ΑΝΟΙΚΤΑ ακαδημαϊκά μαθήματα ΠΠ

Εισαγωγή στον Προγραμματισμό Η/Υ για Χημικούς Μηχανικούς

Παρουσίαση Διαλέξεων: 3. Επιλογή

Καθηγητής Δημήτρης Ματαράς

ΕΠΙΛΟΓΗ

Stanislav Lem (1921-2006)

Δομή **IF..END IF** γενικά

```
IF(έκφραση ελέγχου1)THEN                    !αρχική πρόταση
      ....
      εκφράσεις1                  ! 1η έκταση
      ....
[ELSE IF(έκφραση ελέγχου2)THEN]
   ....
      [εκφράσεις2]                ! 2η έκταση
      ....
[ELSE]
      ....
      [εκφράσεις3]                ! 3η έκταση
      ....
END IF                                      !τελική πρόταση
```

Λογικό **IF** (πρόταση **IF**)

```
...
IF (έκφραση ελέγχου)έκφραση ή πρόταση
...
```

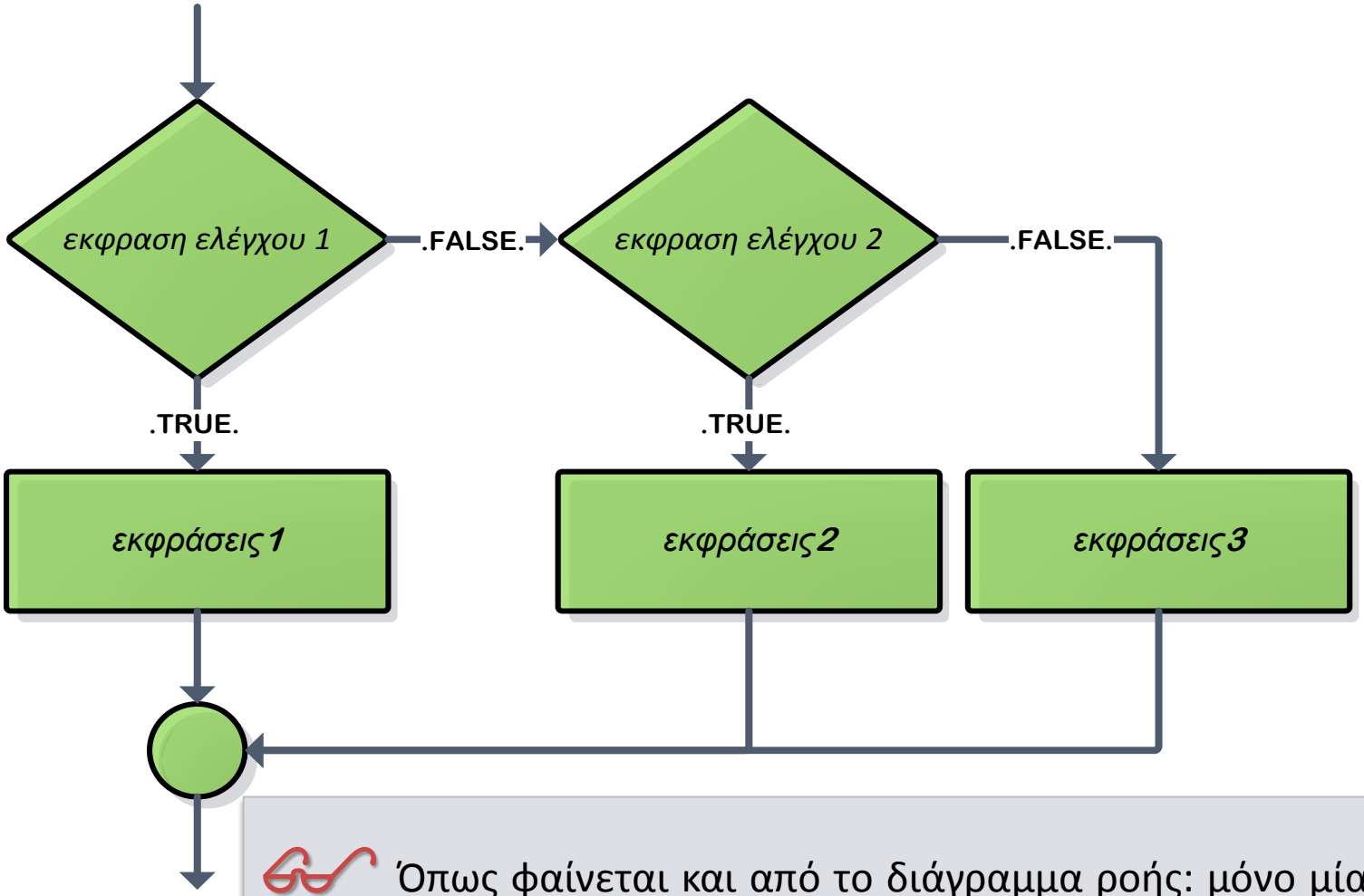
Στη δομή **IF**…**END IF**:

