

Εργαστήριο: Προγραμματισμός Η/Υ

Παραδείγματα απλών προγραμμάτων σε Turbo Pascal

Β. Ν. Νικολαΐδης

Πρόσθετες εκφωνήσεις από: Σ. Οικονομόπουλο, Β. Καψάλη, Μ. Κεσόγλου.

Ver.0.2.6

Εισαγωγή

Η Pascal είναι μια γλώσσα που σχεδιάστηκε από τον N. Wirth το 1968, για να χρησιμοποιηθεί στην εκμάθηση προγραμματισμού υπολογιστών. Παρά την ηλικία της, η Pascal είναι ένα καλό εργαλείο εκπαίδευσης στις βασικές αρχές του προγραμματισμού γιατί:

- Είναι σχετικά απλή.
- Όπως πολλές πιο σύγχρονες γλώσσες, είναι αυστηρή στην εφαρμογή κανόνων που σχετίζονται με τον τύπο των μεταβλητών (είναι “strongly typed”).
- Όπως πολλές πιο σύγχρονες γλώσσες, προσανατολίζεται στην χρήση υπορουτινών (είναι “δομημένη γλώσσα - structured language”).
- Υπάρχουν και χρησιμοποιούνται ακόμα σύγχρονες μετεξελιξίσεις της Pascal, όπως το Delphi, που υποστηρίζουν αντικειμενοστραφή προγραμματισμό, γραφικά, προγραμματισμό για παραθυρικό περιβάλλον ή βάσεις δεδομένων κλπ.
- Λόγω της κοινής καταγωγής τους από την γλώσσα προγραμματισμού Algol, στη Pascal οι βασικές εντολές είναι παρόμοιες με αυτές άλλων δημοφιλών γλωσσών προγραμματισμού όπως η C, C++, C#, Java κλπ.

Κάθε πρόγραμμα Pascal ακολουθεί πάντα μια τυποποιημένη μορφή παρόμοια με την παρακάτω:

program atest (input, output)	Τίτλος
const pi=3.14;	Ορισμός σταθερών
type HMEPA=(DEYTEPA, TPITH, TETAPTH) ;	Ορισμός τύπων
var a,b:integer; day:HMEPA; s:string;	Ορισμός μεταβλητών
begin <εδω γράφεται το κύριο μέρος του προγράμματος> end.	Κύριο μέρος

Στα ονόματα σταθερών, τύπων, ή μεταβλητών επιτρέπεται η χρήση λατινικών χαρακτήρων, αριθμών, κλπ αλλά δεν επιτρέπεται η χρήση κενών και κάποιων συμβόλων.

Οι εντολές (reserved words) της Pascal είναι οι:

Absolute	And	Array	Begin
Case	Const	Div	Do
Downto	Else	End	External
File	For	Forward	Function
Goto	If	In	Inline
Label	Mod	Nil	Not
Of	Or	Overlay	Packed
Procedure	Program	Record	Repeat
Set	Shl	Shr	String
Then	To	Type	Until
Var	While	With	Xor

Επιπρόσθετα στην Turbo Pascal υπάρχουν τυπικές συναρτήσεις όπως οι Abs, ClrScr, Eof, Sqrt, Write, WriteLn που παρέχουν συνήθεις λειτουργίες που απαιτούνται από τον προγραμματιστή. Περισσότερες πληροφορίες για αυτές υπάρχουν στη βοήθεια της Turbo Pascal (πατώντας F1), ή και στο internet (όπως για παράδειγμα στη διεύθυνση <http://www.clipx.net/ng/pascal/index.php>)

Η Pascal και άλλες παρεμφερείς γλώσσες προγραμματισμού

Η Pascal μπορεί να είναι αρχή για την εξερεύνηση άλλων γλωσσών. Το παρακάτω παράδειγμα είναι ένα απλό πρόγραμμα (τυπώνει τη διαίρεση του 1 με όλους τους αριθμούς από το 1 έως το 100) γραμμένο σε διάφορες παραλλαγές. Φαινομενικά υπάρχουν πολλές διαφορές, όπως και πολλές ομοιότητες στα ουσιαστικά σημεία.

<i>Pascal</i>	<i>Javascript</i>
<pre> program unitdiv; uses wincrt; var i:integer; r:real; begin for i:=1 to 100 do begin r:=1/i; write('Ενα δια ',i,' ίσον ',r); writeln; end; end. </pre>	<pre> <html> <body> <script type="text/javascript"> var i=0 var r=0 for (i=1;i<=100;i++) { r = 1/i; document.write("Ενα δια " + i + " ίσον ",r) document.write("
") } </script> </body> </html> </pre>
«Γλώσσα»	<i>Java</i>
<pre> προγραμμα unitdev μεταβλητες ακεραιες : i πραγματικες: r αρχη για i απο 1 μεχρι 100 r:=1/i γραψε 'ενα δια ',i,' ίσον ',r τελος_επαναληψης τελος_προγραμματος unitdev </pre>	<pre> public class unitdev { public static void main(String args[]) { int i; float r; for(i=1;i<=100;i++) { r=1/(float)i; System.out.print("Ενα δια "+i+" ίσον "+r); System.out.println(); } } } </pre>
<i>C/C++</i>	<i>C#</i>
<pre> #include <stdio.h> main() { int i; float r; for(i=1;i<=100;i++) { r=1.0/i; printf("Ενα δια %d ίσον %f",i,r) printf("\n") } } </pre>	<pre> using System; namespace unitdev { class unitdev { static void Main(string[] args) { int i; float r; for(i=1;i<=100;i++) { r=1/(float)i; Console.Write("Ενα δια "+i+" ίσον "+r); Console.WriteLine(); } } } </pre>

Παράδειγμα 1:

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα που να διαβάζει από το πληκτρολόγιο το όνομα, το επώνυμο και την ηλικία σας, και θα τα εμφανίζει στην οθόνη. Για τα ονόματα θα επιτρέπουμε μήκος έως 20 χαρακτήρες.

Παρατήρηση: Στο παράδειγμα βλέπουμε τις εντολές `read` και `readln` που μας επιτρέπουν να πάρουμε τιμή για κάποια μεταβλητή από το πληκτρολόγιο. Έτσι τη στιγμή που ο υπολογιστής εκτελεί μία εντολή `read(x)`, ότι πληκτρολογήσει ο χρήστης αποθηκεύεται στην μεταβλητή `x`. Η Οι εντολές `write` και `writeln` μας επιτρέπουν να «γράψουμε» δεδομένα στη οθόνη. Το επίθεμα `-ln` στις εντολές αντιστοιχεί στη λέξη `line` (γραμμή), και χρησιμοποιείται όταν θέλουμε το πρόγραμμα να διαβάζει από (`readln`) ή γράφει σε (`writeln`) μία νέα γραμμή μετά την εκτέλεση της εντολής.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω

```
program name;
uses wincrt;
var ono, epo : string[20];
var ili : integer;
begin
  clrscr;
  write('DWSE ONOMA:');
  readln(ono);
  write('DWSE EPONYMO:');
  readln(epo);
  write('DWSE HLIKIA:');
  readln(ili);
  writeln('ONOMA: ',ono);
  writeln('EPWNYMO:',epo);
  writeln('HLIKIA: ',ili,' ETWN');
end.
```

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

```
DWSE ONOMA:Bond
DWSE EPONYMO:James
DWSE HLIKIA:33
ONOMA: Bond
EPWNYMO: James
HLIKIA: 33 ETWN
```

program name;	Όνομα προγράμματος
uses wincrt;	Χρησιμοποιεί την οθόνη (αυτό επιτρέπει την χρήση εντολών όπως η clrscr παρακάτω) Προσοχή: Αν δεν χρησιμοποιείτε την Borland Turbo Pascal για windows (TPW) δίνετε: uses crt;
var ono, epo : string[20];	Ορισμός 2 μεταβλητών που είναι string (= σειρά χαρακτήρων κειμένου, κείμενο) με μήκος έως 20 χαρακτήρες (για το όνομα και επώνυμο)
var ili : integer;	Ορισμός 1 μεταβλητής που είναι ακέραιος αριθμός (για την ηλικία)
begin	(Αρχίζει το κύριο μέρος του προγράμματος)
clrscr;	Καθαρίζει την οθόνη
write('DWSE ONOMA:');	Εμφανίζει οδηγία (το κείμενο 'DWSE ONOMA') στο χρήστη
readln(ono);	Διαβάζει από το πληκτρολόγιο ότι όνομα δώσει ο χρήστης και το βάζει στη μεταβλητή ono.
write('DWSE EPONYMO:');	Εμφανίζει οδηγία στο χρήστη
readln(epo);	Διαβάζει από το πληκτρολόγιο ότι όνομα δώσει ο χρήστης και το βάζει στη μεταβλητή epo.
write('DWSE HLIKIA:');	Εμφανίζει οδηγία στο χρήστη
readln(ili);	Διαβάζει από το πληκτρολόγιο ότι αριθμό δώσει ο χρήστης και το βάζει στη μεταβλητή ili
writeln('ONOMA: ',ono);	Εμφανίζει ένα μήνυμα επεξήγησης (το κείμενο 'ONOMA: '), ακολουθούμενο από την τιμή που έχει αποθηκευτεί στη μεταβλητή ono
writeln('EPWNYMO:',epo);	Όπως παραπάνω, εμφανίζει ένα μήνυμα επεξήγησης μαζί με την μεταβλητή epo
writeln('HLIKIA: ',ili,' ETWN');	Όπως παραπάνω, εμφανίζει ένα μήνυμα επεξήγησης μαζί με την μεταβλητή epo
end.	(Τέλος κύριου μέρους του προγράμματος)

