



Εισαγωγή στον Προγραμματισμό Η/Υ (Fortran 90/95/2003)

Ενότητα 6: Μονοδιάστατοι Πίνακες

Νίκος Καραμπετάκης
Τμήμα Μαθηματικών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

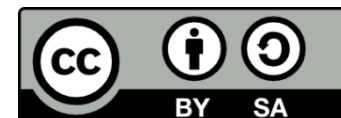


ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» 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.



Περιεχόμενα Ενότητας

1. Τι είναι πίνακας?
2. Είδη μεταβλητών στην Fortran 90/95/2003.
3. Συνήθης διαδικασία επίλυσης προβλημάτων με πίνακες.
4. Κατηγορίες πινάκων στην Fortran 90/95/2003.
5. Μονοδιάστατοι πίνακες.
6. Τοποθέτηση τιμών σε πίνακα.
7. Επεξεργασία-Εμφάνιση τιμών πίνακα.



Σκοποί Ενότητας

1. Η παρουσίαση των σύνθετων μεταβλητών ή πινάκων.
2. Η παρουσίαση του τρόπου επίλυσης μιας άσκησης με πίνακες.
3. Η παρουσίαση των τρόπων ορισμού ενός μονοδιάστατου πίνακα.
4. Η παρουσίαση των τρόπων εισαγωγής τιμών/ εμφάνισης τιμών σε μονοδιάστατο πίνακα.
5. Η παρουσίαση των τρόπων επεξεργασίας των στοιχείων ενός μονοδιάστατου πίνακα.



Παραδείγματα όπου χρειάζονται πίνακες

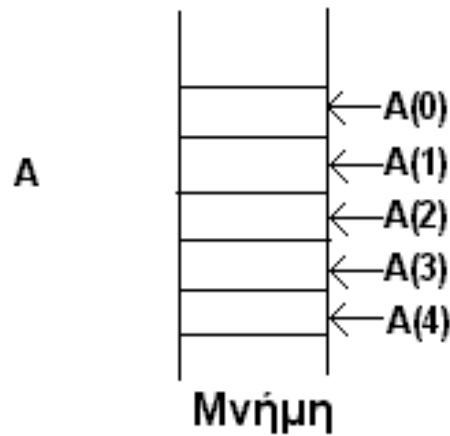
Να γραφεί πρόγραμμα το οποίο, εφόσον διαβάσει N αριθμούς, στη συνέχεια θα υπολογίζει τη μέση τιμή

$$\mu = \frac{\sum_{i=1}^N x_i}{N}, S = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N-1}}$$