- Υποχρεωτικά έχουμε μόνο την αρχική (**IF**(*έκφραση ελέγχου*)**THEN**) και την τελική πρόταση (**END IF**)
- Μπορούμε να έχουμε από κανένα έως όσα **ELSE IF** θέλουμε
- Μπορούμε, αν θέλουμε, να έχουμε **ELSE**, αλλά μόνο ένα
- Μπορούμε να δώσουμε όνομα στην δομή για να την ξεχωρίζουμε από άλλες δομές
- Εκτελείται μόνο μία από τις εκτάσεις: ***η πρώτη έκταση της οποίας η έκφραση ελέγχου είναι αληθής***
- Ο προγραμματιστής πρέπει να διασφαλίσει την αμοιβαία αποκλειστικότητα των εκφράσεων ελέγχου

Στο λογικό **IF**:

- Μπορούμε να έχουμε μία μοναδική έκφραση ή πρόταση

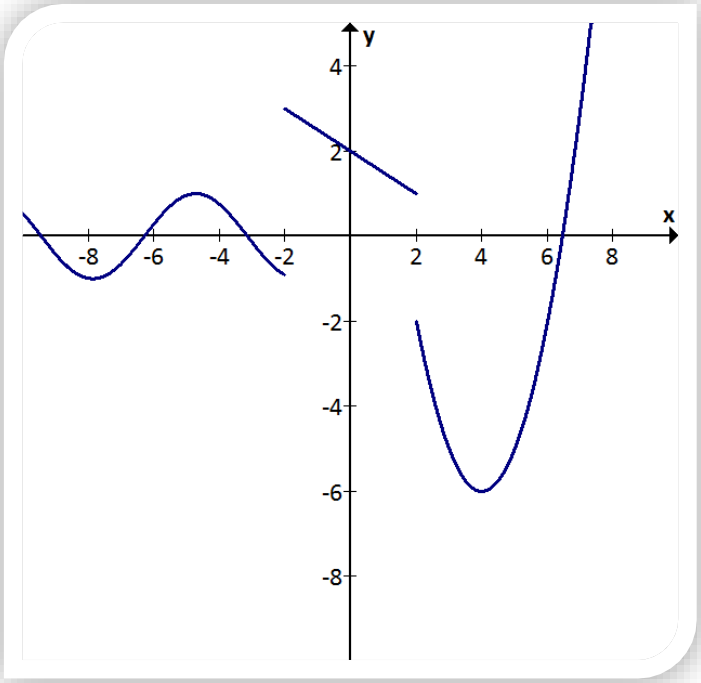
Δομή **IF** με ένα **ELSE IF** και με **ELSE**



Όπως φαίνεται και από το διάγραμμα ροής: μόνο μία από τις εκτάσεις (*εκφράσεις*) θα εκτελεστεί και το πρόγραμμα θα συνεχίσει κανονικά μετά την δομή, αγνοώντας τις υπόλοιπες εκτάσεις (*εκφράσεις*)

Δομή **IF..END IF** παραδείγματος χάριν

```
IF (x<-2)          THEN     !.............αρχική πρόταση
  y = sin(x)                ! 1η έκταση
ELSE IF( x>= 2) THEN
  y = x**2 - 8 * x + 10     ! 2η έκταση
ELSE  !x>=-2.and.x<2
  y=2-x/2                   ! 3η έκταση
END IF                      !.............τελική πρόταση
```

