:K        % ισοδ. με  [J, J+1, ..., K]
J:D:K      % ισοδ. με  [J, J+D, ..., J+m*D],
           % όπου  m = fix((K-J)/D)
```

Παραδείγματα:

```
k = 1:5          % ισοδ. με k = [1 2 3 4 5]
l = 100:103      % ισοδ. με l = [100 101 102 103]
m = -2:2:4       % ισοδ. με m = [-2 0 2 4]
n = 10:-2:4      % ισοδ. με n = [10 8 6 4]
t = 2:2:9        % ισοδ. με t = [2 4 6 8]
```

Δημιουργείται το κενό array `[]` όταν

- `J:K` και `J > K`
- `J:D:K` και `(D==0) || (D>0 && J>K) || (D<0 && J<K)`

Ο μηχανισμός της εντολής `for`

Η εντολή

```
for i = 1:n
```

είναι στην ουσία η

```
v = 1:n;    % δημιουργείται το διάνυσμα  v
for i = v
```

Δηλαδή

- Δημιουργείται αρχικά ένα array, έστω `v`, με τις τιμές που αντιστοιχούν σε κάθε επανάληψη.
- Η μεταβλητή `i` παίρνει διαδοχικά τις τιμές του `v`.
- Θα γίνουν `length(v)` επαναλήψεις.

Είσοδος/έξοδος μονοδιάστατων πινάκων

- **Είσοδος** (Ενα-ενα τα στοιχεία)

```
for i=1:7
   message = sprintf('Δώσε στοιχείο %g ', i);
   t(i)=input(message);
end
```

Ο χρήστης περιμένει το μήνυμα και εισάγει ένα-ένα τα στοιχεία του `t`

- **΄Εξοδος** του στοιχείου `t(i)`:

```
disp(t(i))
```

- **΄Εξοδος** όλων των στοιχείων

```
for i = 1:7
   fprintf('t(%g)=%g\n', i, t(i));
   % disp(t(i))
end
```


sprintf

Η **sprintf** είναι ακριβώς όπως και η `printf`, αλλά το αποτέλεσμα δεν εμφανίζεται στην οθόνη αλλά αποθηκεύεται σε κάποια αλφαριθμητική μεταβλητή (π.χ. όπως η `message` προηγουμένως).

Είσοδος/έξοδος μονοδιάστατων πινάκων (συνεχ.)

- **Είσοδος** Ολόκληρος ο πίνακας

```
t = input('Δώσε τον πίνακα t');
```

Ο χρήστης περιμένει το μήνυμα και εισάγει ολόκληρο τον πίνακα:

```
[16 18, 14, 12 19 21 15]
```

Τα στοιχεία του πίνακα διαχωρίζονται με κενά ή κόμμα.

- **Εξοδος** ολόκληρου του πίνακα:

```
fprintf('%g %g %g\n', t); % 3 στοιχεία ανά γραμμή
% disp(t)
```

Επεξεργασία με μονοδιάστατους πίνακες

```
% Αποκλίσεις από μέση ημερήσια θερμοκρασία για μια εβδομάδα
% T    - πίνακας θερμοκρασιών
% mt   - μέση ημερήσια θερμοκρασία
% DT   - διαφορές θερμοκρασιών από μέση τιμή

T = input('Δώσε τον πίνακα θερμοκρασιών ');
sum = 0;
for i=1:7,
   sum = sum + T(i);
end
mt = sum/7;
disp('Μέση ημερήσια θερμοκρασία');  disp(mt);
for i=1:7,
   DT(i) = T(i) - mt;
end
disp('Θερμοκρασιακές διαφορές');  disp(DT);
```


Μέγιστο και ελάχιστο στοιχείο μονοδιάστατου πίνακα

Ο αλγόριθμος στη γενικότερη μορφή του:

```
list = input('Δώσε τη λίστα');
small = list(1);        % το ελάχιστο στοιχείο της λίστας
large = list(1);        % το μέγιστο στοιχείο της λίστας
for i=2:length(list),   % πλήθος στοιχείων με την size
   if list(i) < small,
      small = list(i);
   else,
      if list(i) > large,
         large = list(i);
      end
   end
end
```

Παραδείγματα με arrays

Εσωτερικό γινόμενο διανυσμάτων