Παράδειγμα 2:

Πρόγραμμα που διαβάζει από το πληκτρολόγιο την τιμή ενός πραγματικού αριθμού x και υπολογίζει την τιμή της συνάρτησης

$$y = \frac{e^{-x} + x^2 + \sqrt{x}}{\ln x + |x|}$$

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program expression;
uses wincrt;
var x,y : real;
begin
  clrscr;
  write('DWSE ARI8MO:');
  readln(x);
  y := ( exp(-x)+sqr(x)+sqrt(x) ) / ( ln(x)+abs(x) );
  writeln('TO APOTELESMA THS EXISWSHS EINAI ', y:0:2);
end.
```

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

```
DWSE ARI8MO: 4
TO APOTELESMA THS EXISWSHS EINAI      3.35
```

Επεξήγηση του προγράμματος

<code>program expression;</code>	<i>Όνομα προγράμματος, στο όνομα του προγράμματος δεν επιτρέπονται κενά</i>
<code>uses wincrt;</code>	<i>Χρησιμοποιεί τη βιβλιοθήκη <code>wincrt</code> (που περιέχει υπορουτίνες για την οθόνη και το πληκτρολόγιο σε <code>windows</code>)</i>
<code>var x,y : real;</code>	<i>Ορισμός 2 μεταβλητών που είναι πραγματικοί αριθμοί. Στο όνομα των μεταβλητών δεν επιτρέπονται κενά</i>
<code>begin</code>	<i>(Αρχίζει το κύριο μέρος του προγράμματος)</i>
<code>clrscr;</code>	<i>Καθαρίζει την οθόνη</i>
<code>write('DWSE ARI8MO:');</code>	<i>Εμφανίζει οδηγία στο χρήστη</i>
<code>readln(x);</code>	<i>Διαβάζει από το πληκτρολόγιο ότι αριθμό δώσει ο χρήστης και το βάζει στο <code>x</code></i>
<code>y := (exp(-x)+sqr(x)+sqrt(x)) / (ln(x)+abs(x));</code>	<i>Υπολογίζει την εξίσωση και βάζει το αποτέλεσμα στο <code>y</code></i>
<code>writeln('TO APOTELESMA THS EXISWSHS EINAI ', y:0:2);</code>	<i>Εμφανίζει το <code>y</code> μαζί με ένα μήνυμα επεξήγησης. Το <code>y</code> εμφανίζεται με μόνο 2 δεκαδικά ψηφία</i>
<code>end.</code>	<i>(Τέλος κύριου μέρος του προγράμματος)</i>

Παράδειγμα 3:

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα που να διαβάζει από το πληκτρολόγιο έναν αριθμό δευτερόλεπτων και να εμφανίζει τις ώρες, λεπτά, δευτερόλεπτα που αντιστοιχούν στον αριθμό αυτό.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program seconds;
uses wincrt;
var secs,o,l,d : integer;
begin
  clrscr;
  write('DWSE DEYTEROLEPTA POY OES NA METATRAPOYN: ');
  readln(secs);
  o := secs DIV 3600;
  d := secs MOD 3600;
  l := d DIV 60;
  d := d MOD 60;
  write('TA ',secs,' DEYTEROLEPTA ANTISTOIXOYN SE ');
  writeln(o,' WRES ',l,' LEPTA KAI ',d,' DEYTERA');
end.
```

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

```
DWSE DEYTEROLEPTA POY OES NA METATRAPOYN: 4001
TA 4001 DEYTEROLEPTA ANTISTOIXOYN SE 1 WRES 6 LEPTA KAI 41 DEYTERA
```


Επεξήγηση του προγράμματος

<code>program seconds;</code>	Όνομα προγράμματος
<code>uses wincrt;</code>	Χρησιμοποιεί την οθόνη
<code>var secs,o,l,d : integer;</code>	Ορισμός 4 μεταβλητών (ακέραιων αριθμών).
<code>begin</code>	(Αρχίζει το κύριο μέρος του προγράμματος)
<code> clrscr;</code>	Καθαρίζει την οθόνη
<code> write('DWSE DEYTEROLEPTA POY 0ES NA METATRAPOYN: ');</code>	Εμφανίζει οδηγία στο χρήστη
<code> readln(secs);</code>	Διαβάζει από το πληκτρολόγιο ότι αριθμό δώσει ο χρήστης και το βάζει στο <code>secs</code> . Έτσι το <code>secs</code> έχει το συνολικό αριθμό δευτερολέπτων
<code> o := secs DIV 3600;</code>	Χρησιμοποιούμε την εντολή ακέραιης διαίρεσης (<code>DIV</code>). Διαιρεί (ακέραια) το συνολικό αριθμό δευτερολέπτων με 3600 και έτσι βρίσκει τις ώρες (που βάζει στο <code>o</code>).
<code> d := secs MOD 3600;</code>	Βρίσκει το υπόλοιπο της διαίρεσης του συνολικού αριθμού δευτερολέπτων με 3600 και το βάζει στο <code>d</code> . Ο αρχικός μας χρόνος είναι ο <code>ωρες</code> και <code>d</code> δευτερά.
<code> l := d DIV 60;</code>	Μετά, διαιρεί (ακέραια) τα δευτερόλεπτα <code>d</code> με 60 και βρίσκει τα λεπτά (που βάζει στο <code>l</code>).
<code> d := d MOD 60;</code>	Βρίσκει το υπόλοιπο της παραπάνω διαίρεσης αντικαθιστώντας τη τιμή στο <code>d</code> . Ο αρχικός μας χρόνος είναι ο <code>ωρες</code> <code>l</code> λεπτά και <code>d</code> δευτερά.
<code> write('TA ',secs,' DEYTEROLEPTA ANTISTOIXOYN SE ');</code>	Εμφανίζει τα αποτελέσματα μαζί με επεξήγηση
<code> writeln(o,' WRES ',l,' LEPTA KAI ',d,' DEYTERA');</code>	-//-
<code>end.</code>	(Τέλος κύριου μέρους του προγράμματος)

Παράδειγμα 4:

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα που να διαβάζει από το πληκτρολόγιο τρεις αριθμούς και να εμφανίζει στην οθόνη τον μεγαλύτερο από αυτούς

Παρατήρηση: Στο παράδειγμα βλέπουμε την εντολή IF που μας επιτρέπει να αλλάξουμε τη ροή των εντολών που εκτελούνται ανάλογα με κάποιο κριτήριο. Οι συνηθισμένες μορφές της IF είναι:

```
IF <κριτήριο> THEN <εντολή 1>;
```

```
IF <κριτήριο> THEN < εντολή 1> ELSE < εντολή 2>;
```

(μετάφραση: ΕΑΝ (ΥΣΧΙΕΙ ΤΟ) < κριτήριο > ΤΟΤΕ <εντολή 1> ΑΛΛΙΩΣ <εντολή 2>;)

Το αποτέλεσμα του ελέγχου του κριτηρίου μπορεί να είναι μόνο true (αληθές) ή false (αναληθές). Αν το κριτήριο (ή αλλιώς «η συνθήκη») είναι αληθές, η εντολή 1 εκτελείται, αλλιώς παρακάμπτεται και (αν υπάρχει ELSE) εκτελείται η εντολή 2. Αν πρέπει να εκτελεστεί παραπάνω από μια εντολή χρησιμοποιούμε το BEGIN...END για να ορίσουμε μπλοκ εντολών. Όπως πάντα στη Pascal, χρησιμοποιούμε το σύμβολο ; για να ορίσουμε που τελειώνει η εντολή IF. Το ; είναι αντίστοιχο της άνω τελείας και σημαίνει ότι το συγκεκριμένο κομμάτι κειμένου τελείωσε. Στην IF δεν μπαίνει ; πριν το ELSE αφού το κείμενο που αφορά την εντολή IF δεν έχει ολοκληρωθεί παρά μετά το ELSE και την εντολή 2.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program findmax;
uses wincrt;
var a,b,c,max: integer;
begin
  clrscr;
  writeln('Dwste 3 ari0moys kai meta RETURN');
  readln(a,b,c);
  max:=0;
  if (a>=b) then max:=a else max:=b;
  if (c>=max) then max:=c;
  writeln('Megalyteros = ',max);
end.
```

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

```
Dwste 3 ari0moys kai meta RETURN
-34 -324 -43
Megalyteros = -34
```

Επεξήγηση του προγράμματος

<pre>program findmax; uses wincrt; var a,b,c,max: integer; begin clrscr; writeln('Dwste 3 ari0moys kai meta RETURN'); readln(a,b,c); if (a>=b) then max:=a else max:=b; if (c>=max) then max:=c; writeln('Megalyteros = ',max); end.</pre>	<p>Όνομα προγράμματος</p> <p>Χρησιμοποιεί την οθόνη (αυτό επιτρέπει την χρήση εντολών όπως η <i>clrscr</i> παρακάτω)</p> <p>Ορισμός 4 μεταβλητών</p> <p>(Αρχίζει το κύριο μέρος του προγράμματος)</p> <p>Καθαρίζει την οθόνη</p> <p>Εμφανίζει οδηγία στο χρήστη</p> <p>Διαβάζει τους 3 αριθμούς που δίνει ο χρήστης από το πληκτρολόγιο και τους βάζει στις μεταβλητές <i>a</i>, <i>b</i>, και <i>c</i> αντίστοιχα</p> <p>Συγκρίνει τα <i>a</i> και <i>b</i> και βάζει στο <i>max</i> τη τιμή του μεγαλύτερου από τα δύο.</p> <p>Συγκρίνει τα <i>c</i> και <i>max</i> και βάζει στο <i>max</i> τη τιμή του μεγαλύτερου από τα δύο.</p> <p>Εμφανίζει το αποτέλεσμα.</p> <p>(Τέλος κύριου μέρους του προγράμματος)</p>
--	--

Παράδειγμα 5:

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα που να βρίσκει τη λύση εξισώσεων της μορφής $ax + b = 0$. (Παρατήρηση: η λύση είναι $x = -b/a$, εκτός αν $a=0$. Αν $a=0$ και $b=0$ η εξίσωση είναι αόριστη, ενώ αν $a=0$ και $b \neq 0$ η εξίσωση είναι αδύνατη)

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program eksisosh;
uses wincrt;
var
a:real;
b:real;
x:real;
begin
clrscr;
writeln('EPILYSH EKSISWSHS ax+b = 0');
writeln('DWSE a ');
read (a);
writeln('DWSE b ');
read (b);
if (a<>0) then
begin
x:=-b/a;
writeln('Yparxei lysh gia x = ',x:10:4);
end
else
if (b=0) then
writeln ('aoristh')
else
writeln('adynath');
end.
end.
```

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

```
EPILYSH EKSISWSHS ax+b =0
DWSE a
34
DWSE b
65
Yparxei lysh gia x = -1.9118
```

Επεξήγηση του προγράμματος

```
program eksisosh;
uses wincrt;

var
a:real;
b:real;
x:real;
begin

clrscr;
writeln('EPILYSH EKISISWSHS ax+b = 0');
writeln('DWSE a ');
read (a);

writeln('DWSE b ');
read (b);

if (a<>0) then

begin
x:=-b/a;
writeln('Υπαρχει lysh gia x = ',x:10:4);
end
else
if (b=0) then
writeln ('aoristh')
else
writeln('adynath');
end.
end.
```

Όνομα προγράμματος

Χρησιμοποιεί την οθόνη (αυτό επιτρέπει την χρήση εντολών όπως η `clrscr` παρακάτω)

Ορισμός 3 μεταβλητών:

(η μεταβλητή a της $ax+b$)

(η μεταβλητή b της $ax+b$)

(το x όπου θα υπολογιστεί η λύση)

(Αρχίζει το κύριο μέρος του προγράμματος)

Καθαρίζει την οθόνη

Εμφανίζει μήνυμα στο χρήστη

Εμφανίζει οδηγία στο χρήστη

Διαβάζει τη τιμή του a από το πληκτρολόγιο

Εμφανίζει οδηγία στο χρήστη

Διαβάζει τη τιμή του b από το πληκτρολόγιο

Εάν το $a \neq 0$ τότε βρίσκει και εμφανίζει τη λύση. (Επειδή θέλουμε να εκτελεστούν παραπάνω από μια εντολές αν $a \neq 0$, ενώνουμε τις εντολές αυτές μέσω `begin ... end`)

-//-

-//-

-//-

-//-

Αλλιώς (αν το $a=0$)...

... αν το $b=0$ η εξίσωση είναι αόριστη

-//-

... αν το $b \neq 0$ η εξίσωση είναι αδύνατη

-//-

(Τέλος κύριου μέρος του προγράμματος)

Παράδειγμα 6:

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα που να παίρνει 2 αριθμούς να εμφανίζει ποίος είναι ο μεγαλύτερος (ή μήνυμα αν είναι ίσοι) και μετά να εμφανίζει το μέσο όρο τους.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
PROGRAM MAX_AVE;
VAR A,B : REAL;
BEGIN
  WRITELN('DWSE ENA ARITHMO');
  READLN(A);
  WRITELN('DWSE ENA ARITHMO');
  READLN(B);
  IF (A=B) THEN
    WRITELN('ΟΙ ΑΡΙΘΜΟΙ ΕΙΝΑΙ ΙΣΟΙ')
  ELSE
    IF (A>B) THEN
      WRITELN('MEGALYTERO TO ',A :0:2)
    ELSE
      WRITELN('MEGALYTERO TO ',B :0:2);
  WRITELN('MESOS OROS          ', (A+B)/2 :0:2);
END.
```

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

```
DWSE ENA ARITHMO
65
DWSE ENA ARITHMO
23
MEGALYTERO TO      65.00
MESOS OROS         44.00
```

Επεξήγηση του προγράμματος

PROGRAM MAX_AVE;	Όνομα προγράμματος
VAR A,B : REAL;	Ορισμός 2 μεταβλητών (οι αριθμοί που θα δοθούν από το χρήστη)
BEGIN	(Αρχίζει το κύριο μέρος του προγράμματος)
WRITELN('DWSE ENA ARITHMO');	Εμφανίζει οδηγία στο χρήστη
READLN(A);	Διαβάζει τη τιμή του A από το πληκτρολόγιο
WRITELN('DWSE ENA ARITHMO');	Εμφανίζει οδηγία στο χρήστη
READLN(B);	Διαβάζει τη τιμή του B από το πληκτρολόγιο
IF (A=B) THEN	Εάν οι αριθμοί είναι ίσοι εμφανίζει μήνυμα
WRITELN('ΟΙ ARITMOI EINAI ISOI')	
ELSE	Αλλιώς (εάν οι αριθμοί δεν είναι ίσοι)...
IF(A>B) THEN	... αν το A>B εμφανίζει το A
WRITELN('MEGALYTERO TO ',A :0:2)	
ELSE	.. αλλιώς (αν το A≤B) εμφανίζει το B
WRITELN('MEGALYTERO TO ',B :0:2);	
WRITELN('MESOS OROS ',(A+B)/2 :0:2);	Τέλος (σε κάθε περίπτωση), υπολογίζει και εμφανίζει τον μέσο όρο των A και B, δηλαδή το (A+B)/2
END.	(Τέλος κύριου μέρους του προγράμματος)

Όπως συμβαίνει συχνά, υπάρχουν πολλοί τρόποι να λυθεί προγραμματιστικά ένα πρόβλημα. Για παράδειγμα, ο παρακάτω κώδικας λειτουργεί παρόμοια με τον παραπάνω (έχοντας όμως την ιδιαιτερότητα να ανταλλάσει τις τιμές των A και B αν A<B).

```
PROGRAM MAX_AVE_2;
VAR A,B,T : REAL;
BEGIN
WRITELN('DWSE ENA ARITHMO');
READLN(A);
WRITELN('DWSE ENA ARITHMO');
READLN(B);
IF (A<B) THEN
    BEGIN
        T:=A;
        A:=B;
        B:=T;
    END;
IF (A=B) THEN WRITELN('ΟΙ ARITMOI EINAI ISOI')
ELSE WRITELN('MEGALYTERO TO ',A :0:2);
WRITELN('MESOS OROS ',(A+B)/2 :0:2);
END.
```

Παράδειγμα 7:

Δίνεται συνάρτηση $f(t)$ η οποία ορίζεται ως εξής:

Εάν $t > 2$ τότε $f(t) = 2(t^2 + t) + 3 \ln t - 6$

Εάν $t = 2$ τότε $f(t) = 1$

Εάν $t < 2$ τότε $f(t) = t^2 - 3t + 1$

Να γραφεί πρόγραμμα που να υπολογίζει την $f(t)$ για t που θα δίνεται από το χρήστη.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program if_equat;
var t, f :real;
begin
  writeln('Parakalw dwste to t');
  read(t);
  if(t>2) then f := 2 * (sqr(t) + t) + 3 * ln(t) - 6;
  if(t<2) then f := 1;
  if(t=2) then f := sqr(t) - 3 * t + exp(t) + 1;
  writeln('f(t) = ', f:10:3);
end.
```

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

```
Parakalw dwste to t
34
f(t) = 2384.579
```


Επεξήγηση του προγράμματος

<code>program if_equat;</code>	Όνομα προγράμματος
<code>var t,f :real;</code>	Ορισμός 2 μεταβλητών
<code>begin</code>	(Αρχίζει το κύριο μέρος του προγράμματος)
<code> writeln('Parakalw dwste to t');</code>	Εμφανίζει οδηγία στο χρήστη
<code> read(t);</code>	Διαβάζει τη τιμή του t από το πληκτρολόγιο
<code> if(t>2) then f := 2 * (sqr(t) + t) + 3 * ln(t) - 6;</code>	Αν το $t > 2$ υπολογίζουμε τον αντίστοιχη εξίσωση και βάζουμε το αποτέλεσμα στην μεταβλητή f
<code> if(t<2) then f := 1;</code>	Αντίστοιχα όταν το $t < 2$
<code> if(t=2) then f := sqr(t) - 3 * t + exp(t) + 1;</code>	Αντίστοιχα όταν το $t = 2$
<code> writeln('f(t) = ',f:10:3);</code>	Δεν χρησιμοποιούμε <code>else</code> γιατί έχουμε καλύψει όλες τις πιθανές περιπτώσεις τιμών του t
<code>end.</code>	Τέλος (σε κάθε περίπτωση), υπολογίζει και εμφανίζει το f (Τέλος κύριου μέρους του προγράμματος)

Παράδειγμα 8:

Γράψτε πρόγραμμα με όνομα CONVERT το οποίο θα διαβάζει από το πληκτρολόγιο την τιμή έντασης ηλεκτρικού ρεύματος σε Amperes και θα τα μετατρέπει σε :

μΑ (αν η τιμή είναι μικρότερη από 0.001Α)
mA (αν η τιμή είναι μικρότερη από 1Α)
kA (αν η τιμή είναι μεγαλύτερη ή ίση με 1000 Α)

ενώ θα εμφανίζει μήνυμα αν η τιμή είναι αρνητική.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program CONVERT;
VAR A: REAL;
BEGIN
WRITELN ('DWSE ENTASE SE Amperes (A) ');
READLN (A);
WRITE('TA Amperes POY DWSATE ANTOISTIXOYN SE ');
IF (A>=1000) THEN WRITELN (A/1000:8:2,'kA')
ELSE IF (A>=1) THEN WRITELN (A:8:2,'A')
      ELSE IF (A>=0.001) THEN WRITELN (A*1000:8:2,'mA')
      ELSE IF (a<0) then writeln('APNHTIKH TIMH')
      ELSE writeln(A*1000000:8:7,'uA');
END.
```

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

```
DWSE ENTASE SE Amperes (A)
.34
TA Amperes POY DWSATE ANTOISTIXOYN SE 340.00mA
```

Επεξήγηση του προγράμματος

program CONVERT;	Όνομα προγράμματος
VAR A: REAL;	Ορισμός μεταβλητής A στην οποία θα διαβάσουμε την τιμή που θα δώσει ο χρήστης (τα amperes)
BEGIN	(Αρχίζει το κύριο μέρος του προγράμματος)
WRITELN ('DWSE ENTASE SE Amperes (A)');	Εμφανίζει οδηγία στο χρήστη
READLN (A);	Διαβάζει τη τιμή του A από το πληκτρολόγιο
WRITE('TA Amperes POY DWSATE ANTOISTIXOYN SE ');	Εμφανίζει μέρος της εξόδου (είναι κοινό σε κάθε περίπτωση)
IF (A>=1000) THEN WRITELN (A/1000:8:2,'kA')	Αν το A>1000 εμφανίζει τη μετατροπή του A σε kA (δηλ. το A/1000).
ELSE IF(A>=1) THEN WRITELN (A:8:2,'A')	Το πρόγραμμα φτάνει στο σημείο αυτό αν το A<1000. Αν το A>=1 τότε δεν χρειάζεται μετατροπή.
ELSE IF(A>=0.001) THEN WRITELN (A*1000:8:2,'mA')	Το πρόγραμμα φτάνει στο σημείο αυτό αν το A<1. Αν το A>=0.001 εμφανίζει τη μετατροπή του A σε mA..
ELSE IF (a<0) then writeln('APNHTIKH TIMH')	Το πρόγραμμα φτάνει στο σημείο αυτό αν το A<0.001. Αν το A<0 εμφανίζει μήνυμα.
ELSE writeln(A*1000000:8:7,'uA');	Το πρόγραμμα φτάνει στο σημείο αυτό αν το A>0 και (λόγω των προηγούμενων) ταυτόχρονα αν A<0.001), οπότε εμφανίζει τη μετατροπή του A σε μΑ. Επειδή εδώ τελειώνει και το αρχικό if, βάζουμε semicolon (;)
END.	Τέλος κύριον μέρος του προγράμματος)

Παράδειγμα 9:

Γράψτε πρόγραμμα με όνομα average το οποίο θα διαβάζει από το πληκτρολόγιο ένα αριθμό n , και θα υπολογίζει το $n!$ δηλαδή το $1 \cdot 2 \cdot 3 \cdot \dots \cdot n$. Επίσης να υπολογίζει το $1! + 2! + 3! + \dots + n!$

Παρατήρηση: Στο παράδειγμα βλέπουμε την εντολή FOR που μας επιτρέπει να επαναλάβουμε μια εντολή (ή σειρά εντολών) για συγκεκριμένο αριθμό επαναλήψεων. Οι συνηθισμένες μορφές της FOR είναι:

FOR <μεταβλητή>:= <αρχική τιμή> TO <τελική τιμή> DO <εντολή>;
(αν η αρχική τιμή είναι μεγαλύτερη της τελικής τιμής)

FOR <μεταβλητή>:= <αρχική τιμή> DOWNTO <τελική τιμή> DO <εντολή>;
(αν η τελική τιμή είναι μεγαλύτερη της αρχικής τιμής)

(μετάφραση: ΓΙΑ <μεταβλητή>:= <αρχική τιμή> ΕΩΣ <τελική τιμή> ΚΑΝΕ <εντολή>;;)

Η μεταβλητή είναι συνήθως integer (δεν μπορεί να είναι real). Για παράδειγμα αν έχουμε ορίσει μία μεταβλητή x που είναι ακέραιος αριθμός, το FOR $x:=5$ TO 15 DO writeln(x); δίνει στη μεταβλητή x την αρχική τιμή (5), εκτελεί την εντολή writeln, αυξάνει την τιμή της μεταβλητής x κατά 1, εκτελεί πάλι την εντολή writeln, αυξάνει την τιμή της μεταβλητής x κατά 1, κλπ μέχρι το x να γίνει μεγαλύτερο του 15. Αν πρέπει να εκτελεστεί παραπάνω από μια εντολή χρησιμοποιούμε το BEGIN...END για να ορίσουμε μπλοκ εντολών.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
PROGRAM PARAG;  
VAR I,N,P,SUM:INTEGER;  
BEGIN  
P:=1;  
SUM:=0;  
WRITE('#');  
READLN(N);  
FOR I:=1 TO N DO  
BEGIN  
P:=P*I;  
SUM:=SUM+P;  
END;  
WRITELN('TO ',N,' PARAGONTIKO EINAI ',P);  
WRITELN('TO A8POISMA TWN PARAGONTIKWN EINAI ',SUM);  
END.
```

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

#4

TO 4 PARAGONTIKO EINAI 24

TO A8POISMA TWN PARAGONTIKWN EINAI 33

Επεξήγηση του προγράμματος

PROGRAM PARAG;	Όνομα προγράμματος
VAR I,N,P,SUM:INTEGER;	Ορισμός μεταβλητών
BEGIN	
P:=1;	
SUM:=0;	Εδώ θα υπολογίσουμε το άθροισμα
WRITE('#');	Ζητά τον αριθμό n
READLN(N);	Διαβάζει το n από το πληκρολόγιο
FOR I:=1 TO N DO	Και ξεκινάμε μία επαναληπτική διαδικασία (n επαναλήψεις)...
BEGIN	.. όπου το I γίνεται 1, μετά 2 κλπ
P:=P*I;	...και το p γίνεται 1*1, ενώ στη δεύτερη επανάληψη γίνεται (1*1)*2, στην τρίτη γίνεται (1*1*2)*3, μετά (1*1*2*3)*4 κλπ. Δηλαδή στο p είναι το εκάστοτε i!
SUM:=SUM+P;	... οπότε το προσθέτουμε στην τρέχουσα τιμή του sum και αποθηκεύουμε το αποτέλεσμα πάλι στην sum.
END;	(τέλος της επαναληπτικής διαδικασίας)
Writeln('TO ',N,' PARAGONTIKO EINAI ',P);	Εμφάνιση αποτελεσμάτων. Το τελευταίο I για το οποίο έτρεξε η επαναληπτικής διαδικασία είναι το n άρα το p έχει το n!
Writeln('TO A8POISMA TWN PARAGONTIKWN EINAI ',SUM);	Και το sum έχει το 1!+2!+...+n!
END.	Τέλος προγράμματος

Παράδειγμα 10:

Γράψτε πρόγραμμα με όνομα average το οποίο θα διαβάζει από το πληκτρολόγιο n αριθμούς, και θα εκτυπώνει τον μικρότερο, την θέση του μικρότερου, και το μέσο όρο όλων των αριθμών.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program average;
uses wincrt;
var n,x,min,minpos,sum,i:integer;
begin
clrscr;
writeln('Posoi ari8moi?');
readln (n);
if(n>0) then
begin
writeln('dose 1o arithmo');
readln (x);
sum:=x;
min:=x;
minpos:=1;
for i:=2 to n do
begin
writeln('dose ',i,'o arithmo');
readln (x);
if min>x then
begin
min:=x;
minpos:=i;
end;
sum:=sum+x;
end;
writeln('o mesos oros einai', sum/n :6:2);
writeln('o mikroteros einai o ',minpos,'os ari0mos dhladh to ',min);
end;
end.
```

Αυτό το κάνουμε για να δώσουμε μια σωστή αρχική τιμή στο min (που θα θέλαμε να είναι αρχικά ίσο με +∞, μεγαλύτερο από κάθε αριθμό).
Αντί αυτού και αφού δεν θα υπάρχει min για να γίνει σύγκριση όταν «διαβάσουμε» από το χρήστη τον πρώτο αριθμό, αυτός ο αριθμός γίνεται απευθείας το νέο min.

Εκτέλεση του προγράμματος

Ενδεικτικά, η έξοδος μίας εκτέλεσης («τρεξίματος») του προγράμματος είναι:

```
Posoi ari8moi?
3
dose 1o arithmo
34
dose 2o arithmo
-3
dose 3o arithmo
23
o mesos oros einai 18.00
o mikroteros einai o 2os ari0mos dhladh to -3
```

Επεξήγηση του προγράμματος

program average;	Όνομα προγράμματος
uses wincrt;	
var n,x,min,minpos,sum,i:integer;	Ορισμός μεταβλητών
begin	
clrscr;	Καθαρίζει την οθόνη
writeln('Posoi arithmoi?');	Ερ: Πόσοι θα είναι οι αριθμοί?
readln (n);	Απ: n
if(n>0) then	Αν το n είναι 0 ή λιγότερο δεν κάνουμε τίποτα.
begin	Αν το n είναι μεγαλύτερο από 0...
writeln('dose 1o arithmo');	...διαβάζουμε τον πρώτο αριθμό
readln (x);	... δηλαδή τον x
sum:=x;	... και τον προσθέτουμε στο σύνολο των αριθμών sum
min:=x;	... και επειδή είναι ο μόνος αριθμός μέχρι στιγμής, είναι και ο μικρότερος.
minpos:=1;	... ενώ η θέση του μέχρις στιγμής μικρότερου αριθμού είναι το 1 (ο πρώτος που διαβάσαμε)
for i:=2 to n do	Έχουμε άλλους n-1 αριθμούς να διαβάσουμε οπότε ξεκινάμε μία επαναληπτική διαδικασία...
begin	
writeln('dose ',i,'o arithmo');	...διαβάζουμε τον αριθμό
readln (x);	... δηλαδή τον νέο x
if min>x then	... συγκρίνουμε με την τρέχουσα ελάχιστη τιμή (το min) και αν το x είναι μικρότερο τότε έχουμε μία νέα ελάχιστη τιμή, οπότε...
begin	
min:=x;	...κρατάμε την τιμή στο min
minpos:=i;	...και τη θέση της στο minpos
end;	(Τέλος του IF (γραμμή19))
sum:=sum+x;	Σε κάθε περίπτωση προσθέτουμε τον νέο αριθμό στο sum.
end;	(Τέλος του FOR (γραμμή15))
writeln('o mesos oros einai', sum/n :0:2);	Στο τέλος το sum έχει το άθροισμα όλων των αριθμών, οπότε ο μέσος όρος είναι sum/n. Το εμφανίζουμε,
writeln('o mikroteros einai o ',minpos,'os ari0mos dhladh to ',min);	... μαζί με τα άλλα αποτελέσματα.
end;	(Τέλος του IF (γραμμή 8))
end.	Τέλος προγράμματος

Παράδειγμα 11:

Γράψτε πρόγραμμα το οποίο θα διαβάζει από το πληκτρολόγιο 5 βαθμολογίες μαθητών και θα εμφανίζει το βαθμό του μαθητή με τον μεγαλύτερο βαθμό καθώς και τον αριθμό των μαθητών με βαθμό ≥ 18.5 . Το πρόγραμμα πρέπει να επιτρέπει την εισαγωγή μόνο έγκυρων βαθμών (δηλαδή από 0 έως 20). Αν εισαχθεί βαθμός εκτός ορίων εμφανίζει μήνυμα λάθους και ζητά πάλι το βαθμό από το χρήστη.

Παρατήρηση: Στο παράδειγμα βλέπουμε την εντολή WHILE που μας επιτρέπει να επαναλάβουμε μια εντολή (ή σειρά εντολών) εφόσον μία συνθήκη είναι αληθής. Η μορφή της WHILE είναι:

```
WHILE <κριτήριο> DO <εντολή>;
```

(μετάφραση: ΟΣΟ (ΥΣΧΙΕΙ ΤΟ) <κριτήριο> ΚΑΝΕ <εντολή>)

Όπως και στην IF, το αποτέλεσμα του ελέγχου του <κριτηρίου> μπορεί να είναι μόνο true (αληθές) ή false (αναληθές). Ο έλεγχος του κριτηρίου γίνεται πριν την εκτέλεση των εντολών. Αν και όσο το κριτήριο είναι αληθές, η εντολή εκτελείται. Η διαδικασία επαναλαμβάνεται μέχρι το κριτήριο να γίνει αναληθές. Αν πρέπει να εκτελεστεί παραπάνω από μια εντολή χρησιμοποιούμε το BEGIN...END για να ορίσουμε μπλοκ εντολών.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program ba0moi;
var ba0mos, best : real;
    pli0os, i : integer;
begin
best:=0;
pli0os:=0;
for i:=1 to 5 do
begin
write ('Dwse ba0mo gia ',i,'o ma0hth:');
readln (ba0mos);
while(ba0mos<0) or (ba0mos>20) do
begin
write ('Akyros ba8mos, parakalw dwse swsto ba0mo:');
readln (ba0mos);
end;
if(ba0mos>best) then best:=ba0mos;
if(ba0mos>=18.5) then pli0os:=pli0os+1;
end;
writeln('O kalyteros ba0mos einai to ',best:2:1);
writeln(pli0os,' ma8htes eixan 18.5 kai panw');
end.
```


Εκτέλεση του προγράμματος

```
Dwse ba0mo gia 1o ma0hth:17
Dwse ba0mo gia 2o ma0hth:18.9
Dwse ba0mo gia 3o ma0hth:-3
Akyros ba8mos, parakalw dwse swsto ba0mo:43
Akyros ba8mos, parakalw dwse swsto ba0mo:542
Akyros ba8mos, parakalw dwse swsto ba0mo:19.1
Dwse ba0mo gia 4o ma0hth:14
Dwse ba0mo gia 5o ma0hth:15
O kalyteros ba0mos einai to 19.1
2 ma8htes eixan 18.5 kai panw
```

Επεξήγηση του προγράμματος

<pre>program ba0moi;</pre>	Όνομα
<pre>var ba0mos, best : real;</pre>	Ορισμός μεταβλητών
<pre> pli0os, i : integer;</pre>	
<pre>Begin</pre>	
<pre>best:=0;</pre>	Ο καλύτερος βαθμός είναι 0
<pre>pli0os:=0;</pre>	Ο αριθμός μαθητων με 18.5 και πάνω είναι 0.
<pre>for i:=1 to 5 do</pre>	Επανάλαβε για 5 μαθητες...
<pre> Begin</pre>	
<pre> write ('Dwse ba0mo gia ',i,'o ma0hth:');</pre>	Ζήτα και..
<pre> readln (ba0mos);</pre>	...διάβασε το βαθμό
<pre> while(ba0mos<0) or (ba0mos>20) do</pre>	και όσο ο βαθμός είναι εκτός ορίων
<pre> begin</pre>	...επανάλαβε την διαδικασία:
<pre> write ('Akyros ba8mos, parakalw dwse swsto ba0mo:');</pre>	Ζήτα ένα σωστό
<pre> readln (ba0mos);</pre>	... και διάβασέ τον μετά, από το πληκτρολόγιο.
<pre> end;</pre>	
<pre> if(ba0mos>best) then best:=ba0mos;</pre>	Σώσε τον καλύτερο βαθμό
<pre> if(ba0mos>=18.5) then pli0os:=pli0os+1;</pre>	Μέτρα τους αριστούχους
<pre> end;</pre>	
<pre>writeln('O kalyteros ba0mos einai to ',best:2:1);</pre>	Δείξε τα αποτελέσματα
<pre>writeln(pli0os,' ma8htes eixan 18.5 kai panw');</pre>	
<pre>end.</pre>	

Παράδειγμα 12:

Το ίδιο πρόγραμμα (το οποίο διαβάζει από το πληκτρολόγιο 5 βαθμολογίες μαθητών κλπ) μπορεί να γίνει και χρησιμοποιώντας την εντολή REPEAT.

Παρατήρηση: Στο παράδειγμα βλέπουμε την εντολή REPEAT που (όπως και η WHILE) μας επιτρέπει να επαναλάβουμε μια εντολή (ή σειρά εντολών) όσο μία συνθήκη είναι αληθής. Η μορφή της REPEAT είναι:

```
REPEAT <εντολές> UNTIL <κριτήριο>;
```

(μεταφραση: ΕΠΑΝΕΛΑΒΕ <εντολές> ΕΩΣ (ΟΤΟΥ ΝΑ ΥΣΧΙΕΙ ΤΟ) <κριτήριο>)

Αντίθετα από την WHILE, εδώ οι εντολές εκτελούνται τουλάχιστον μια φορά και ο έλεγχος του κριτηρίου γίνεται μετά την εκτέλεση των εντολών. Όπως και στις IF/WHILE, το αποτέλεσμα του ελέγχου του <κριτηρίου> μπορεί να είναι μόνο true (αληθές) ή false (αναληθές). Οι επαναλήψεις συνεχίζονται έως ότου να ισχύσει (να γίνει αληθές) το κριτήριο. Έτσι εάν μετά την πρώτη εκτέλεση των εντολών το κριτήριο είναι αναληθές οι εντολές εκτελούνται πάλι και η διαδικασία επαναλαμβάνεται μέχρι το κριτήριο να γίνει αληθές (να ισχύει). Σημείωση: η REPEAT δεν χρειάζεται απαραίτητα BEGIN...END (μπλοκ εντολών) για να εκτελέσουμε πολλές εντολές.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program ba0moi;
var ba0mos, best : real;
    pli0os, i : integer;
begin
best:=0;
pli0os:=0;
for i:=1 to 5 do
begin
repeat
write ('Dwse ba0mo gia ',i,'o ma0hth:');
readln (ba0mos);
if (ba0mos<0) or (ba0mos>20) then write ('Akyros ba8mos.');
```