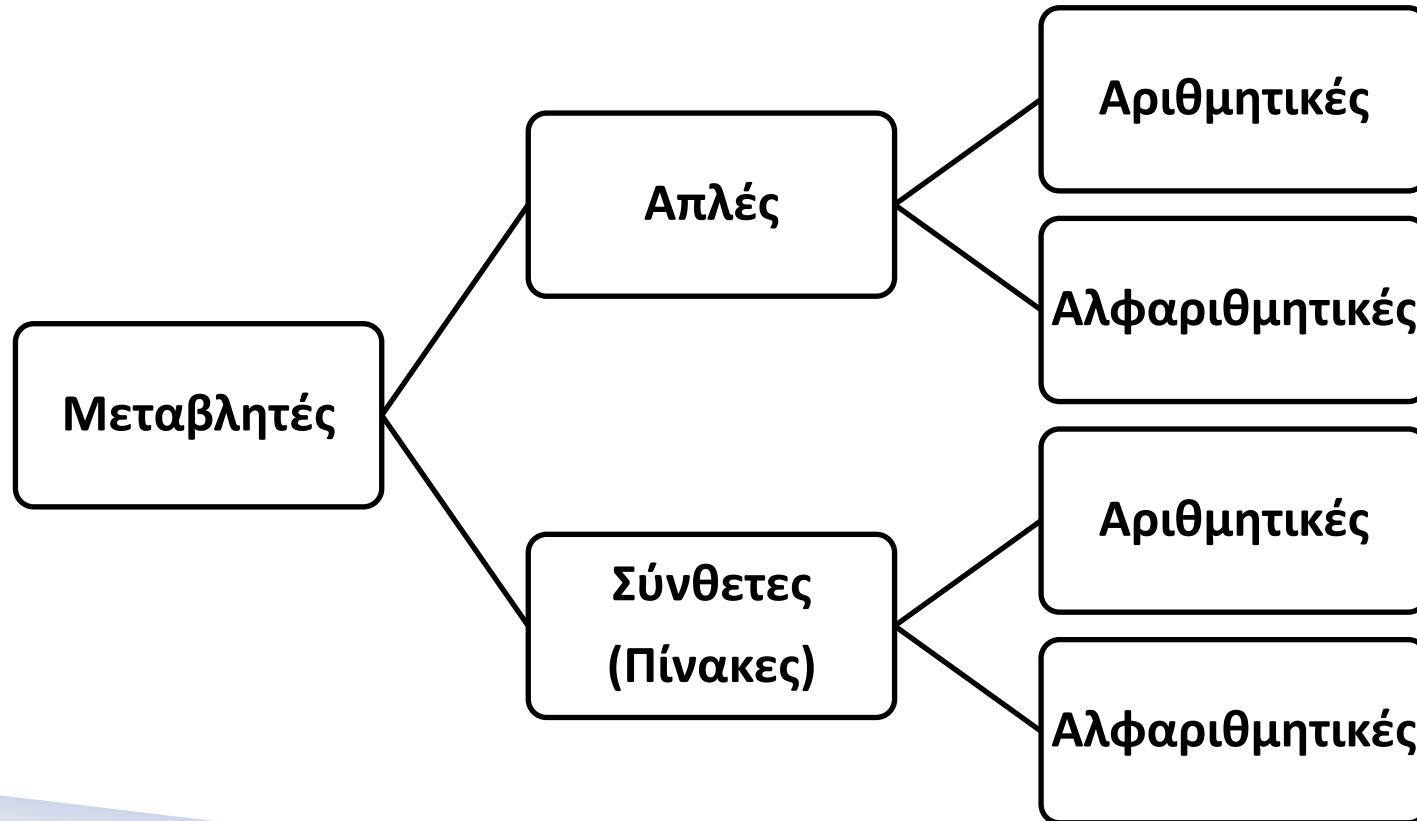


Τι είναι πίνακας ;

Πίνακας είναι μια σύνθετη μεταβλητή που καταλαμβάνει παραπάνω από μια θέση στην μνήμη του Η/Υ, έχει ένα συγκεκριμένο όνομα και δέχεται ένα συγκεκριμένο τύπο δεδομένων.



Είδη μεταβλητών στην FORTRAN 90/95/2003



Πότε χρησιμοποιούμε πίνακες ;

Συνήθεις λόγοι για τη χρήση των πινάκων είναι οι εξής :

1. Η χρησιμοποίηση των δεδομένων παραπάνω από μια φορά (πρώτο παράδειγμα).
2. Η αποθήκευση των δεδομένων πριν τη χρησιμοποίησή τους (ταξινόμηση).



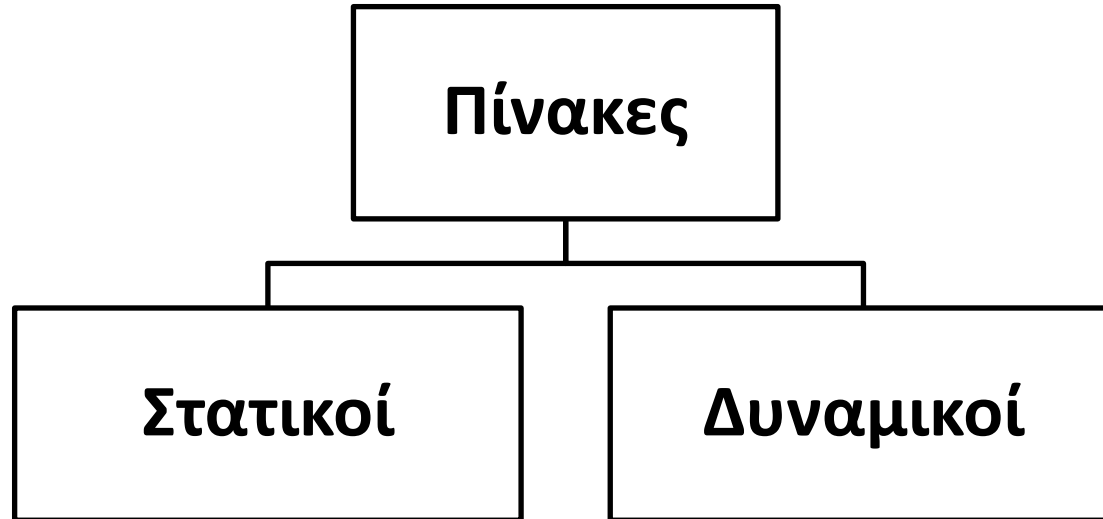
Συνήθης διαδικασία επίλυσης προβλημάτων

Η συνήθης διαδικασία επίλυσης προβλημάτων που περιλαμβάνουν πίνακες, απαρτίζεται από τα παρακάτω τέσσερα βήματα :

- Ορισμός πίνακα.
- Εισαγωγή στοιχείων στον πίνακα.
- Επεξεργασία στοιχείων πίνακα (εύρεση πλήθους, αθροίσματος και γινομένου στοιχείων που ικανοποιούν κάποια συνθήκη (ή όχι), εύρεση μικρότερου-μεγαλύτερου, ταξινόμηση στοιχείων σε αύξουσα ή φθίνουσα σειρά κ.α.)
- Εμφάνιση αποτελεσμάτων και στοιχείων του πίνακα.