```
p = 0;
for i = 1:length(x)
   p = p + x(i)*y(i);
end
```

Μέγιστο στοιχείο

```
max = -inf;
for i = 1:length(a)
   if a(i) > max, max = a(i); end
end
```

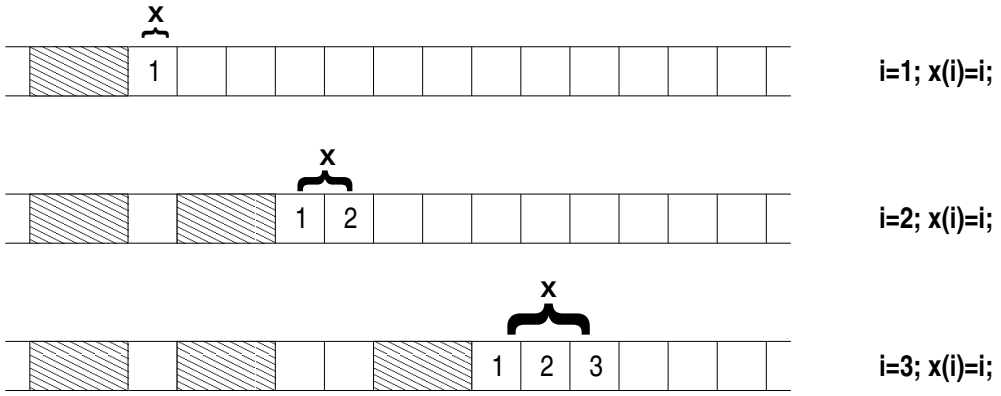
Αντιστροφή σειράς

```
N = length(b);
for i = 1:floor(N/2)
   temp = b(i);  b(i) = b(N-i+1);  b(N-i+1) = temp;
end
```


Αρχικοποίηση πινάκων

```
for i=1:100,
   x(i) = i;
   % εντολές που δεσμεύουν μνήμη
end
```

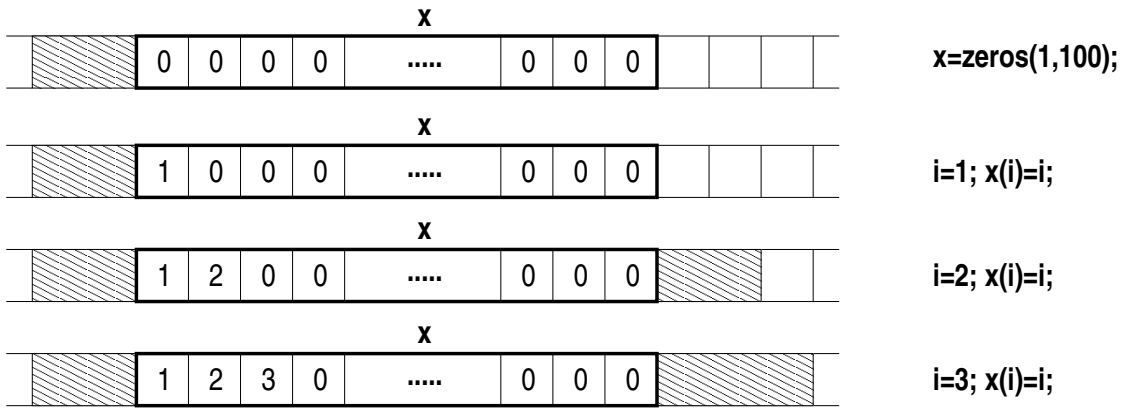
Η εικόνα της μνήμης:



Αρχικοποίηση πινάκων (συνεχ.)