```
until (ba0mos>=0) and (ba0mos<=20);
if (ba0mos>best) then best:=ba0mos;
if (ba0mos>=18.5) then pli0os:=pli0os+1;
end;
writeln('O kalyteros ba0mos einai to ',best:2:1);
writeln(pli0os,' ma8htes eixan 18.5 kai panw');
```

```
end.
```

Εκτέλεση του προγράμματος

```
Dwse ba0mo gia 1o ma0hth:14
Dwse ba0mo gia 2o ma0hth:19
Dwse ba0mo gia 3o ma0hth:12
Dwse ba0mo gia 4o ma0hth:032
Akyros ba8mos.Dwse ba0mo gia 4o ma0hth:322
Akyros ba8mos.Dwse ba0mo gia 4o ma0hth:-3
Akyros ba8mos.Dwse ba0mo gia 4o ma0hth:15
Dwse ba0mo gia 5o ma0hth:8
O kalyteros ba0mos einai to 19.0
1 ma8htes eixan 18.5 kai panw
```

Επεξήγηση του προγράμματος

```
program ba0moi;
var ba0mos, best : real;
    pli0os, i : integer;
begin
best:=0;

pli0os:=0;

for i:=1 to 5 do
begin
repeat
write ('Dwse ba0mo gia ',i,'o ma0hth:');
readln (ba0mos);
if (ba0mos<0) or (ba0mos>20) then
write ('Akyros ba8mos.');
```

Όνομα προγράμματος

Ορισμός μεταβλητών

Ο καλύτερος μέχρι στιγμής βαθμός είναι 0

Ο αριθμός μαθητων με 18.5 και πάνω είναι 0.

Επανάλαβε για 5 μαθητες...

Επανάλαβε τα παρακάτω...

Διάβασε βαθμό

Αν είναι λάθος δείξε μήνυμα

Αν είναι λάθος γύρνα στην αρχή (στο repeat) και επανάλαβε.

Σώσε τον καλύτερο βαθμό

Μέτρα τους αριστούχους

```
until (ba0mos>=0) and (ba0mos<=20);

if(ba0mos>best) then best:=ba0mos;

if(ba0mos>=18.5) then pli0os:=pli0os+1;

end;

writeln('O kalyteros ba0mos einai to ',best:0:1);
writeln(pli0os,' ma8htes eixan 18.5 kai panw');
end.
```

Δείξε τα αποτελέσματα

Παράδειγμα 13:

Να γραφτεί πρόγραμμα το οποίο διαβάζει από το πληκτρολόγιο 10 λέξεις και μετά να τις εμφανίζει με αντίστροφη σειρά (από αυτή με την οποία τις δώσαμε).

Παρατήρηση:

Για να ορίσουμε μια μεταβλητή που είναι μονοδιάστατος πίνακας από 10 string (σειρά χαρακτήρων κειμένου, κείμενο) γράφουμε:

```
onoma_pinaka : array [1..10] of string;
```

στην περιοχή που ορίζουμε μεταβλητές.

Αντίστοιχα, για να ορίσουμε μια μεταβλητή που είναι μονοδιάστατος πίνακας από 10 ακέραιους αριθμούς γράφουμε:

```
var onoma_pinaka : array [1..10] of integer;
```

Το *onoma_pinaka* είναι ο πίνακας, το *onoma_pinaka*[1] το πρώτο στοιχείο του, το *onoma_pinaka*[10] το τελευταίο.

Για να ορίσουμε μια μεταβλητή που είναι διδιάστατος πίνακας 10x10 ακέραιων αριθμών γράφουμε:

```
var onoma_pinaka : array [1..10,1..10] of integer;
```

Το *onoma_pinaka* είναι ο πίνακας, το *onoma_pinaka* [1][1] το πρώτο στοιχείο της πρώτης γραμμής, το *onoma_pinaka* [10][10] το τελευταίο στοιχείο της τελευταίας γραμμής.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program reverse;  
const  
  n=10;  
var  
  w:array[1..n] of string;  
  i:integer;  
begin  
  for i:=1 to n do  
    begin  
      write ('Dwse le3h :');  
      readln(w[i]);  
    end;  
  for i:=n downto 1 do  
    writeln(w[i]);  
end.
```

Εκτέλεση του προγράμματος

```
Dwse le3h :computer
Dwse le3h :pascal
Dwse le3h :program
Dwse le3h :error
Dwse le3h :loop
Dwse le3h :test
Dwse le3h :data
Dwse le3h :class
Dwse le3h :object
Dwse le3h :number
number
object
class
data
test
loop
error
program
pascal
computer
```

Επεξήγηση του προγράμματος

```
program reverse;
const
  n=10;
var
  w:array[1..n] of string;

  i:integer;
begin
  for i:=1 to n do
    begin
      write ('Dwse le3h :');
      readln(w[i]);

      end;
  for i:=n downto 1 do
    writeln(w[i]);
  end.
```

Όνομα προγράμματος

Ορισμός σταθερών:

Ορισμός σταθεράς n
(πάντοτε ίση με 10)

Ορισμός μεταβλητών:

Ορισμός ενός πίνακα 10×1
από strings (για να
αποθηκεύσουμε τις λέξεις)

Για $i = 1, 2, 3, \dots, 10$ επανέλαβε...

... διάβασε μια λέξη και
σώσε τη στο $w[i]$

Για $i = 10, 9, 8, \dots, 1$ επανέλαβε...

...δειξε το $w[i]$

Παράδειγμα 14:

Να γραφτεί πρόγραμμα το οποίο θα διαβάζει τα αποτελέσματα σε 5 αγώνες ποδοσφαίρου, τα οποία θα αποθηκεύει σε πίνακα, και μετά θα τα εκτυπώνει με σχόλιο που αφορά αν υπήρξε ισοπαλία, νίκη γηπεδούχου ή επισκέπτη.

Πρόγραμμα:

Μία τέτοια λύση είναι η παρακάτω:

```
program scores;
uses wincrt;
var
  s,g:integer;
  score:array [1..5,1..2] of integer;
begin
for g:=1 to 5 do
  for s:=1 to 2 do
    begin
      write('DWSE GOALS ',s,'hs OMADAS STON ',g,'o AGWNA:');
      readln(score[g,s]);
    end;
  writeln;
  writeln('  APOTELESMATA:');
  writeln('  1o  2o  Sxolio');
  writeln;
for g:=1 to 5 do
  begin
    for s:=1 to 2 do
      write(score[g,s]:4);
      write(' ');
      if score[g,1] = score[g,2] then write ('ISOPALIA')
      else if score[g,1] > score[g,2] then write ('NIKH GHPEDOYXOY')
      else write ('NIKH EPISKEPTH');
      writeln;
    end;
  end.
end.
```

Εκτέλεση του προγράμματος

```
DWSE GOALS 1hs OMADAS STON 1o AGWNA:1
DWSE GOALS 2hs OMADAS STON 1o AGWNA:0
DWSE GOALS 1hs OMADAS STON 2o AGWNA:0
DWSE GOALS 2hs OMADAS STON 2o AGWNA:0
DWSE GOALS 1hs OMADAS STON 3o AGWNA:2
DWSE GOALS 2hs OMADAS STON 3o AGWNA:2
DWSE GOALS 1hs OMADAS STON 4o AGWNA:2
DWSE GOALS 2hs OMADAS STON 4o AGWNA:1
DWSE GOALS 1hs OMADAS STON 5o AGWNA:0
DWSE GOALS 2hs OMADAS STON 5o AGWNA:3
```

```
APOTELESMATA:
1o  2o  Sxolio

  1   0  NIKH GHPEDOYXOY
  0   0  ISOPALIA
  2   2  ISOPALIA
  2   1  NIKH GHPEDOYXOY
  0   3  NIKH EPISKEPTH
```

Επεξήγηση του προγράμματος

<pre>program scores; uses wincrt; var s,g:integer; score:array [1..5,1..2] of integer; begin for g:=1 to 5 do for s:=1 to 2 do begin write('DWSE GOALS ',g,'hs STON ',s,'o AGWNA:'); readln(score[g,s]); end; writeln; writeln(' APOTELESMATA:'); writeln(' 1o 2o Sxolio'); writeln; for g:=1 to 5 do Begin for s:=1 to 2 do write(score[g,s]:4); write(' '); if score[g,1] = score[g,2] then write ('ISOPALIA') else if score[g,1] > score[g,2] then write ('NIKH GHPEDOUXOY') else write ('NIKH EPISKEPTH'); writeln; end; end.</pre>	<p>Όνομα προγράμματος κλπ</p> <p>Ορισμός μεταβλητών:</p> <p>s για στήλη (αγώνες), g για γραμμή (ομάδες).</p> <p>...και ο πίνακας των σκορ.</p> <p>Πρώτα διαβάζουμε τους τα γκολ της κάθε ομάδας και γεμίζουμε τον πίνακα. Για 5 αγώνες...</p> <p>... και 2 αντίπαλες ομάδες</p> <p>--</p> <p>...διαβάζουμε το σκορ του γηπεδούχου στη στήλη 1 ενώ του επισκέπτη στη στήλη 2</p> <p>Μετά εμφανίζουμε μερικές επικεφαλίδες...</p> <p>--</p> <p>--</p> <p>--</p> <p>Μετά εμφανίζουμε τον ίδιο τον πίνακα. Για κάθε αγώνα...</p> <p>...εμφάνισε το αποτέλεσμα των δύο ομάδων</p> <p>.. ένα κενό...</p> <p>...και δίπλα ένα σχόλιο σχετικά με τον νικητή. Ήταν το σκορ στη πρώτη στήλη ίσο με το δεύτερο (ισοπαλία)?</p> <p>...η μήπως μεγαλύτερο? (νίκη γηπεδούχου)</p> <p>...αν τίποτε από τα παραπάνω τότε νίκησε ο επισκέπτης.</p> <p>Τέλος πάμε στην επόμενη γραμμή της οθόνης (και στον επόμενο αγώνα αν υπάρχει)</p> <p>Τέλος προγράμματος</p>
---	--

Παράδειγμα 15:

Να γραφτεί πρόγραμμα το οποίο διαβάζει από το πληκτρολόγιο 10 αριθμούς και τους ταξινομεί από το μικρότερο στον μεγαλύτερο (sorting).