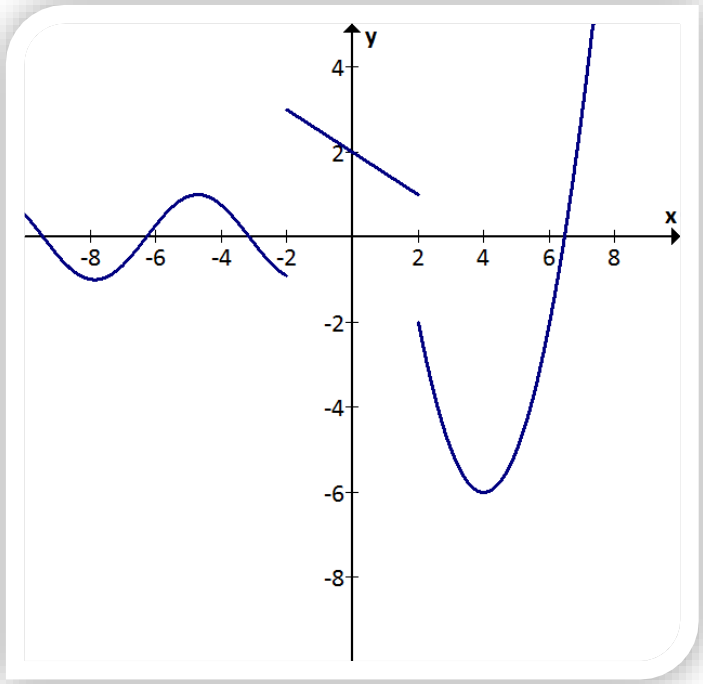


Δομή **IF..END IF** με όνομα

```
[όνομα δομής:]IF(έκφραση ελέγχου1)THEN   !αρχική πρόταση
      ....
      εκφράσεις1                 ! 1η έκταση
      ....
[ELSE IF(έκφραση ελέγχου2)THEN[όνομα δομής]]
      ....
      [εκφράσεις2]               ! 2η έκταση
      ....
[ELSE[όνομα δομής]]
      ....
      [εκφράσεις3]               ! 3η έκταση
      ....
END IF[όνομα δομής]                      !τελική πρόταση
```

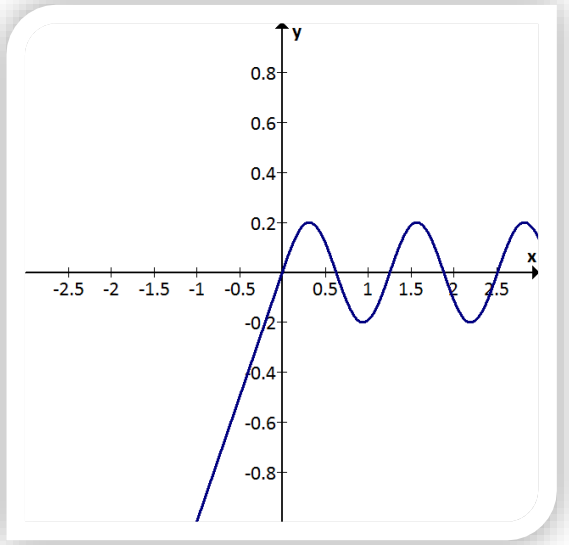
Δομή **IF..END IF** παραδείγματος χάριν με όνομα

```
first:IF  (x < -2)  THEN  !.............αρχική πρόταση
  y = sin(x)                ! 1η έκταση
ELSE IF(x >= 2)     THEN first
  y = x**2 - 8 * x + 10     ! 2η έκταση
ELSE  first   !x>=-2.and.x<2
  y=2-x/2                   ! 3η έκταση
END IF  first               !.............τελική πρόταση
```



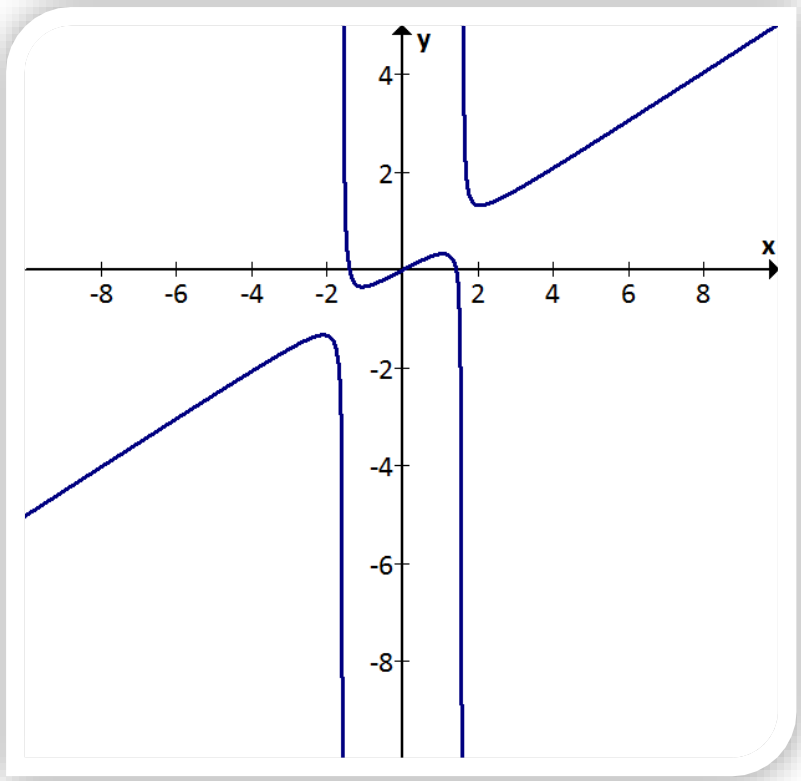
Δομή **IF..END IF** παραδείγματος χάριν

```
IF (x >= 0)        THEN        !...........αρχική πρόταση
  y = 1. / 5 * sin(5 * x)      ! 1η έκταση
ELSE  !x<0
  y = x                        ! 2η έκταση
END IF                         !...........τελική πρόταση
```



Δομή **IF..END IF** παραδείγματος χάριν

```
IF (x**2/=5)  THEN                    !αρχική πρόταση
   y = (x**3 - 2 * x) / 2 * (x**2 - 5)   ! 1η έκταση
END IF                                !τελική πρόταση
```



Λύση του τριωνύμου


```
  implicit none

! δηλώσεις:
  real   ::a, b, c, d
  real   ::root, root1, root2, rootreal, rootimag

! αρχή:
  call execute_command_line('chcp 1253')
  ! η παραπάνω πρόταση επιτρέπει την εμφάνιση ελληνικών

  print *, 'ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ'
  read  *, a, b, c          ! εισαγωγή συντελεστών
  d = b**2 - 4. * a * c     ! διακρίνουσα

  if(a == 0)  then

    print *, 'Η ΕΞΙΣΩΣΗ ΔΕΝ ΕΙΝΑΙ ΔΕΥΤΕΡΟΒΑΘΜΙΑ'
    root = -c / b; print *, root
```

Λύση του τριωνύμου


```
 else if   (d >= 0)      then

  print *,'Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΠΡΑΓΜΑΤΙΚΕΣ ΡΙΖΕΣ'
  root1 = (-b + sqrt(d))/(2. * a)
  root2 = (-b - sqrt(d))/(2. * a)
  print *, root1, root2

 else

  print *, 'Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ'
  rootreal = (-b) / (2. * a)
  rootimag = sqrt(abs(d)) / (2. * a)
  print *, rootreal, '+', rootimag, 'i'
  print *, rootreal, '-', rootimag, 'i'

 end if

end
```

Στιγμιότυπα εκτέλεσης του προγράμματος:


```
 ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ
1 2 3
 Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ
  -1.000000     +   1.414214     i
  -1.000000     -   1.414214     i
```

```
 ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ
1 3 1
 Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΠΡΑΓΜΑΤΙΚΕΣ ΡΙΖΕΣ
 -0.3819660      -2.618034
```

```
 ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ
1 5 -5
 Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΠΡΑΓΜΑΤΙΚΕΣ ΡΙΖΕΣ
  0.8541019      -5.854102
```

```
ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ
0 1 3
 Η ΕΞΙΣΩΣΗ ΔΕΝ ΕΙΝΑΙ ΔΕΥΤΕΡΟΒΑΘΜΙΑ
  -3.000000
```

Δομή SELECT CASE...END SELECT

```
[όνομα δομής:]SELECT CASE(έκφραση ελέγχου1)  !αρχική πρόταση

   CASE(λίστα1)[όνομα δομής]      !δευτερεύουσα πρόταση
      ....
      εκφράσεις1                  ! 1η έκταση
      ....

   [CASE(λίστα2)[όνομα δομής]]    !δευτερεύουσα πρόταση
      ....
      [εκφράσεις2]                ! 2η έκταση
      ....

   [CASE DEFAULT[όνομα δομής]]    !δευτερεύουσα πρόταση
      ....
      [εκφράσεις3]                ! 3η έκταση
      ....

END SELECT [όνομα δομής]                    !τελική πρόταση
```

Δομή **SELECT CASE...END SELECT**

Στη δομή **SELECT CASE...END SELECT**:

- Υποχρεωτικά έχουμε μόνο την αρχική (**SELECT CASE** (*έκφραση ελέγχου*)), την τελική πρόταση (**END SELECT**) και τουλάχιστον μία **CASE**
- Μπορούμε να έχουμε όσα **CASE** θέλουμε
- Μπορούμε, αν θέλουμε, να έχουμε ένα μόνο **CASE DEFAULT**
- Μπορούμε να δώσουμε όνομα στην δομή για να την ξεχωρίζουμε από άλλες δομές
- Η έκφραση ελέγχου έχει αποτέλεσμα: ***ακέραιο***, ***αλφαριθμητικό*** ή ***λογικό***
- Κάθε λίστα μπορεί να περιλαμβάνει διακριτές τιμές και διαστήματα τιμών ίδιου τύπου με την έκφραση ελέγχου
  - Πχ: `(:0, 2:7, 9:11, 13, 15, 20:)` ή
    `('a', 'c', 'e':'h', 'l':)`
- Εκτελείται μόνο μία από τις εκτάσεις: ***η πρώτη της οποίας η λίστα περιλαμβάνει την τιμή της έκφρασης ελέγχου***
- Ο μεταγλωττιστής δεν επιτρέπει να περιλαμβάνεται η ίδια τιμή σε δύο **CASE** (διασφαλίζει την αμοιβαία αποκλειστικότητα)

Απλή αριθμομηχανή


```
 implicit none

! δηλώσεις:
   real       ::i, j, k
   character  ::praxis
   logical    ::flag = .true.

!αρχή:
   call execute_command_line('chcp 1253')
   ! η παραπάνω πρόταση επιτρέπει την εμφάνιση ελληνικών

   print *, 'Δώσε δύο μη μηδενικούς αριθμούς'
   read  *, i, j
   print *,'Δώσε ένα τελεστή'
   read  *, praxis
```

Απλή αριθμομηχανή


```
 select case(praxis)
    case ('+')
     k = i + j
    case ('-')
     k = i - j
    case ('*')
     k = i * j
    case ('/')
     k = i / j
    case default !οποιοσδήποτε άλλος χαρακτήρας
     print *,'Λάθος Τελεστής'; flag = .false.
 end select
     if(flag) print *, 'το αποτέλεσμα είναι:',k
end
```

```
Δώσε δύο μη μηδενικούς αριθμούς
2 3
 Δώσε ένα τελεστή
+
 Το Αποτέλεσμα Είναι:   5.000000
```

Εμφώλευση δομών **IF…END IF**

```
[όνομα δομής Α:]IF(έκφραση ελέγχου1)THEN
        εκφράσεις Α1
    [όνομα δομής Β:]IF(έκφραση ελέγχου Β1)THEN
        [εκφράσεις Β1]
    [ELSE IF(έκφραση ελέγχου Β2)THEN[όνομα δομής Β]]
        [εκφράσεις Β2]
    [ELSE[όνομα δομής Β]]
        [εκφράσεις Β3]
     END IF [όνομα δομής Β]
[ELSE IF(έκφραση ελέγχου Α2)THEN[όνομα δομής Α]]
        [εκφράσεις Α2]
[ELSE[όνομα δομής Α]]
        [εκφράσεις Α3]
END IF [όνομα δομής Α]
```

Η εμφωλευμένη δομή πρέπει να <u>*εμπεριέχεται πλήρως*</u> (δηλαδή από την αρχική μέχρι και την τελική πρόταση) <u>*σε μία από τις εκτάσεις*</u> της δομής που την περιβάλλει.

Λύση του τριωνύμου με εμφώλευση δομών


```
   implicit none

!  δηλώσεις:
   real ::a, b, c, d
   real ::root, root1, root2, rootreal, rootimag

!  αρχή:
   call execute_command_line('chcp 1253')
   print *, 'ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ'
   read *,a, b, c            ! εισαγωγή συντελεστών
   d = b**2 - 4. * a * c     ! διακρίνουσα

   if(a == 0) then
     print *, 'Η ΕΞΙΣΩΣΗ ΔΕΝ ΕΙΝΑΙ ΔΕΥΤΕΡΟΒΑΘΜΙΑ'
     root = -c / b; print *, root
   else
```

Λύση του τριωνύμου με εμφώλευση δομών


```
else

   if (d >= 0)   then
      print *,'Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΠΡΑΓΜΑΤΙΚΕΣ ΡΙΖΕΣ'
      root1 = (-b + sqrt(d))/(2. * a)
      root2 = (-b - sqrt(d))/(2. * a)
      print *, root1, root2
   else
      print *,'Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ'
      rootreal =(-b) / (2. * a)
      rootimag = sqrt(abs(d))/(2. * a)
      print *, rootreal, '+', rootimag, 'i'
      print *, rootreal, '-', rootimag, 'i'
   end if

end if

end
```

Λύση του τριωνύμου με εμφώλευση δομών το ίδιο με ονόματα 1

```
   implicit none

!  δηλώσεις:
   real ::a, b, c, d
   real ::root, root1, root2, rootreal, rootimag

!  αρχή:
   call execute_command_line('chcp 1253')
   print *, 'ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ'
   read  *, a, b, c          ! εισαγωγή συντελεστών
   d = b**2 - 4. * a * c     ! διακρίνουσα

   first:if(a == 0)   then
    print *, 'Η ΕΞΙΣΩΣΗ ΔΕΝ ΕΙΝΑΙ ΔΕΥΤΕΡΟΒΑΘΜΙΑ'
    root = -c / b; print *, root

   else first
```

Λύση του τριωνύμου με εμφώλευση δομών το ίδιο με ονόματα 2

```
    else first

        second:if  (d >= 0)   then
            print *, 'Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΠΡΑΓΜΑΤΙΚΕΣ ΡΙΖΕΣ'
            root1 = (-b + sqrt(d))/(2. * a)
            root2 = (-b - sqrt(d))/(2. * a)
            print *, root1, root2

        else second
            print *, 'Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ'
            rootreal = (-b) / (2. * a)
            rootimag = sqrt(abs(d)) /( 2. * a)
            print *, rootreal, '+', rootimag, 'i'
            print *, rootreal, '-', rootimag, 'i'
        end if second

    end if first
end
```

Βαθμοί με **IF**


```
program ifgrades
   implicit none
!  δηλώσεις:
   real::grade

!  αρχή:
   call execute_command_line('chcp 1253')
   print*, 'δώσε το βαθμό(0-10)' ;read*,grade

   if (grade>10.) then
      print*, 'λάθος βαθμός'
   elseif (grade>=9.) then
      print*, 'βαθμός: A'
   elseif (grade>=8.) then
      print*, 'βαθμός: B'
   elseif (grade>=7.) then
      print*, 'βαθμός: C'
```

Βαθμοί με **IF**


```
   elseif (grade>=6.) then
      print*, 'βαθμός: D'
   elseif (grade>=5.) then
      print*, 'βαθμός: E'
   elseif (grade>=0.) then
      print*, 'βαθμός: F'
   else
      print*, 'λάθος βαθμός'
   endif

end program
```

Βαθμοί με **SELECT CASE**


```
program selectgrades
    implicit none
!   δηλώσεις:
    real::grade

!   αρχή:
    call execute_command_line('chcp 1253')
    print*, 'δώσε το βαθμό(0-10)' ;read*,grade

    !πως να χρησιμοποιήσω το select case με πραγματικούς
    select case (nint(grade*10))
        case(90:100)
            print*, 'βαθμός: A'
        case(80:89)
            print*, 'βαθμός: B'
        case(70:79)
            print*, 'βαθμός: C'
```

Βαθμοί με **SELECT CASE**


```
        case(60:69)
            print*, 'βαθμός: D'
        case(50:59)
            print*, 'βαθμός: E'
        case(0:49)
            print*, 'βαθμός: F'
        case default
            print*, 'λάθος βαθμός'
    end select

end program
```

Λύση του τριωνύμου με **complex**


```
  implicit none

! δηλώσεις:
  real      ::a, b, c, d
  real      ::root, root1, root2
  complex   ::z1, z2

! αρχή:
  call execute_command_line('chcp 1253')
  ! η παραπάνω πρόταση επιτρέπει την εμφάνιση ελληνικών

  print *, 'ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ'
  read  *, a, b, c          ! εισαγωγή συντελεστών
  d = b**2 - 4. * a * c     ! διακρίνουσα

  if(a == 0)  then

    print *, 'Η ΕΞΙΣΩΣΗ ΔΕΝ ΕΙΝΑΙ ΔΕΥΤΕΡΟΒΑΘΜΙΑ'
    root = -c / b; print *, root
```

Λύση του τριωνύμου με **complex**


```
 else if   (d >= 0)      then
  print *,'Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΠΡΑΓΜΑΤΙΚΕΣ ΡΙΖΕΣ'
  root1 = (-b + sqrt(d))/(2. * a)
  root2 = (-b - sqrt(d))/(2. * a)
  print *, root1, root2

 else
  print *, 'Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ'
  z1 = (-b + sqrt(cmplx(d))) / (2 * a)
  z2 = (-b - sqrt(cmplx(d))) / (2 * a)
  !Ή αλλιώς…
  !z1 = (cmplx(-b, + sqrt(abs(d)))) / (2 * a)
  !z2 = (cmplx(-b, - sqrt(abs(d)))) / (2 * a)
  print *, z1
  print *, z2

 end if
end
```