```
% δημιουργία μηδενικού διανύσματος 100 θέσεων
x = zeros(1,100);
for i=1:100,
   x(i) = i;
   % εντολές που δεσμεύουν μνήμη
end
```

Η εικόνα της μνήμης:



Το κόσκινο του Ερατοσθένη (3ος αιώνας π.Χ.)

Αλγόριθμος για εύρεση όλων των πρώτων αριθμών από $2 - N$.

Βήμα 1 Γράψε τους αριθμούς.

Βήμα 2 Κύκλωσε τον πρώτο διαθέσιμο αριθμό στη λίστα (θα είναι σίγουρα πρώτος).

Βήμα 3 Διέγραψε τα πολλαπλάσια του (είναι σύνθετοι).

Βήμα 4 Πήγαινε στο **Βήμα 2**.

Παράδειγμα για $N = 10$:

$2, 3, 4, 5, 6, 7, 8, 9, 10$

$\boxed{2}, 3, \cancel{4}, 5, \cancel{6}, 7, \cancel{8}, 9, \cancel{10}$

$\boxed{2}, \boxed{3}, \cancel{4}, 5, \cancel{6}, 7, \cancel{8}, \cancel{9}, \cancel{10}$

...

Οι κυκλωμένοι αριθμοί είναι πρώτοι και οι διαγραμμένοι σύνθετοι.

```
N = input('N? ');
P = 1:N;         % estw oti oloi einai prwtoi
P(1) = 0;        % o 1 den einai prwtos
for i = 2:sqrt(N)
   if P(i)
      for j = 2*i:i:N,  P(j) = 0;  end
   end
end

% metrhma twn prwtwn pou brejhkan
NP = 0;
for i = 2:N
   if P(i), NP = NP + 1; end
end
fprintf('Found %g primes\n', NP);

% antigrafh twn prwtwn se neo array
Primes = zeros(1, NP);
k = 0;
for i = 2:N
   if P(i)
      k = k + 1;
      Primes(k) = i;
   end
end
disp(Primes)
```


Δισδιάστατοι πίνακες

	Μαθήματα Πληροφορικής		
ΑΜ Φοιτητή	Πληροφορική I	Πληροφορική II	Δομές Δεδομένων
201000	5	9	2
222222	10	8	0
250001	7	2	5
...	...	...	...
230712	6	6	6

- Αν n είναι το πλήθος των φοιτητών που εξετάστηκαν στα 3 αυτά μαθήματα πληροφορικής, τότε τα παραπάνω δεδομένα παριστάνονται σαν ένας πίνακας $n \times 3$ (n γραμμών και 3 στηλών).
- Κάθε γραμμή του πίνακα έχει τον ίδιο αριθμό στηλών: ο φοιτητής 222222 ή έχει πάρει 0 στο μάθημα 'Δομές Δεδομένων' δεν έχει εξεταστεί σε αυτό.

Ειδική περίπτωση: μονοδιάστατοι πίνακες

Ενας μονοδιάστατος πίνακας X μπορεί είναι

- ένας πίνακας $1 \times n$ με μια γραμμή και n στήλες (όπως προηγουμένως).
- ένας πίνακας $n \times 1$ με και n γραμμές και 1 στήλη.

Αναθέσεις τιμών σε μεταβλητές πινάκων

- Μονοδιάστατοι πίνακες γραμμή

```
x = [1 2 3];
y = [5, 5, 6, 7, 8, 9];
```

- Μονοδιάστατοι πίνακες στήλη

```
z = [4
     5];
w = [10; 20; 30; 40];
```

- Δισδιάστατοι πίνακες

```
A = [1 2 3
     4 5 6];
B = [1 2 3; 4, 5 6];
```

- Αναθέσεις τιμών σε μεμονωμένα στοιχεία ενός πίνακα:

```
x(1) = 8.5;
A(2,1) = 11
```


Είσοδος/έξοδος δισδιάστατων πινάκων

- Ενα-ενα τα στοιχεία

```
for i=1:100
   for j=1:3
      M(i,j)=input('Δώσε στοιχείο');
   end
end
for i=1:100
   for j=1:3, disp(M(i,j)); end
end
```

Ο χρήστης περιμένει το μήνυμα και εισάγει ένα-ένα τα στοιχεία του M

- Ολόκληρος ο πίνακας

```
M = input('Δώσε τον πίνακα M');
disp(M);
```

Ο χρήστης περιμένει το μήνυμα και εισάγει ολόκληρο τον πίνακα.

Επεξεργασία με δισδιάστατους πίνακες

Δίνονται οι βαθμολογίες 100 φοιτητών που εξετάστηκαν σε 6 μαθήματα πληροφορικής. Να βρεθει:

1. Ο μέσος όρος του κάθε φοιτητή.
2. Το μάθημα στο οποίο πέρασαν οι περισσότεροι φοιτητές.