Κατηγορίες πινάκων στην FORTRAN



Μονοδιάστατοι πίνακες – Ορισμός

Στατικοί πίνακες

```
REAL, DIMENSION(10) :: A
```

```
INTEGER, DIMENSION(1980:1990) :: YEAR
```

Δυναμικοί πίνακες

```
REAL, DIMENSION(:), ALLOCATABLE :: A
```

.....

```
ALLOCATE ( A(10) )
```

.....

```
DEALLOCATE( A )
```



Εναλλακτικός Ορισμός

ΑΛΛΟΣ ΤΡΟΠΟΣ

Στατικοί πίνακες

INTEGER A(10), B(20)

DOUBLE PRECISION VEL(30)

COMPLEX I(100)

CHARACTER (LEN=20) :: CODE(10)

LOGICAL TR(5)

Δυναμικοί πίνακες

REAL, ALLOCATABLE :: N(:)



Fortran 2003

ALLOCATE(λίστα πινάκων, STAT=ακέρ. μεταβλητή, ERRMSG=αλφαρ. μεταβλ.)
DEALLOCATE(λίστα πινάκων, STAT=ακέρ. μεταβλ., ERRMSG=αλφαρ. μεταβλ.)

Επιστρέφεται μήνυμα στην αλφαριθμητική μεταβλητή αν STAT/=0.

```
REAL, DIMENSION(:), ALLOCATABLE :: A
```

```
REAL, DIMENSION(5) :: B
```

```
B=(/1,2,3,4,5/)
```

```
A=B ! Αυτόματα allocate(A(5))
```



Τοποθέτηση τιμών σε πίνακα (1ος τρόπος)

1^{ος} Τρόπος (με χρήση της εντολής DO-END DO)

```
REAL A(10)
```

```
DO I=1,10
```

```
    READ*,A(I)
```

```
END DO
```



Τοποθέτηση τιμών σε πίνακα (2ος τρόπος)

2^{ος} Τρόπος (απευθείας ανάγνωση του A)

```
REAL, DIMENSION(10) :: A
```

```
READ*,A
```

READ(u2,f) P1,P2,...,Pn

όπου

u2=* ή 0 ή 5

f=είναι ο αριθμός της format (αν f=* δεν έχουμε δική μας format)
και P1,P2,...,PN είναι ονόματα πινάκων.



Τοποθέτηση τιμών σε πίνακα (3ος τρόπος)

3^{ος} Τρόπος (με implied do)

```
REAL, DIMENSION(10) :: A
```

```
READ*,(A(I),I=1,10)
```



Άσκηση 1

Να γραφεί πρόγραμμα που θα διαβάζει 15 ζευγάρια τιμών τα οποία θα αντιστοιχούν σε όνομα πόλης και βαθμό θερμοκρασίας και θα τα τοποθετεί σε δύο πίνακες αντίστοιχα.



Λύση 1.1

Σωστή λύση 1

```
CHARACTER (LEN=20), DIMENSION(15) :: TOWN
```

```
REAL, DIMENSION(15) :: TEMP
```

```
DO I=1,15
```

```
    READ*,TOWN(I),TEMP(I)
```

```
END DO
```



Λύση 1.2

Σωστή λύση 2

```
CHARACTER (LEN=20), DIMENSION(15) :: TOWN
```

```
REAL, DIMENSION(15) :: TEMP
```

```
READ*,(TOWN(I),TEMP(I),I=1,15)
```



Λάθος λύση

Λάθος λύση

```
CHARACTER (LEN=20), DIMENSION(15) :: TOWN
```

```
REAL, DIMENSION(15) :: TEMP
```

```
READ*,TOWN,TEMP
```



Τοποθέτηση τιμών μέσα από το πρόγραμμα (1ος τρόπος)

DATA λίστα μεταβλητών /λίστα σταθερών/

```
INTEGER A(5)
```

```
DATA A /15,23,34,56,78/
```

```
INTEGER A(5)
```

```
DATA (A(I),I=1,5) /15,23,34,56,78/
```



Τοποθέτηση τιμών μέσα από το πρόγραμμα (2ος τρόπος)

Όνομα πίνακα=(/λίστα τιμών/)

A = (/15,23,34,56,78/)

Fortran 2003

Όνομα πίνακα=[λίστα τιμών]

A = [15,23,34,56,78]



Τοποθέτηση τιμών που ικανοποιούν συγκεκριμένη ιδιότητα

Με χρήση DO-END DO

```
INTEGER, DIMENSION(100) :: A
```

```
DO I=1,100
```

```
    A(I)=I
```

```
END DO
```

Όνομα πίνακα=(/(συνάρτηση(I),I=αρχ. τιμή, τελ. τιμή [,βήμα])/)

```
REAL, DIMENSION(100) :: A
```

```
A=/(I,I=1,100) /)
```