Στιγμιότυπα εκτέλεσης του προγράμματος:


```
 ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ
1 2 3
 Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ
 ( -1.00000000    ,  1.41421354    )
 ( -1.00000000    , -1.41421354    )
```

```
 ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ
1 0 1
 Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ
 (  0.00000000    ,  1.00000000    )
 ( -0.00000000    , -1.00000000    )
```

```
ΔΩΣΤΕ ΤΟΥΣ ΣΥΝΤΕΛΕΣΤΕΣ ΤΟΥ ΤΡΙΩΝΥΜΟΥ
3 8 21
 Η ΕΞΙΣΩΣΗ ΕΧΕΙ 2 ΜΙΓΑΔΙΚΕΣ ΡΙΖΕΣ
 ( -1.33333337    ,  2.28521824    )
 ( -1.33333337    , -2.28521824    )
```

Ο Πύραυλος

- Το μέγιστο ύψος h το οποίο μπορεί να φτάσει πύραυλος που εκτοξεύεται με ταχύτητα v είναι:

$$h = \frac{R_E}{1 - \frac{v^2}{2gR_E}}$$

Όπου η ακτίνα της γης $R_E = 6.366 \times 10^6\ m$

- Η ταχύτητα διαφυγής $v_e = \sqrt{2gR_E} = 1.117 \times 10^4\ m/s$
- Για ταχύτητες $v > v_e$ η τελική ταχύτητα θα είναι:

$$v_f = \sqrt{v^2 - 2gR_E}$$

Να φτιάξουμε πρόγραμμα που περιγράφει την πορεία του πυραύλου για ναι ταχύτητα που δίνεται από το χρήστη

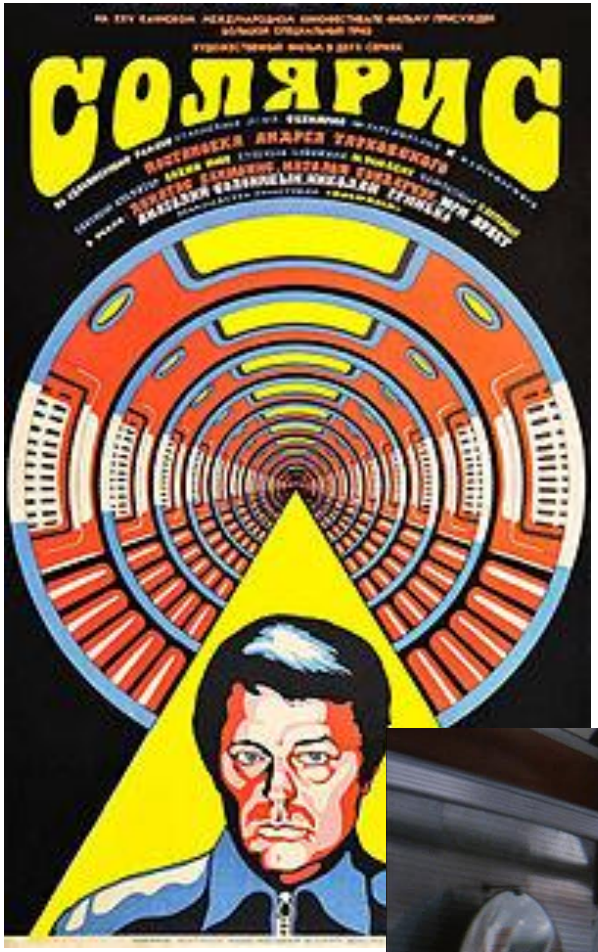
Ο Πύραυλος

βρείτε τα 2 λάθη 2

```
   implicit none
!  δηλώσεις:
   real             :: v, h, vf
   real, parameter  :: g = 9.8, R = 6.366e6
!  αρχή:
   call execute_command_line('chcp 1253 ')
   print *, 'δώσε την αρχική ταχύτητα σε m/s'   ;read *, v
   if (v**2 < 2 * g * R) then
      h = R / (1 - v ** 2 / (2 * g * R))
      print *, "Ο πύραυλος φτάνει σε ύψος:", h - R, "m"
      print *, "από την επιφάνεια της Γης " // &
               "πριν επιστρέψει στη Γη."
   elseif (v**2 == 2 * g * R) then
      print *, "Ο πύραυλος μόλις διαφεύγει" // &
               "από το βαρυτικό πεδίο της Γης."
   else
      vf = sqrt(v**2 - 2 * g * R)
      print *, "Ο πύραυλος διαφεύγει με ταχύτητα:", vf, "m/s."
   end if
end
```