```
% M                - pinakas bajmologiwn
% numpass          - arijmos foithtwn pou perasan sto majhma i
% student_friendly - majhma me tous perissoterous epituxontes

M = input('Πίνακας βαθμολογιών');

for i=1:100,             % gia kaje foithth
   % upologise to meso oro
   sum = 0;
   for j=1:6, sum = sum + M(i,j); end
   disp(sum/6);
end

% arijmoc epituxontwn se kaje majhma
for j=1:6,
   numpass(j) = 0;
   for i=1:100,
      if M(i,j) >= 5,
         numpass(j) = numpass(j) + 1;
      end
   end
end
student_friendly = 1;
for i=2:6,
   if numpass(i) > numpass(student_friendly),
      student_friendly = i;
   end
end
disp(student_friendly);
```


Χρήσιμες συναρτήσεις

- **`size`** - Χρησιμοποιείται για την εύρεση του αριθμού των στηλών ή των γραμμών ενός πίνακα.

  Αν A είναι ένας δισδιάστατος πίνακας $p \times q$ τότε:

  - **`size`**`(A,1)` δίνει τον αριθμό των γραμμών p του A, και
  - **`size`**`(A,2)` δίνει τον αριθμό των στηλών q του A.

  Το πλήθος των στοιχείων ενός μονοδιάστατου πίνακα-γραμμή είναι **`size`**`(x,2)`, και ενός πίνακα-στήλη **`size`**`(x,1)`. Η απλούστερα και στις 2 περιπτώσεις **`length`**`(x)`.

- **`zeros`**`(M,N)` - Μηδενικός πίνακας $M \times N$.
- **`ones`**`(M,N)` - Πίνακας $M \times N$ με στοιχεία μονάδες.
- **`zeros`**`(N)`, **`ones`**`(N)` - Αντίστοιχα, για τετραγωνικούς πίνακες.

Συναρτήσεις με παραμέτρους πίνακες

Εσωτερικό γινόμενο διανυσμάτων:

```
function p = inner_product(x, y)
p = 0;
for i = 1:length(x)
   p = p + x(i)*y(i);
end
```

Παράδειγμα κλήσης:

```
a = [1.0, 2.5, 3.1, 4.8];
b = [0.2, 3.4, 8.0, 2.0];
t = inner_product(a, b);
```


Συναρτήσεις που επιστρέφουν διανύσματα

Πολ/σμός πίνακα με διάνυσμα:

```
function y = matvec(A, x)
M = size(A,1);  % arijmos grammwn
N = size(A,2);  % arijmos sthlwn
y = zeros(M,1);
for i = 1:M
   for j = 1:N
      y(i) = y(i) + A(i,j)*x(j);
   end
end
```


Συναρτήσεις που επιστρέφουν πίνακες

Πολλαπλασιαμός πινάκων $C = AB$

```
function C = matmult(A, B)
M = size(A,1);  % arijmos grammwn A
N = size(A,2);  % arijmos sthlwn A
P = size(B,2);  % arijmos sthlwn B
C = zeros(M, P);
for j = 1:P
    for i = 1:M
       for k = 1:N
          C(i,j) = C(i,j) + A(i,k)*B(k,j);
       end
    end
end
```

Παράδειγμα

Μέθοδος που δέχεται 2-διάστατο πίνακα ακεραίων και επιστρέφει 1-διάστατο πίνακα (διάνυσμα) με στοιχεία τους μέσους όρους των γραμμών του 2-διάστατου πίνακα.