Εμφάνιση των στοιχείων του πίνακα (1ος + 2ος τρόπος)

1^{ος} τρόπος (με DO-CONTINUE)

```
DO I=1,5
```

```
    PRINT*,A(I)    (ή WRITE(6,*) A(I))
```

```
END DO
```

2^{ος} τρόπος (με implied DO)

```
PRINT*,(A(I),I=1,5)    (ή WRITE(6,*) (A(I),I=1,5))
```



Εμφάνιση των στοιχείων του πίνακα (3ος τρόπος)

WRITE(u1,f) P1,P2,...,PN

όπου :

$u2=* \text{ ή } 0 \text{ ή } 6$

f είναι ο αριθμός της format (αν $f=* \text{ δεν έχουμε δική μας format}$)

και $P1,P2,...,PN$ είναι ονόματα πινάκων.

3ος τρόπος (απευθείας όλο τον πίνακα)

PRINT*,A (ή WRITE(6,*) A)



Παράδειγμα 1

Να εμφανιστούν ανά ζεύγη τα 3 πρώτα στοιχεία, από τα 5, των πινάκων A και B.

$$A = \begin{pmatrix} 10 \\ 15 \\ 21 \\ 32 \\ 43 \end{pmatrix} ; B = \begin{pmatrix} 21 \\ 22 \\ 34 \\ 12 \\ 25 \end{pmatrix}$$



Επίλυση 1

1^{ος} τρόπος (με χρήση της DO-END DO)

```
.....  
DO I=1,3  
    PRINT*,A(I),B(I)    (ή WRITE(6,*) A(I),B(I))  
END DO
```

2^{ος} τρόπος (με implied DO)

```
.....  
PRINT*,(A(I),B(I),I=1,3)    (ή WRITE(6,*) (A(I),B(I),I=1,15))
```

Λάθος τρόπος

```
.....  
PRINT*, A,B    (ή WRITE(6,*) A,B)
```



Επεξεργασία μονοδιάστατων πινάκων

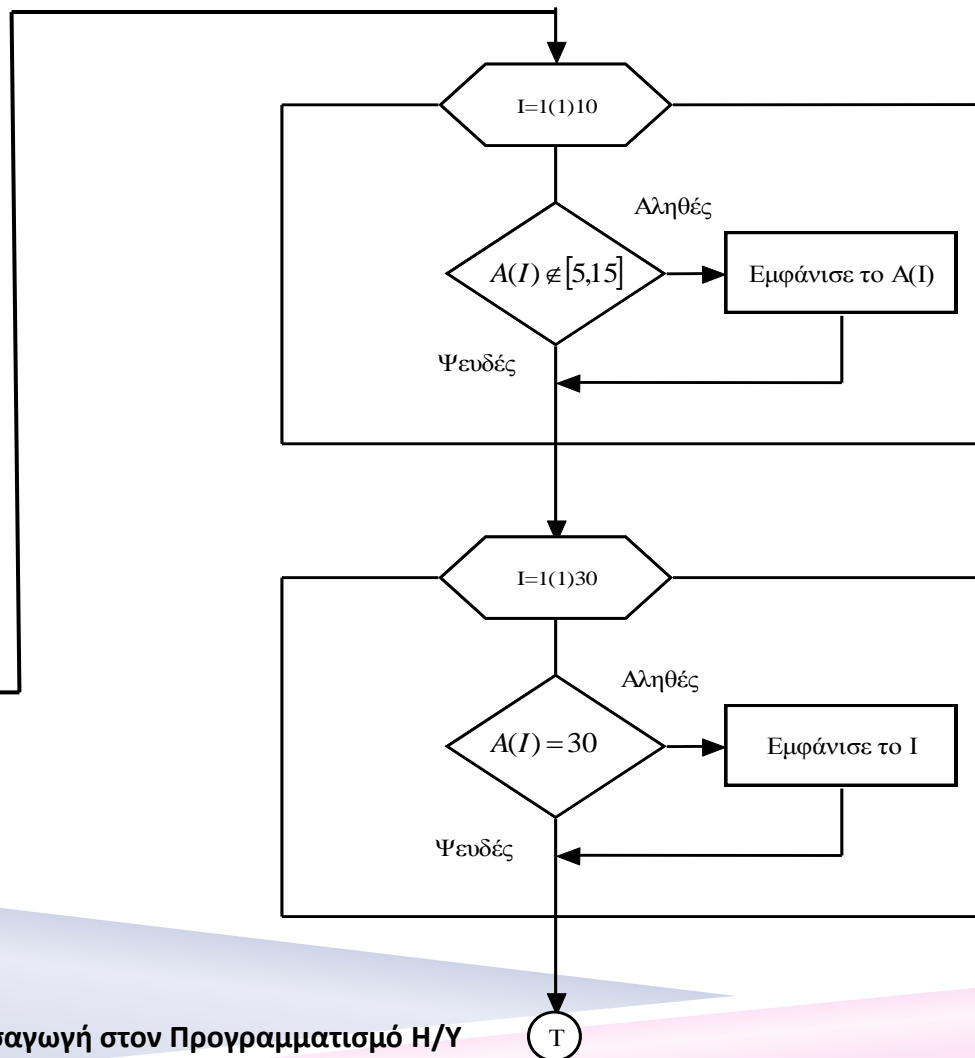
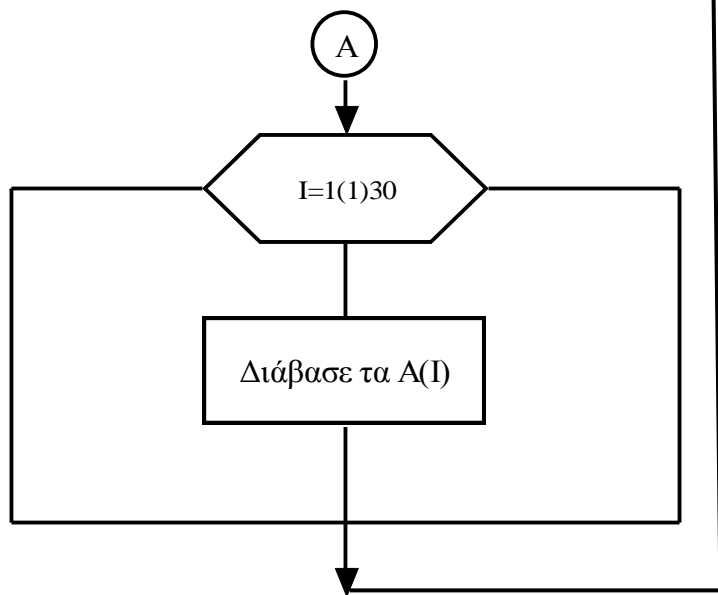
Εμφάνιση στοιχείων του πίνακα που ικανοποιούν κάποια ιδιότητα – Παράδειγμα 2

Να γραφεί πρόγραμμα το οποίο, αφού θα διαβάσει 30 αριθμούς, θα τους τοποθετεί σε ένα πίνακα και στη συνέχεια, αφού υπολογίσει, θα εμφανίζει τα εξής:

- α) τους αριθμούς που είναι έξω από το διάστημα $[5,15]$.
- β) τη θέση των στοιχείων του πίνακα που έχουν τιμή 30.



Αλγόριθμος 2



Πρόγραμμα 2

```
PROGRAM EYRESH
IMPLICIT NONE
INTEGER, DIMENSION(3)      :: A
INTEGER                    :: I
READ*,(A(I),I=1,30)
!-----
PRINT*,"ΤΙΜΕΣ ΤΟΥ ΠΙΝΑΚΑ ΣΤΟ [5,15]"
DO I=1,30
    IF ((A(I)<5).OR.(A(I)>15)) THEN
        PRINT*,A(I)
    END IF
END DO
```



Πρόγραμμα 2 (...συνέχεια)

```
!-----  
PRINT*,"ΘΕΣΕΙΣ ΤΟΥ ΠΙΝΑΚΑ ΠΟΥ ΕΧΟΥΝ ΤΙΜΗ 30"  
  DO I=1,30  
    IF (A(I)==30) THEN  
      PRINT*,I  
    END IF  
  END DO  
END PROGRAM EYRESH
```



Επεξεργασία μονοδιάστατων πινάκων

Εμφάνιση στοιχείων του πίνακα που ικανοποιούν κάποια ιδιότητα – Συνήθεις συνθήκες

	Συμβολισμός
Το x βρίσκεται στο διάστημα [A,B]	(X.GE.A.AND.X.LE.B) ((X>=A).AND.(X<=B))
Το x βρίσκεται εκτός του διαστήματος [A,B]	(X.LT.A.OR.X.GT.B) ((X<A).OR.(X>B))
Το x είναι πολλαπλάσιο του a	(MOD(X,A).EQ.0) (MOD(X,A)==0)
Το x είναι θετικό (αρνητικό)	(X.GT.0) ((X.LT.0)) (X>0) ((X<0))
Το x είναι ίσο με τον αριθμό a	(X.EQ.A) ή (X==A)
Το x είναι διάφορο του a	(X.NE.A) ή (X/=A)



Άσκηση 2

Να γράψεις πρόγραμμα το οποίο, αφού θα διαβάσει τους βαθμούς 28 φοιτητών και τους τοποθετήσει σε ένα πίνακα A, θα εμφανίσει τα εξής :

- α) τους βαθμούς που είναι μεγαλύτεροι ή ίσοι του 5,
- β) τις θέσεις των βαθμών που είναι μεγαλύτεροι ή ίσοι του 8.



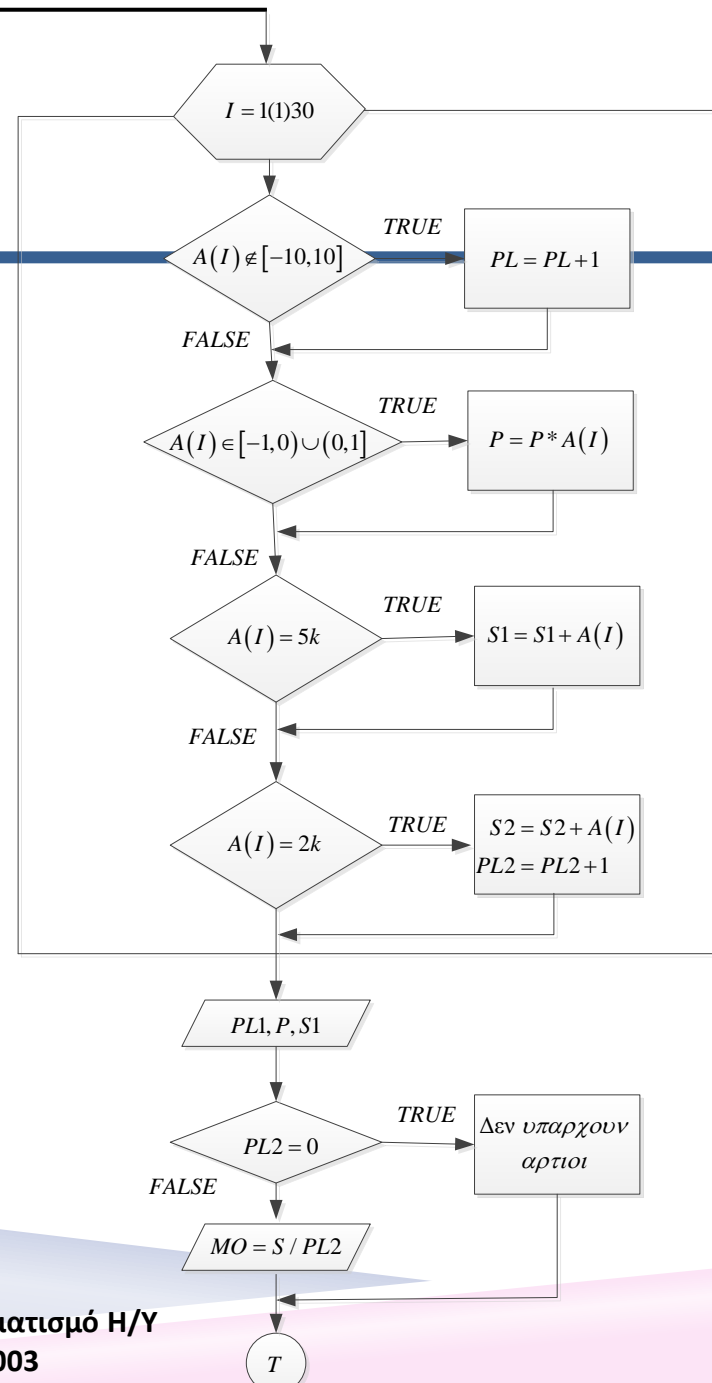
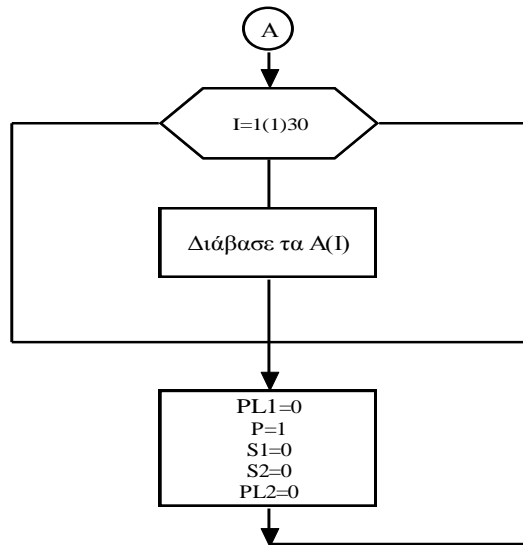
Υπολογισμός αθροίσματος, γινόμενου, πλήθους, μέσου όρου των στοιχείων ενός πίνακα που ικανοποιούν κάποια ιδιότητα

Να διαβαστούν 30 ακέραιοι αριθμοί και να τοποθετηθούν σε ένα πίνακα A. Να βρεθούν και να εμφανισθούν :

- α) το πλήθος των αριθμών που βρίσκονται έξω από το διάστημα $[-10,10]$,
- β) το γινόμενο των αριθμών που είναι διάφοροι του μηδενός και βρίσκονται στο διάστημα $[-1,1]$,
- γ) το άθροισμα των αριθμών που είναι πολλαπλάσια του 5,
- δ) ο μέσος όρος των άρτιων αριθμών.



Αλγόριθμος



Πρόγραμμα

PROGRAM EXAMPLE

```
!-----  
      IMPLICIT NONE  
      REAL                :: S1,S2,MO,P  
      INTEGER             :: I,PL1,PL2  
      INTEGER, DIMENSION(30) :: A  
      READ*, (A(I),I=1,30)  
  
!-----  
      PL1=0  
      P=1  
      S1=0  
      S2=0  
      PL2=0
```



Πρόγραμμα (...συνέχεια 1)

```
DO I=1,30
  IF ((A(I)<-10).OR.(A(I)>10)) PL1=PL1+1
  IF (((A(I)>=-1).AND.(A(I)<=1)).AND.(A(I)/=0)) P=P*A(I)
  IF (MOD(A(I),5)==0) S1=S1+A(I)
  IF (MOD(A(I),2)==0) THEN
    S2=S2+A(I)
    PL2=PL2+1
  END IF
END DO

!-----
PRINT*,"Πλήθος αριθμών έξω από το [-10,10]=",PL1
PRINT*,"Γινόμενο μη μηδενικών αριθμών μέσα στο [-1,1] =",P
PRINT*,"Άθροισμα πολλαπλασίων του 5 =",S1
```



Πρόγραμμα (...συνέχεια 2)

```
IF (PL2==0) THEN
    PRINT*,"Δεν υπάρχουν άρτιοι"
ELSE
    MO=S2/PL2
    PRINT*,"Μέσος όρος άρτιων αριθμών =",MO
END IF
END PROGRAM EXAMPLE
```



Υπολογισμός αθροίσματος, γινόμενου, πλήθους, μέσου όρου των στοιχείων ενός πίνακα που δεν ικανοποιούν κάποια ιδιότητα

Χωρίς συνθήκη	Πριν το loop	Εντός loop	Εκτός loop
Άθροισμα	$S=0$	$S=S+A(I)$	PRINT*,S
Γινόμενο	$P=1$	$P=P*A(I)$	PRINT*,P
Μέσος Όρος	$S1=0$	$S1=S1+A(I)$	MO=S1/PL1 PRINT*,MO



Υπολογισμός αθροίσματος, γινόμενου, πλήθους, μέσου όρου των στοιχείων ενός πίνακα που ικανοποιούν κάποια ιδιότητα

Με συνθήκη	Πριν το loop	Εντός loop	Εκτός loop
Άθροισμα	S=0	IF «συνθήκη» S=S+A(I)	PRINT*,S
Γινόμενο	P=1	IF «συνθήκη» P=P*A(I)	PRINT*,P
Πλήθος	PL=0	IF «συνθήκη» PL=PL+1	PRINT*,PL
Μέσος Όρος	S1=0, PL1=0	IF «συνθήκη» THEN S1=S1+A(I) PL1=PL1+1 END IF	IF (PL1==0) THEN PRINT*,"ΔΕΝ ΟΠΙΖΕΤΑΙ" ELSE MO=S1/PL1 ; PRINT*,MO END IF

