

ΑΣΚΗΣΕΙΣ

1. Χωρίς να γράψετε το πρόγραμμα βρείτε τι θα τυπώσουν τα ακόλουθα προγράμματα (Απαντήστε στο χώρο που σας δίνεται). Θα πρέπει να γράψετε τους αριθμούς όπως θα τυπωθούν από τον υπολογιστή. (5β + 5β)

(A)

```
C=====
      program test1
C=====
      integer i,j
      real    x,y
      real    c,d
      F(x,y)=L*x+K*y
      L=5
      K=2
      x=1.5
      y=2.0
      c=F(5.,3.)
      d=F(L,K)
      write(6,10)c,d
10  format(1x,'c =',1x,f4.1,/,
&1x,'d= ',1x,f4.1)
      End
```

Σύμφωνα με τον ορισμό της συνάρτησης εντός, την πρώτη φορά που καλείται ($c = F(5, 3)$) τα x , και y θα πάρουν τις τιμές $x=5$ και $y=3$ και το πρόγραμμα θα υπολογίσει $c = L*5 + K*3 = 5*5 + 2*3 = 31$

Τη δεύτερη φορά το ρόλο του x και y παίρνει το K και L

Δηλαδή: $d = F(5, 2)$ οπότε :

$$d = L*5 + K*2 = 5*5 + 2*2 = \underline{\underline{29}}$$

(B)

```
C=====
      program test3
C=====
      INTEGER J
      REAL A, B, D, C
      REAL EXCH
C
      A = 10
      B = 5
      C = EXCH(A,B)
      DO J = 1, 2
         D = C + (B - A)*J
      ENDDO
      WRITE(6,10)D
10  FORMAT(1x,'D =',1x,F5.1)
      END
C=====
      REAL FUNCTION EXCH(B,A)
C=====
      REAL A,B,T
      EXCH = -5.0
      T = B = 10
      B = A = 5
      A = T = 10
      RETURN
      END
```

Οι περισσότεροι κάνουν λάθος στην άσκηση αυτή.

Οι παράμετροι μιας συνάρτησης ή υπορουτίνας έχουν συγκεκριμένη θέση στον ορισμό της συνάρτησης ή υποπρογράμματος

Το συμβολικό όνομα που χρησιμοποιούμε δεν έχει σημασία

Όπως εδώ, στο κύριο πρόγραμμα λέμε $EXCH(A,B)$ ενώ

στην function (B,A) ουσιαστικά η αντιστοιχία είναι στη θέση στη σειρά των παραμέτρων.

Έτσι όταν καλείται η συνάρτηση και η εκτέλεση του προγράμματος μεταφέρεται στην συνάρτηση $EXCH$

το $B=10$ και το $A=5$. Οπότε μέσα στη

συνάρτηση οι τιμές των A και B εναλλάσσονται.

Οπότε γυρνώντας στο κύριο πρόγραμμα το

B έχει γίνει 10 και το $A=5$.

2. Χωρίς να χρησιμοποιήσετε τον υπολογιστή βρείτε τα λάθη στα ακόλουθα προγράμματα και διορθώστε τα ανάλογα. Γράψτε τι θα τυπώσουν τα προγράμματα σε κάθε περίπτωση (5β + 5β).

(A)

```

C=====
program test5
C=====
INTEGER J, N
REAL A(2), B, Z D
REAL SUMSQ
C
Y(A(I), B) = A(I)**2 + B**2
C
DO J = 1, 2
  A(J) = J+1
ENDDO
C
B = 4
DO J = 1, 2
  Z(J) = Y(A(I), B)
ENDDO
D = SUMSQ(A(2), B)
C
WRITE(6,10) Z, D
10 FORMAT(1x, 'D = ', 1x, F5.1, /,
& 1x, 'Z = ', 1x, F5.1)
END
C=====
REAL FUNCTION SUMSQ(X, Y)
C=====
REAL A, X(2), Y, Z(2)
INTEGER J
C
A = 11
DO J = 1, 2
  Z(J) = A + X(J)**2 + Y**2
ENDDO
SUMSQ = Z(J)
RETURN
END

```

Θα μπορούσα να πω
ότι το Z δεν είναι πίνακας
και τότε θα δίνει
αλλά $Z = 11 + X(2) * X(2) + Y^2$

Προφανώς η διαγ και η τετραγωνική
είναι λίγο αμφιλεγόμενες για το
τι κάνει η subroutine για να
επιστρέψουμε να διορθώσαμε
το αποτέλεσμα.

Θα μπορούσε να είναι:

$SUMSQ = Z(2)$ ή

$SUMSQ = Z(1) + Z(2)$ ή

Θα μπορούσαμε να βάλουμε την
ενσωλή μέσα στο LOOP οπότε
 $SUMSQ = SUMSQ + Z(3)$ και
πριν το loop $SUMSQ = 0$.

(B)

```

C=====
program test6
C=====
INTEGER J, K, N, XK, XM
INTEGER A(3,3), B(3,3)
REAL C(3,3), X, TEST
C
N = 3
DO 10 J = 1, N
  DO 20 K = 1, N
    A(J,K) = J*K
    B(J,K) = 2.0*A(J,K)
    C(J,K) = 1.0
  CONTINUE
CONTINUE
C
DO 30 J = 1, N
  DO 40 K = 1, N
    X = J*K
    XM = A(J,K)
    XK = B(J,K)
    C(J,K) = C(J,K) - XM/XK*X
  CONTINUE
CONTINUE
TEST = DIAGONAL(C, N)
WRITE(6,50) TEST
50 FORMAT(1x, 'To athroisma C = ',
& 1x, f6.3)
END
C=====
REAL FUNCTION DIAGONAL(C, N)
C=====
INTEGER N, J, K
REAL C(N,N), SUM
C
SUM = 0
DO J = 1, N
  SUM = SUM + C(J, J)
ENDDO
DIAGONAL = SUM
RETURN
END

```

(10)
(20)

(30)
(40)

(50)

(60)

(70)

(80)

(90)

(100)

(110)

(120)

(130)

(140)

(150)

(160)

(170)

(180)

(190)

(200)

(210)

(220)

(230)

(240)

(250)

(260)

(270)

(280)

(290)

(300)

(310)

(320)

(330)

(340)

(350)

(360)

(370)

(380)

(390)

(400)

(410)

(420)

(430)

(440)

(450)

(460)

(470)

(480)

(490)

(500)

(510)

(520)

(530)

(540)

(550)

(560)

(570)

(580)

(590)

(600)

(610)

(620)

(630)

(640)

(650)

(660)

(670)

(680)

(690)

(700)

(710)

(720)

(730)

(740)

(750)

(760)

(770)

(780)

(790)

(800)

(810)

(820)

(830)

(840)

(850)

(860)

(870)

(880)

(890)

(900)

(910)

(920)

(930)

(940)

(950)

(960)

(970)

(980)

(990)

(1000)

(1010)

(1020)

(1030)

(1040)

(1050)

(1060)

(1070)

(1080)

(1090)

(1100)

(1110)

(1120)

(1130)

(1140)

(1150)

(1160)

(1170)

(1180)

(1190)

(1200)

(1210)

(1220)

(1230)

(1240)

(1250)

(1260)

(1270)

(1280)

(1290)

(1300)

(1310)

(1320)

(1330)

(1340)

(1350)

(1360)

(1370)

(1380)

(1390)

(1400)

(1410)

(1420)

(1430)

(1440)

(1450)

(1460)

(1470)

(1480)

(1490)

(1500)

(1510)

(1520)

(1530)

(1540)

(1550)

(1560)

(1570)

(1580)

(1590)

(1600)

(1610)

(1620)

(1630)

(1640)

(1650)

(1660)

(1670)

(1680)

(1690)

(1700)

(1710)

(1720)

(1730)

(1740)

(1750)

(1760)

(1770)

(1780)

(1790)

(1800)

(1810)

(1820)

(1830)

(1840)

(1850)

(1860)

(1870)

(1880)

(1890)

(1900)

(1910)

(1920)

(1930)

(1940)

(1950)

(1960)

(1970)

(1980)

(1990)

(2000)

(2010)

(2020)

(2030)

(2040)

(2050)

(2060)

(2070)

(2080)

(2090)

(2100)

(2110)

(2120)

(2130)

(2140)

(2150)

(2160)

(2170)

(2180)

(2190)

(2200)

(2210)

(2220)

(2230)

(2240)

(2250)

(2260)

(2270)

(2280)

(2290)

(2300)

(2310)

(2320)

(2330)

(2340)

(2350)

(2360)

(2370)

(2380)

(2390)

(2400)

(2410)

(2420)

(2430)

(2440)

(2450)

(2460)

(2470)

(2480)

(2490)

(2500)

(2510)

(2520)

(2530)

(2540)

(2550)

(2560)

(2570)

(2580)

(2590)

(2600)

(2610)

(2620)

(2630)

(2640)

(

ΑΣΚΗΣΕΙΣ

1. Χωρίς να πληκτρολογήσετε το κώδικα βρείτε τι θα τυπώσει το ακόλουθο πρόγραμμα. Θα πρέπει να γράψετε τους αριθμούς όπως θα τυπωθούν από τον υπολογιστή. (5β + 5β).

(A)

```
C=====
      program test2
C=====
      REAL    x,y
      REAL    c,d
      F(x,y)=K/L*(x**2+(L-K)*y)
      L=12
      K=6
      X=4.0
      Y=3.0
      C=F(5.,3.)
      D=F(K,L)
      WRITE(6,10)C,D
10  FORMAT(1x,'c =',1x,f4.1,/,1x,'d= ',1x,f4.1)
      End
```

Από τη συνάρτηση έχουμε ότι: $\frac{K}{L} = \frac{6}{12} = 0$ λόγω
διαίρεσης ονείρων και επομένως το αποτέλεσμα
θα είναι πάντοτε 0 για τα δεδομένα $K \neq L$
Έτσι $C=0$ & $D=0$

(B)

```
C=====
      program test4
C=====
      INTEGER J
      REAL A, B, Z, D
C
      D = 0
      Z = 3
      A = 10
      B = 5
      CALL ROTATE(A,B,Z)
      DO J = 1, 2
         D = D + (B - A)*J
         Z = Z + D
      ENDDO
      WRITE(6,10)D, Z
10  FORMAT(1x,'D =',1x,F5.1,/,
      &      1x,'Z =',1x,F5.1)
      END
C=====
      SUBROUTINE ROTATE(Z,A,B)
C=====
      REAL A,B,Z,T
      Z = A - B
      T = B
      B = A
      A = T
      RETURN
      END
```

Οι περισσότεροι ακολουθούν τα χράμματα A,B,Z στη
subroutine, αλλά είναι λάθος. Τα χράμματα είναι
μεταβλητές και θα μπορούσαν να είναι οτιδήποτε.
Η σειρά που δίνονται οι παράμετροι έχει σημασία.
Έτσι όταν καλείται η subroutine rotate στο το
κύριο πρόγραμμα αυτό που καλείται είναι:

ROTATE(10,5,3)

Έτσι και μέσα στη subroutine οι τοπικές μεταβλητές
Z,A,B θα είναι:

$Z=10, A=5, B=3.$

Στην α' εντολή $Z = 5-3 = 2$

$T = 3$

$B = 5$

$A = 3$

Όταν επιστρέφει στο κύριο πρόγραμμα τα A,B,Z
έχουν τιμές:

$A=2$ αντιστοιχεί στο Z

$B=3$ " στο A

$Z=5$ " στο B

Μετά τις πράξεις

$D = 3$

$Z = 9$

2. Χωρίς να χρησιμοποιήσετε τον υπολογιστή να διορθωθούν τα λάθη στα ακόλουθα προγράμματα και να βρεθεί το τι θα εκτυπώσει ο υπολογιστής (5β + 5β):

(A)

```

C=====
program test7
C=====
  INTEGER K, J, N, A, B
  INTEGER X(10), Y(10)
  INTEGER Z(10)
  INTEGER SUM, SUMB
  INTEGER ADD
  C
  N = 5
  DO J = 1, N
    K = J + 1
    X(K) = (-1.) * J * (N/J)
    Y(J) = 2J * (K-J)
  ENDDO
  CALL SUMA(X(10), Y(10), N, SUM)
  SUMA = ADD(X, Y, TES)
  WRITE(5,100) SUM, SUMB
  FORMAT(1x, 'H subroutine dinei: ', F5.1, ' ', F5.1, ' ', F5.1)
  & // 1x, 'H function dinei: ', F5.1, ' ', F5.1, ' ', F5.1
  END
  SUBROUTINE SUMA(XX, YY, N, RESULT)
  C=====
  INTEGER XX(N), YY(N), RESULT
  INTEGER N, J
  C
  SUMA = 0.
  DO J = 1, N
    SUMA = SUMA + XX(J) + YY(J)
  ENDDO
  RESULT = SUMA
  RETURN
  END
  REAL FUNCTION ADD(XX, YY, N, TES)
  C=====
  INTEGER N, XX(N), YY(N)
  REAL TES
  TES = 0.0
  DO J = 1, N
    TES = TES + XX(J) + YY(J)
  ENDDO
  RETURN
  END