```
function avg = avgArray(arr)
M = size(arr,1);  % arijmos grammwn
N = size(arr,2);  % arijmos sthlwn
avg = zeros(1,M);
for i = 1:M
   for j = 1:N
       avg(i) = avg(i) + arr(i,j);
   end
   avg(i) = avg(i) / N;
end
```

Ποια θα είναι η αντίστοιχη μέθοδος για μ.ο. στηλών?

Γραφικές παραστάσεις: η εντολή plot

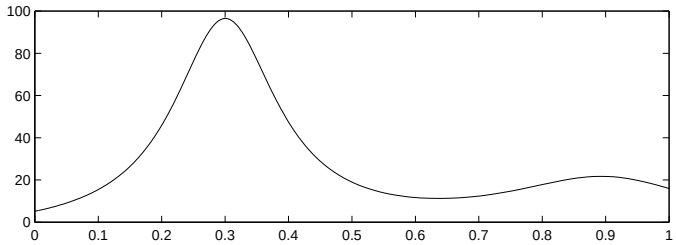
Αν $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$ είναι οι συντεταγμένες των σημείων αποθηκευμένες στους μονοδιάστατους πίνακες x και y,
τότε η εντολή

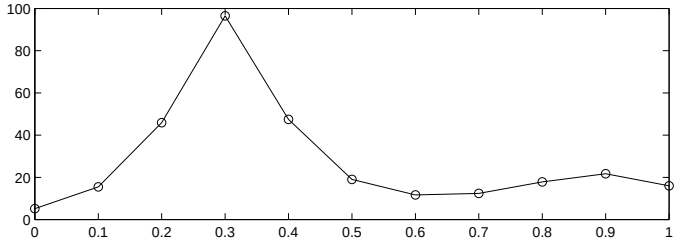
```
plot(x, y)
```

ενώνει γραφικά, με ευθύγραμμα τμήματα το σημείο (x_1, y_1) με το (x_2, y_2), το σημείο (x_2, y_2) με το (x_3, y_3), ..., το σημείο (x_{n-1}, y_{n-1}) με το (x_n, y_n)

Π.χ για τη συνάρτηση:

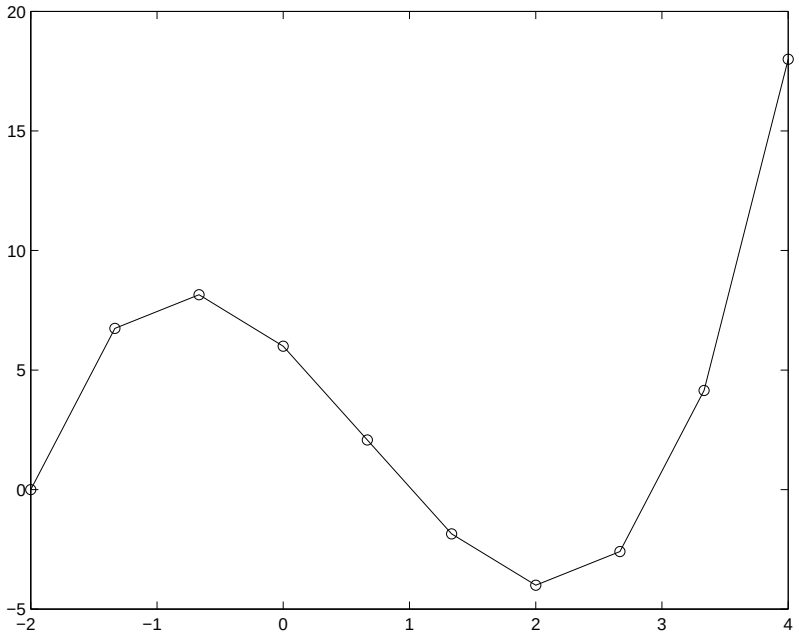
$$\frac{1}{(x-0.3)^2+0.1} + \frac{1}{(x-0.9)^2+0.4} - 6$$





Γραφική παράσταση πολυωνύμου

Εστω η: $y = x^3 - 2x^2 - 5x + 6, \qquad x \in [-2, +4].$



... χρειάζεται ο υπολογισμός τιμών (x_i, y_i) σε διακριτά σημεία.

Υπολογισμός τιμών πολυωνύμου Ι

Για δεδομένο h και $x_1 = -2,\ x_2 = x_1 + h,\ x_3 = x_2 + h,\ \ldots$