Πρόγραμμα:

Υπάρχουν πολλοί τύποι sorting, ένα τέτοιο πρόγραμμα (bubblesort) είναι το παρακάτω:

```
program sort;
uses wincrt;
const
  n = 10;
var
  i, j, temp : integer;
  pinakas : array [1..n] of integer;
begin

  for i:= 1 to n do
  begin
    write ('Dwse ',i,'o ari8mo :');
    readln(pinakas[i]);
  end;

  for i := 1 to n do
    for j := (i+1) to n do
      if pinakas[j] < pinakas[i] then
        begin
          temp := pinakas[j];
          pinakas[j] := pinakas[i];
          pinakas[i] := temp;
        end;
    for i := 1 to n do
      write(pinakas[i], ' ');
    end.
end.
```

Εκτέλεση του προγράμματος

```
Dwse 1o ari8mo :39
Dwse 2o ari8mo :72
Dwse 3o ari8mo :323
Dwse 4o ari8mo :53
Dwse 5o ari8mo :2
Dwse 6o ari8mo :5
Dwse 7o ari8mo :-35
Dwse 8o ari8mo :634
Dwse 9o ari8mo :-34
Dwse 10o ari8mo :2305
-35 -34 2 5 39 53 72 323 634 2305
```


Επεξήγηση του προγράμματος

```
program sort;
uses wincrt;
const
    n = 10;

var
    i, j, temp : integer;
    pinakas : array [1..n] of integer;

begin

    for i:= 1 to n do
        begin
            write ('Dwse ',i,'o ari8mo :');

            readln(pinakas[i]);
            end;

    for i := 1 to n do
        for j := (i+1) to n do

            if pinakas[j] < pinakas[i] then

                begin
                    temp := pinakas[j];

                    pinakas[j] := pinakas[i];
                    pinakas[i] := temp;
                end;

        for i := 1 to n do
            write(pinakas[i], ' ');
        end.
```

Όνομα προγράμματος κλπ

Ορισμός σταθερών:

Ορισμός σταθεράς *n*
(πάντοτε ίση με 10)

Ορισμός μεταβλητών:

Ορισμός ενός πίνακα 10x1
από *integer* (για να
αποθηκεύσουμε τους
αριθμούς)

Πρώτα διαβάζουμε τους
αριθμούς και γεμίζουμε τον
πίνακα

Για *i = 1,2,3,...,10* επανέλαβε...

... διάβασε ένα αριθμό και
σώσε το στο *pinakas[i]*

-//-

Μετά ταξινομούμε...

Για *i = 1,2,3,...,10*

...και για *j* να ξεκινά από το
επόμενο στοιχείο του
εκάστοτε *i* έως το τέλος του
πίνακα, επανέλαβε...

... αν το στοιχείο στο *j* (δεξιά
του στοιχείου στο *i*) είναι
μικρότερο από το στοιχείο
στο *i*...

...άλλαξε το στοιχείο στο *j*
με το στοιχείο στο *i*.

-//-

-//-

Εμφάνισε τον τελικό πίνακα

Το πρόγραμμα για κάθε στοιχείο i του πίνακα, βρίσκει το μικρότερο στοιχείο του πίνακα δεξιά του i , και αλλάζει τα δύο. Καθώς τρέχει, ο πίνακας αλλάζει όπως βλέπουμε παρακάτω:

Βήμα	pinakas[1]	pinakas[2]	pinakas[3]	pinakas[4]	pinakas[5]	pinakas[6]	pinakas[7]	pinakas[8]	pinakas[9]	pinakas[10]
Αρχή	39	72	323	53	2	5	-35	634	-34	2305
i=1	2	72	323	53	39	5	-35	634	-34	2305
i=1	-35	72	323	53	39	5	2	634	-34	2305
i=2	-35	53	323	72	39	5	2	634	-34	2305
i=2	-35	39	323	72	53	5	2	634	-34	2305
i=2	-35	5	323	72	53	39	2	634	-34	2305
i=2	-35	2	323	72	53	39	5	634	-34	2305
i=2	-35	-34	323	72	53	39	5	634	2	2305
i=3	-35	-34	72	323	53	39	5	634	2	2305
i=3	-35	-34	53	323	72	39	5	634	2	2305
i=3	-35	-34	39	323	72	53	5	634	2	2305
i=3	-35	-34	5	323	72	53	39	634	2	2305
i=3	-35	-34	2	323	72	53	39	634	5	2305
i=4	-35	-34	2	72	323	53	39	634	5	2305
i=4	-35	-34	2	53	323	72	39	634	5	2305
i=4	-35	-34	2	39	323	72	53	634	5	2305
i=4	-35	-34	2	5	323	72	53	634	39	2305
i=5	-35	-34	2	5	72	323	53	634	39	2305
i=5	-35	-34	2	5	53	323	72	634	39	2305
i=5	-35	-34	2	5	39	323	72	634	53	2305
i=6	-35	-34	2	5	39	72	323	634	53	2305
i=6	-35	-34	2	5	39	53	323	634	72	2305
i=7	-35	-34	2	5	39	53	72	634	323	2305
i=8	-35	-34	2	5	39	53	72	323	634	2305

Παράδειγμα 16:

Να γραφτεί πρόγραμμα το οποίο υπολογίζει το τετράγωνο των αριθμών από το 1 έως το 10. Χρησιμοποιούμε υπορουτίνα (function) για τον υπολογισμό των τετραγώνων.

Πρόγραμμα:

Ένα πρόγραμμα που κάνει όσα ζητάμε είναι το παρακάτω:

```
program tetragwna;
uses wincrt;

var i : integer;

function tetragwno (x: Real): Real;
begin
  tetragwno := x*x;
end;

begin
  for i := 1 to 10 do
    Writeln ('to tetragwno toy ',i,' einai ',tetragwno (i):6:2);
  WriteLn;
end.
```

Εκτέλεση του προγράμματος

```
to tetragwno toy 1 einai 1.00
to tetragwno toy 2 einai 4.00
to tetragwno toy 3 einai 9.00
to tetragwno toy 4 einai 16.00
to tetragwno toy 5 einai 25.00
to tetragwno toy 6 einai 36.00
to tetragwno toy 7 einai 49.00
to tetragwno toy 8 einai 64.00
to tetragwno toy 9 einai 81.00
to tetragwno toy 10 einai 100.00
```

Επεξήγηση του προγράμματος

```
program tetragwna;  
uses wincrt;  
var i : integer;
```

Όνομα προγράμματος

Ορισμός μεταβλητών

```
function tetragwno (x: Real): Real;
```

Ορισμός μιας υπορουτίνας με όνομα *tetragwno* για τον υπολογισμό του τετραγώνου. Η ρουτίνα δέχεται (σαν «παράμετρο») έναν αριθμό *x* και επιστρέφει πάλι αριθμό.

```
Begin
```

```
    Tetragwno := x*x;
```

Αρχή υπορουτίνας

Υπολόγισε το x^2 και επέστρεψε το.

Χρησιμοποιούμε το όνομα της υπορουτίνας για να επιστρέψουμε το αποτέλεσμα

```
end;
```

Τέλος υπορουτίνας

```
Begin
```

```
    for i := 1 to 10 do
```

```
        Writeln ('to tetragwno toy ',i,' einai  
' ,tetragwno (i):6:2);
```

(Αρχίζει το κύριο μέρος του προγράμματος)

Για $i = 1,2,3,..,10$ επανέλαβε...

... κάλεσε την *tetragwno* με παράμετρο το i και δείξε το αποτέλεσμα

```
        Writeln;
```

-//-

```
end.
```

(Τέλος κύριου μέρους του προγράμματος)

Παράδειγμα 17:

Να γραφτεί πρόγραμμα το οποίο να παίζει τρίλιζα (tic-tac-toe ή X-O). Χρησιμοποιούμε διαδικασίες (procedures) για κάποια σημαντικά μέρη του προγράμματος, πίνακες, και διάφορες άλλες χρήσιμες τεχνικές που εξηγούνται παρακάτω.

Πρόγραμμα:

(Δες επεξήγηση του προγράμματος, παρακάτω)

Εκτέλεση του προγράμματος

```
___  
___  
___  
  
Dialexe grammi:33  
Dialexe stili:3  
** LA00S EPILOGH **  
3anadialexe grammi:3  
3anadialexe stili:3  
SKEFTOMAI...
```

```
_O_  
___  
_X
```

```
Dialexe grammi:3  
Dialexe stili:1  
SKEFTOMAI...
```

```
_O_  
___  
XOX
```

```
Dialexe grammi:1  
Dialexe stili:1  
SKEFTOMAI.....
```

```
XO_  
O_  
XOX
```

```
Dialexe grammi:2  
Dialexe stili:2
```

```
XO_  
OX_  
XOX
```

KERDISES!

Επεξήγηση του προγράμματος

<pre>program tictactoe; uses wincrt; {----- STA0ERES -----} const KANENAS = 0; COMPUTER = 1; ANOROPOS = 2; AGNWSTOS = 3; {----- KAOLIKES METABLHTES -----} var t : array [1..3,1..3] of integer; nikiths : integer; {-----} procedure ARXIKOPOIHSE_PINAKA; var g,s:integer; begin for g := 1 to 3 do for s := 1 to 3 do t[g][s]:=KANENAS; end; end;</pre>	<p>Όνομα προγράμματος Βιβλιοθήκη</p> <p>(Σχόλιο προγραμματιστή)</p> <p>Σταθερές 0 συμβολίζει κανέναν και ισοπαλία 1 το computer 2 τον άνθρωπο-παίχτη 3 άγνωστος νικητής (δηλαδή το παιχνίδι συνεχίζεται)</p> <p>(Σχόλιο προγραμματιστή)</p> <p>Ο πίνακας του παιχνιδιού Μεταβλητή που περιέχει το νικητή αν υπάρχει.</p> <p>(Σχόλιο προγραμματιστή)</p> <p>Διαδικασία που αρχικοποιεί τον πίνακα. Χρησιμοποιεί τοπικές μεταβλητές: g για γραμμή, s για στήλη. Για κάθε γραμμή... και στήλη... ...θέσε τη θέση ως κενή</p>
<pre>{-----} procedure EMFANISE_PINAKA; var g,s:integer; begin writeln; for g := 1 to 3 do begin for s := 1 to 3 do Begin if t[g][s]=KANENAS then write(' '); if t[g][s]=ANOROPOS then write('X'); if t[g][s]=COMPUTER then write('O'); end; writeln; end; writeln; end; end;</pre>	<p>(Σχόλιο προγραμματιστή)</p> <p>Διαδικασία που εμφανίζει τον πίνακα: Τοπικές μεταβλητές: g για γραμμή, s για στήλη. Αφήσε μία κενή γραμμή. Μετά για κάθε γραμμή του πίνακα και... για κάθε μια στήλη της γραμμής</p> <p>Γράψε κενό, X ή O ανάλογα με το τι περιέχει ο πίνακας μένοντας όμως στην ίδια γραμμή της οθόνης. Μετά πήγαινε στην επόμενη γραμμή της οθόνης.</p> <p>Τέλος αφήσε μια ακόμα κενή γραμμή οθόνης.</p>
<pre>{-----} procedure PARE_KINHSH_XRHSTH; var g,s:integer; begin write('Dialexe grammi:'); readln(g);</pre>	<p>(Σχόλιο προγραμματιστή)</p> <p>Διαδικασία που ζητά μία έγκυρη θέση για να βάλει το X του άνθρωπου-παίχτη Ζήτη γραμμή Διάβασε γραμμή</p>

```

write('Dialexe stili:');
readln(s);
while (g<1) or
      (g>3) or
      (s<1) or
      (s>3) or
      (t[g][s]<>KANENAS) do
  begin
    writeln('** LA00S EPILOGH **');
    write('3anadialexe grammi:');
    readln(g);
    write('3anadialexe stili:');
    readln(s);
    end;
  t[g][s]:=ANOROPOS;
end;

```

Ζήτα στήλη
 Διάβασε στήλη
 Όσο η γραμμή
 ή η στήλη που δόθηκε είναι
 εκτός ορίων πίνακα
 ή
 η θέση δεν είναι κενή...

... εμφανίζει μήνυμα και
 ζητά μία νέα
 θέση για να βάλει το
 X του άνθρωπου-παίχτη

Τέλος μαρκάρει τη θέση.

{-----}

```

procedure BRES_KINSHSH_YPOLOGISTH;
var g,s:integer;

```

```

begin
  g:=random(3)+1;
  s:=random(3)+1;
  write('SKEFTOMAI...');
  while (t[g][s]<>KANENAS) do
    begin
      write('.');
      g:=random(3)+1;
      s:=random(3)+1;
    end;
  writeln;
  t[g][s]:=COMPUTER;
end;

```

(Σχόλιο προγραμματιστή)
 Διαδικασία που βρίσκει μία
 τυχαία έγκυρη θέση για να
 βάλει το O του computer-
 παίχτη.
 [Όχι και η καλύτερη
 στρατηγική βέβαια]
 Ψευδοτυχαία γραμμή από το
 1 έως το 3
 Ψευδοτυχαία στήλη από το
 1 έως το 3

Όσο η θέση δεν είναι κενή
 ... δοκιμάζει νέα
 ... ψευδοτυχαία γραμμή
 ... και στήλη.

Τέλος μαρκάρει τη θέση.

{-----}

```

procedure BRES_NIKHSH;

```

```

var g,s,c:integer;
begin
  nikiths := AGNWSTOS;
  for g:=1 to 3 do
    begin
      c:= t[g][1];
      if (c<>KANENAS) and
         (t[g][2]=c) and
         (t[g][3]=c) then
        nikiths := c;
      end;
  for s:=1 to 3 do
    begin
      c:= t[1][s];

```

(Σχόλιο προγραμματιστή)
 Διαδικασία που ερευνά αν
 υπάρχει νικητής στο
 παιχνίδι (καλείται μετά από
 κάθε κίνηση)

Αρχικά θεώρησε ότι δεν
 υπάρχει νικητής..

Για κάθε γραμμή...
 βάλε στο c τον παίχτη που
 έχει καταλάβει την πρώτη
 στήλη της γραμμής.
 Αν η θέση δεν είναι κενή..
 ...και στη 2^η στήλη είναι ό
 ίδιος ... όπως και στην 3^η
 ...τότε ο παίχτης νίκησε.

Για κάθε στήλη...
 Βάλε στο c τον παίχτη που
 έχει καταλάβει την πρώτη

```

if (c<>KANENAS) and
  (t[2][s]=c) and
  (t[3][s]=c) then
  nikiths := c;
end;

c:= t[1][1];
if (c<>KANENAS) and
  (t[2][2]=c) and
  (t[3][3]=c) then
  nikiths := c;

c:= t[1][3];
if (c<>KANENAS) and
  (t[2][2]=c) and
  (t[3][1]=c) then
  nikiths := c;

if nikiths = AGNWSTOS then

  begin
  nikiths:= KANENAS;
  for g:=1 to 3 do
    for s:=1 to 3 do
      if t[g][s]=KANENAS then
        nikiths:=AGNWSTOS;
      end;
    end;
  end;

{-----}
{-----TO KYRIWS PROGRAMMA-----}
{-----}

begin

nikiths:=AGNWSTOS;
ARXIKOPOIHSE_PINAKA;
while nikiths=AGNWSTOS do
  begin
  EMFANISE_PINAKA;
  if nikiths=AGNWSTOS then PARE_KINHSH_XRHSTH;
  BRES_NIKHTH;
  if nikiths=AGNWSTOS then BRES_KINHSH_YPOLOGISTH;
  BRES_NIKHTH;
  end;

EMFANISE_PINAKA;
if(nikiths=KANENAS) then writeln('ISOPALIA!');
if(nikiths=ANOROPOS) then writeln('KERDISES!');
if(nikiths=COMPUTER) then writeln('EXASES!');

end.

```

γραμμή της στήλης.
 Αν η θέση δεν είναι κενή..
 ...και στη 2^η στήλη είναι ο
 ίδιος... όπως και στην 3^η
 ...τότε ο παίχτης νίκησε.

Βάλε στο c τον παίχτη που
 έχει καταλάβει την θέση 1,1.
 Αν η θέση δεν είναι κενή..
 ...και η θέση 2,2 είναι του
 ίδιου...όπως και η 3,3...
 τότε νίκησε (διαγώνιος).

Βάλε στο c τον παίχτη που
 έχει καταλάβει την θέση 1,3.
 Αν η θέση δεν είναι κενή..
 ...και η θέση 2,2 είναι του
 ίδιου...όπως και η 3,1...
 τότε νίκησε (2η διαγώνιος).
 Αν παρόλα αυτά δεν
 βρέθηκε νικητής, τότε

Θεώρησε ισοπαλία, αλλά
 για κάθε γραμμή και στήλη
 έλεγξε αν υπάρχει κενή
 θέση, οπότε το παιχνίδι
 συνεχίζεται και ο νικητής
 είναι ακόμα άγνωστος.

(Σχόλιο προγραμματιστή)
 Το κυρίως πρόγραμμα...

Αρχικά δεν υπάρχει νικητής
 Καλεί/τρέχει την διαδικασία
 Όσο δεν υπάρχει νικητής...

Καλεί διαδικασία εμφάνισης
 Παίξει ο χρήστης
 Καλεί διαδικασία νικητή
 Παίξει ο υπολογιστής
 Καλεί διαδικασία νικητή

Το παιχνίδι τελείως ...
 εμφάνισε πίνακα και
 ..αποτελέσματα.

Θέματα για παραδείγματα που απευθύνονται σε πιο προχωρημένα επίπεδα :

Εντολή CASE, Σύνολα (SET), Εγγραφές (RECORD), Μονο/πολυδιάστατοι πίνακες που περιέχουν εγγραφές, Εγγραφές που περιέχουν μονο/πολυδιάστατους πίνακες, Δείκτες, Δομές δεδομένων που χρησιμοποιούν τα παραπάνω.