```

(B)

```

C=====
program test8
C=====
  INTEGER J, K, L, X(3)
  REAL A(3), D(3), C
  C
  L = 2
  DO J = 1, 3
    X(J) = F(3,1) + cos(3)**2
    A(J) = X(J)/J
  CONTINUE
  J = 0
  J = J + 1
  CALL DIVIDE(A(J), J, K)
  IF (K = 0) THEN
    PRINT *, 'Pol/sio tou', J
    GOTO 2
  ELSE
    PRINT *, 'List is over'
  ENDIF
  RETURN
  END
  SUBROUTINE DIVIDE(A, J, K)
  C=====
  INTEGER J, K, L
  REAL A
  C
  L = A/J
  K = A - L*J
  RETURN
  END

```

Handwritten notes and corrections:

- Program test7:**
 - `INTEGER X(10), Y(10)`: *Δίνει παρένθεση*
 - `INTEGER ADD`: *Ορίζεται REAL*
 - `CALL SUMA(X(10), Y(10), N, SUM)`: *Αν K=J+1 τότε X(K) δεν ορίζεται και θα είναι τρέχον. Μπορούμε X(3)*
 - `SUMA = ADD(X, Y, TES)`: *χρησιμοποιεί όλο τον πίνακα*
 - `WRITE(5,100) SUM, SUMB`: *Λείπει το N (έχει 4 παραμέτρους)*
 - `FORMAT(1x, 'H subroutine dinei: ', F5.1, ' ', F5.1, ' ', F5.1)`: *Λάθος FORMAT*
 - `END`: *πληκτρολόγησε*
 - `SUBROUTINE SUMA(XX, YY, N, RESULT)`: *Αυτό είναι ok αλλά N ≤ 5*
 - `SUMA = 0.`: *όχι 0*
 - `DO J = 1, N`: *όχι 0*
 - `SUMA = SUMA + XX(J) + YY(J)`: *χρησιμοποιεί κα*
 - `RESULT = SUMA`: *Function*
 - `RETURN`: *ILLEGAL*
 - `END`: *END*
 - `REAL FUNCTION ADD(XX, YY, N, TES)`: *ILLEGAL*
 - `INTEGER N, XX(N), YY(N)`: *ILLEGAL*
 - `REAL TES`: *ILLEGAL*
 - `TES = 0.0`: *ILLEGAL*
 - `DO J = 1, N`: *ILLEGAL*
 - `TES = TES + XX(J) + YY(J)`: *ILLEGAL*
 - `ENDDO`: *ILLEGAL*
 - `RETURN`: *ILLEGAL*
 - `END`: *ILLEGAL*
- Program test8:**
 - `INTEGER J, K, L, X(3)`: *Δίνει παρένθεση*
 - `REAL A(3), D(3), C`: *Δίνει παρένθεση*
 - `L = 2`: *ορίζεται L*
 - `DO J = 1, 3`: *ορίζεται J*
 - `X(J) = F(3,1) + cos(3)**2`: *ορίζεται X*
 - `A(J) = X(J)/J`: *ορίζεται A*
 - `CONTINUE`: *ορίζεται C*
 - `J = 0`: *ορίζεται J*
 - `J = J + 1`: *ορίζεται J*
 - `CALL DIVIDE(A(J), J, K)`: *ορίζεται K*
 - `IF (K = 0) THEN`: *ορίζεται K*
 - `PRINT *, 'Pol/sio tou', J`: *ορίζεται J*
 - `GOTO 2`: *ορίζεται J*
 - `ELSE`: *ορίζεται J*
 - `PRINT *, 'List is over'`: *ορίζεται J*
 - `ENDIF`: *ορίζεται J*
 - `RETURN`: *ορίζεται J*
 - `END`: *ορίζεται J*
 - `SUBROUTINE DIVIDE(A, J, K)`: *ορίζεται J*
 - `INTEGER J, K, L`: *ορίζεται J*
 - `REAL A`: *ορίζεται J*
 - `L = A/J`: *ορίζεται J*
 - `K = A - L*J`: *ορίζεται J*
 - `RETURN`: *ορίζεται J*
 - `END`: *ορίζεται J*