```
h = input('βήμα αύξησης του x? ');
xleft = -2;
xright = 4;
n = fix((xright-xleft)/h) + 1;  % αριθμός σημείων
x = zeros(1,n);                 % αρχικοποίηση x
y = x;                          % αρχικοποίηση y
i = 1;
for t=xleft:h:xright, % δεν υπολογίζεται πάντα το xright
   x(i) = t;
   y(i) = t^3 - 2*t^2 - 5*t + 6;
   i = i + 1;
end
plot(x,y);
```

Υπολογισμός τιμών πολυωνύμου II

Για δεδομένο n και $x_1 = -2,\ x_2,\ x_3,\ \ldots,\ x_n = +4$.

```
n = input('αριθμός σημείων? ');
xleft = -2;
xright = 4;
h = (xright-xleft)/(n-1);      % βήμα x
x = zeros(1,n);                % αρχικοποίηση x
y = x;                         % αρχικοποίηση y
i = 1;
for t=xleft:h:xright,
   x(i) = t;
   y(i) = t^3 - 2*t^2 - 5*t + 6;
   i = i + 1;
end
plot(x,y);
```


Υπολογισμός τιμών πολυωνύμου III

Για δεδομένο n και $x_1 = -2,\ x_2,\ x_3,\ \ldots,\ x_n = +4$.
Με τη συνάρτηση **linspace**

```
n = input('αριθμός σημείων? ');
xleft = -2;
xright = 4;
x = linspace(xleft, xright, n); %
y = zeros(1,n);                 % αρχικοποίηση y
for i = 1:length(x)
   t = x(i);
   y(i) = t^3 - 2*t^2 - 5*t + 6;
end
plot(x,y);
title('Grafikh parastash');
xlabel('x'); ylabel('f(x)');
```


Ο κανόνας του Horner

Σε κάθε επανάληψη (για κάθε σημείο (x_i, y_i)), για τον υπολογισμό της τιμής της έκφρασης:

```
y(i) = t^3 - 2*t^2 - 5*t + 6;
```

γίνονται 5 πολλαπλασιασμοί (2 για `t^3`, 2 για `2*t^2` και 1 για `5*t`).
Αν η εκφραση υπολογιστεί ισοδύναμα σαν:

```
y(i) = ((t-2)*t - 5)*t + 6;
```

τότε εκτελούνται μόνο 2 πολλαπλασιασμοί.
Προφανώς: μεγαλύτερο υπολογιστικό όφελος για πολυώνυμα μεγάλου βαθμού και/ή πολλές επαναλήψεις (σημεία).

Υπολογισμός ρητών συναρτήσεων

Για $h = 0.25$, υπολογισμός: $y = (x^3 + 2x^2 - 5x + 6)/(x^2 - x - 2)$, στο $[-4, 3]$

```
h = 0.25;  xleft = -4;  xright = 3;
n = fix((xright-xleft)/h) + 1;  % αριθμός σημείων
x = zeros(1,n);  y = x;  i = 1;
for t=xleft:h:xright,
   x(i) = t;
   denom = t^2 - t - 2; % υπολογισμός παρανομαστή
   if denom == 0,
      disp('η συνάρτηση δεν ορίζεται στο:');  disp(t);
   else,
      y(i) = (t^3 - 2*t^2 - 5*t + 6) / denom;
      disp(x(i));  disp(y(i));
   end
   i = i + 1;
end
```

Σημειώματα

Σημείωμα Αναφοράς

Copyright Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών, Μιχάλης Δρακόπουλος, 2014.
Μιχάλης Δρακόπουλος. «Πληροφορική Ι. Ενότητα 4: Πίνακες (arrays)». Έκδοση: 1.0. Αθήνα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: http://opencourses.uoa.gr/modules/document/?course=MATH105.

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».

[1] http://creativecommons.org/licenses/by-nc-sa/4.0/

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

- Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης