

ΦΥΣ 145 – Μαθηματικές Μέθοδοι στη Φυσική

Πρόοδος

13 Μαρτίου 2006

Ομάδα 2^η

Γράψτε το ονοματεπώνυμο και αριθμό ταυτότητάς σας στο πάνω μέρος της αυτής της σελίδας.

Πρέπει να απαντήσετε και στα 6 προβλήματα που σας δίνονται.

Ο χρόνος εξέτασης είναι 90 λεπτά.

Από τη στιγμή αυτή δεν υπάρχει συνεργασία/συζήτηση ανταλλαγή αρχείων και e-mails με κανένα και φυσικά κουδούνισμα κινητού που πρέπει να κλείσουν. Σημειώσεις, χαρτάκια κλπ απαγορεύονται όπως και επισκέψεις σε ιστοσελίδες που δεν αναφέρονται στην ιστοσελίδα του μαθήματος.

Καλή επιτυχία

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
```

(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
```

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
C
      CALL SUMA(X(10), Y(10), N, SUM)
      SUMA = ADD(X, Y, TES)
C
      WRITE(5,100) SUM, SUMB
      FORMAT(1x, 'H subroutine dinei: ', F5.1,
&          /, 1x, 'H function dinei: ', F5.1)
      END
C=====
      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
C=====
      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
      F(X,K) = LX + K(sinx)**2
C
      L = 2
      DO 20 J = 1, 3
        X(J) = F(3,1) + cos(3)**2
        A(J) = X(J)/J
      CONTINUE
      J = 0
      2 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
C=====
      SUBROUTINE DIVIDE (A, J, K)
C=====
      INTEGER J, K, L
      REAL A
C
      L = A/J
      K = A - L * J
      RETURN
      END

```

Ασκήσεις για τον υπολογιστή

Τις παρακάτω 3 ασκήσεις θα πρέπει να τις στείλετε με e-mail αφού πρώτα τις κάνετε tar στο phy145@ucy.ac.cy. Το e-mail σας θα πρέπει να έχει subject:midterm_groupB ενώ το attachment θα πρέπει να είναι στη μορφή <username>_groupB.tar όπου username ο e-mail account σας στο πανεπιστήμιο.

Μην ξεχάσετε να γράψετε το ονοματεπώνυμό σας και αριθμό ταυτότητας σε κάθε file που αντιστοιχεί στο πρόγραμμα που στέλνετε.

3. Για δύο θετικούς ακεραίους I και J με $I \leq J$, η γενικευμένη σειρά Fibonacci ορίζεται από τη σχέση: $JFIB(1) = I$, $JFIB(2) = J$, και $JFIB(N) = JFIB(N-2) + JFIB(N-1)$ για $N > 2$

Δηλαδή οι δύο πρώτοι όροι είναι ο I και J αντίστοιχα και κάθε όρος αργότερα προκύπτει από το άθροισμα των 2 προηγούμενων όρων.

Να γράψετε ένα πρόγραμμα το οποίο διαβάζει τα I και J από το πληκτρολόγιο, τυπώνει τους πρώτους 30 όρους της σειράς έτσι ώστε να υπάρχουν 3 αριθμοί σε κάθε γραμμή και να υπάρχουν 4 κενά μεταξύ τους και σταματά τους υπολογισμούς όταν η σειρά παίρνει τιμή η οποία ξεπερνά 10,000,000. Στο σημείο αυτό το πρόγραμμά σας θα πρέπει να τυπώσει τον όρο για τον οποίο η σειρά δίνει τιμή μικρότερη από 10,000,000.

Για παράδειγμα: Η σειρά Fibonacci για $I = J = 1$ είναι 1, 1, 2, 3, 5, 8, 13, 21,... **(10 β)**

4. Να γραφεί ένα πρόγραμμα το οποίο περιέχει μια συνάρτηση με το όνομα natural και η οποία δέχεται σα παράμετρο ένα REAL αριθμό x και υπολογίζει (και επιστρέφει στο κύριο πρόγραμμα) το όριο της ακόλουθης σειράς όταν το n πλησιάζει στο άπειρο:

$$S(x, n) = \sum_{n=1}^{\infty} \left(\frac{x}{n} \right)^n$$

Αρχίζοντας από $n=1$, ο αλγόριθμος υπολογίζει το $S(x, n)$ για αυξανόμενες τιμές του n μέχρι το σημείο που η απόλυτη τιμή της διαφοράς μεταξύ δύο διαδοχικών όρων γίνεται μικρότερη από κάποια τιμή epsilon. Η συνάρτηση θα πρέπει να έχει επίσης σαν παράμετρο την μεταβλητή epsilon που καθορίζει την ακρίβεια του υπολογισμού. Το κύριο πρόγραμμα θα πρέπει σας ζητά την ακρίβεια epsilon και να τυπώνει τις τιμές της σειράς για x στο διάστημα $x=1$ με $x=3$ το οποίο αυξάνει με βήμα 0.1. Η εκτύπωση θα πρέπει να γίνεται σε πίνακα με το x να εκτυπώνεται με ένα δεκαδικό ψηφίο και το αποτέλεσμα της σειράς με 6 δεκαδικά ψηφία **(10 β)**

5. Κάθε δεδομένο στο αρχείο data.txt (πρέπει να το κατεβάσετε από το <http://www.ucy.ac.cy/~phy145/exams/data.txt>) αποτελείται από ένα θετικό ακεραίο αριθμό. Το τελευταίο στοιχείο του αρχείου είναι -1 και χρησιμοποιείται για έλεγχο του αρχείου. Για διάφορους ορισμούς θεωρείστε ότι το αρχείο δεν περιέχει περισσότερους από 10000 αριθμούς. Θέλουμε να γράψουμε ένα πρόγραμμα το οποίο ελέγχει την κατανομή των αριθμών των ψηφίων σε αυτούς τους ακεραίους χωρίς να λαμβάνεται υπόψη το τελευταίο νούμερο. Δηλαδή χρειάζεται να μάθουμε πόσοι από αυτούς τους ακεραίους αποτελούνται από 1 ψηφίο, πόσοι από δύο, τρία, ... εννέα ψηφία. Για παράδειγμα αν στο αρχείο υπήρχαν οι αριθμοί 23, 45, 104, 222, 103230, τότε το πρόγραμμά μας θα έπρεπε να τυπώνει:

2-ψηφίων αριθμοί: 2

3-ψηφίων αριθμοί: 2

5-ψηφίων αριθμοί: 1

Παρατηρήστε ότι περιπτώσεις στις οποίες δεν έχουμε αριθμούς με συγκεκριμένο αριθμό ψηφίων δεν τυπώνονται (π.χ. αριθμοί με 4 ψηφία δεν υπάρχουν, όπως και αριθμοί με 1 ψηφίο). Το πρόγραμμά σας θα πρέπει να θεωρεί ότι οι ακέραιοι πρέπει να έχουν τουλάχιστον 1 ψηφίο και το πολύ 9 ψηφία και να τυπώνει το πλήθος των αριθμών με n-ψηφία σύμφωνα με το παραπάνω παράδειγμα (δε θα πρέπει να τυπώνονται οι περιπτώσεις που δεν έχουν βρεθεί ακέραιοι με κάποιο αριθμό ψηφίων). [Υπόδειξη: επαναλαμβανόμενη διαίρεση ενός ακεραίου με 10 εξάγει τον αριθμό των ψηφίων.] **(10 β)**