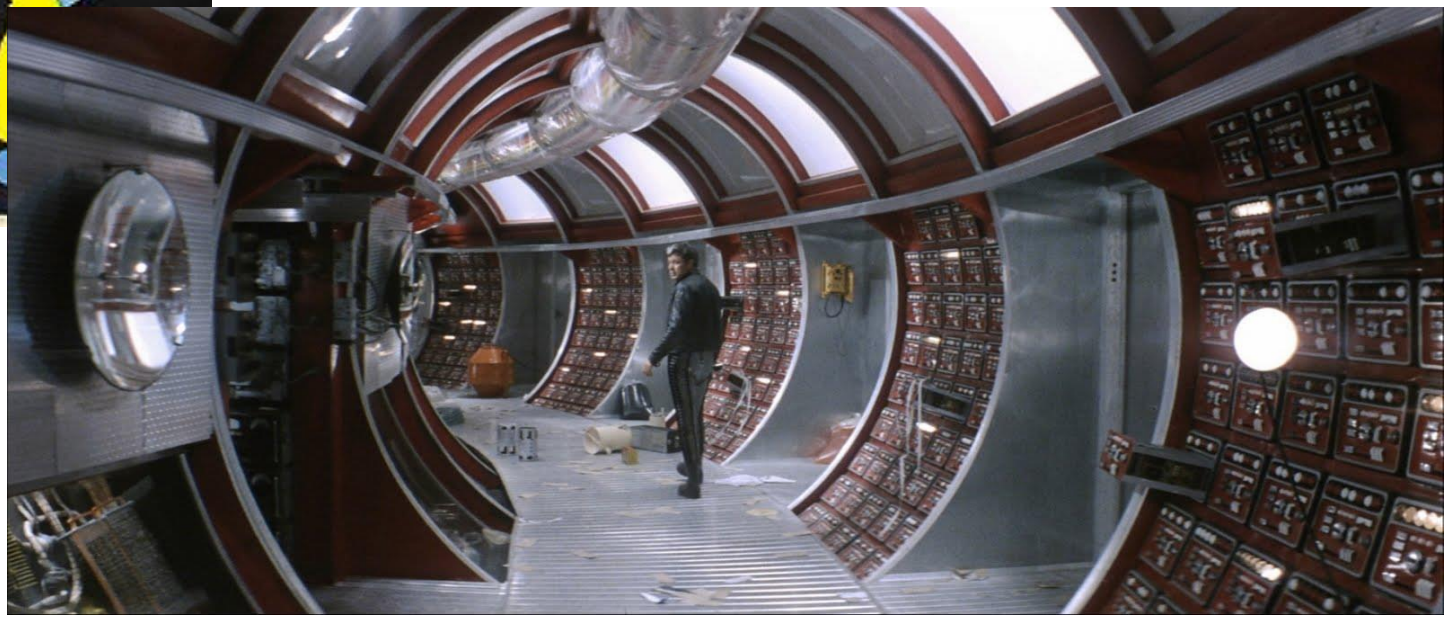
Ο Πύραυλος σωστά!


```
   implicit none
!  δηλώσεις:
   real              :: v, h, vf
   real, parameter   :: g = 9.8, R = 6.366e6
!  αρχή:
   call execute_command_line('chcp 1253 ')
   print *, 'δώσε την αρχική ταχύτητα σε m/s'   ;read *, v
   if (v**2 - 2 * g * R < 1e-6) then
      print *, "Ο πύραυλος μόλις διαφεύγει" // &
               "από το βαρυτικό πεδίο της Γης."
   else if (v**2 < 2 * g * R) then
      h = R / (1 - v ** 2 / (2 * g * R))
      print *, "Ο πύραυλος φτάνει σε ύψος:", h - R, "m"
      print *, "από την επιφάνεια της Γης " // &
               "πριν επιστρέψει στη Γη."
   else
      vf = sqrt(v**2 - 2 * g * R)
      print *, "Ο πύραυλος διαφεύγει με ταχύτητα:", vf, "m/s."
   end if
end program missile
```



SOLARIS (Βραβείο Φεστιβάλ Καννών 1972) Α. Ταρκόφσκι

Η αδυναμία του ανθρώπινου είδους να επικοινωνήσει με άλλα είδη. Μια από τις σπουδαιότερες ταινίες επιστημονικής φαντασίας στην ιστορία του κινηματογράφου, βασισμένη στο βιβλίο του Σ. Λεμ



Σημείωμα Αναφοράς

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 την 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.

Σημείωμα Αδειοδότησης