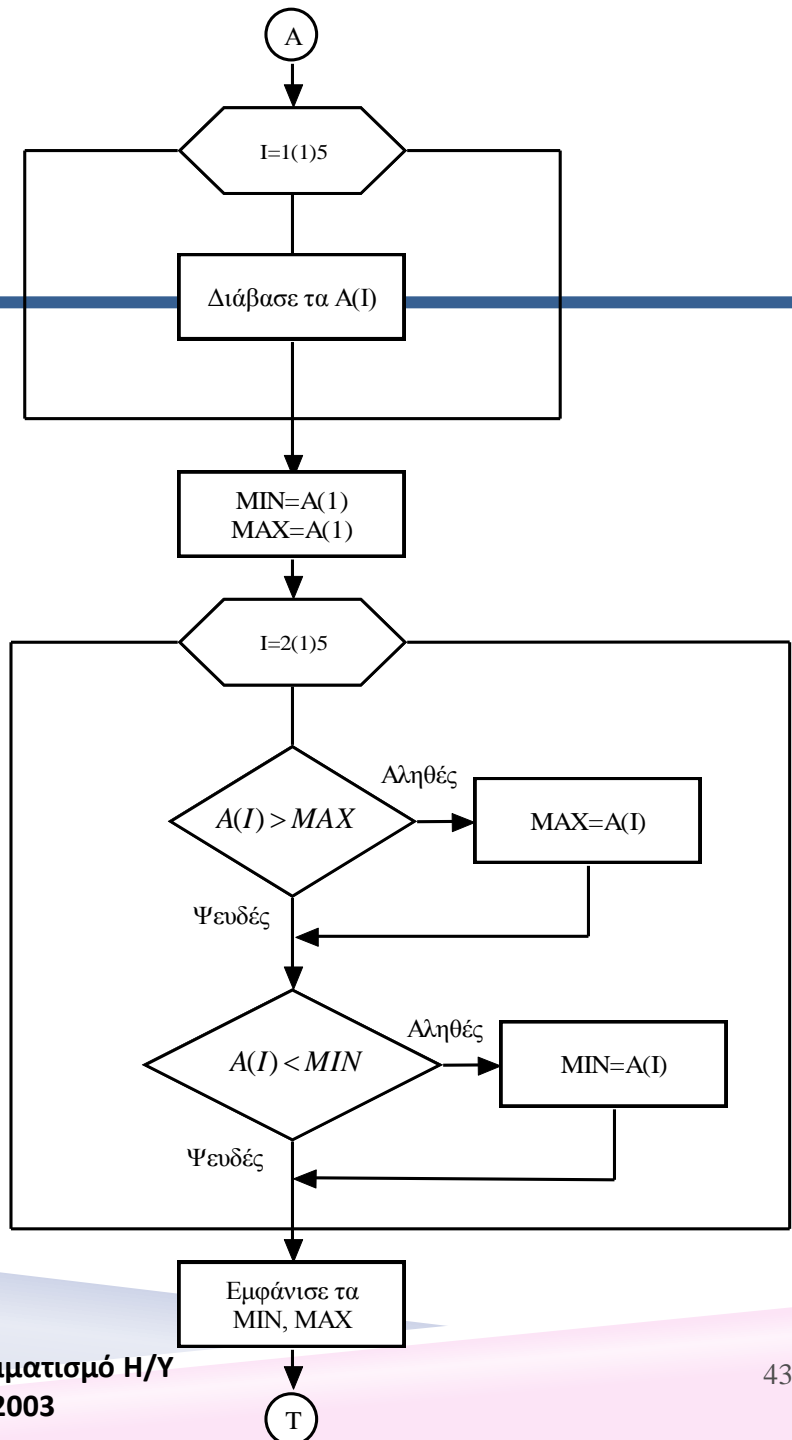


Υπολογισμός ελαχίστου, μεγίστου

Να γραφεί πρόγραμμα το οποίο, εφόσον θα διαβάσει 5 αριθμούς, θα τους τοποθετεί σε ένα πίνακα A και στη συνέχεια θα εκτυπώνει το ελάχιστο και το μέγιστο των στοιχείων αυτών.



Αλγόριθμος (Ελάχιστο – Μέγιστο)



Πρόγραμμα (Ελάχιστο – Μέγιστο)

PROGRAM EXAMPLE3

```
!-----  
      REAL A(5),MAX,MIN  
      READ*,(A(I),I=1,5)  
!-----  
      MAX=A(1)  
      MIN=A(1)  
!-----  
      DO I=1,5  
          IF (A(I)<MIN) MIN=A(I)  
          IF (A(I)>MAX) MAX=A(I)  
      END DO  
!-----  
      PRINT*,"Ελάχιστο =",MIN  
      PRINT*,"Μέγιστο =",MAX  
END PROGRAM EXAMPLE3
```



Υπολογισμός ελαχίστου, μέγιστου – Χρήσιμες συναρτήσεις

Όνομα	Περιγραφή
MAXLOC (όνομα πίνακα)	Υπολογίζει τη θέση του μέγιστου στοιχείου ενός πίνακα, π.χ. MAXLOC(A) υπολογίζει τη θέση του μέγιστου στοιχείου στον πίνακα A.
MINLOC (όνομα πίνακα)	Υπολογίζει τη θέση του ελάχιστου στοιχείου ενός πίνακα, π.χ. MINLOC(A) υπολογίζει τη θέση του ελάχιστου στοιχείου στον πίνακα A.
MAXVAL (όνομα πίνακα)	Υπολογίζει το μέγιστο στοιχείο ενός πίνακα, π.χ. MAXVAL(A) υπολογίζει το μέγιστο στοιχείο του πίνακα A.
MINVAL (όνομα πίνακα)	Υπολογίζει το ελάχιστο στοιχείο ενός πίνακα, π.χ. MINVAL(A) υπολογίζει το ελάχιστο στοιχείο του πίνακα A.



Χρήσιμες συναρτήσεις 1

Όνομα	Περιγραφή
LBOUND (όνομα πίνακα [,DIM])	Επιστρέφει ένα μονοδιάστατο πίνακα με τις ελάχιστες τιμές των δεικτών όλου του πίνακα ή αυτού που προσδιορίζεται από την παράμετρο DIM, π.χ. LBOUND(A,1) προσδιορίζει τον ελάχιστο πρώτο δείκτη του πίνακα A.
UBOUND (όνομα πίνακα [,DIM])	Επιστρέφει ένα μονοδιάστατο πίνακα με τις μέγιστες τιμές των δεικτών όλου του πίνακα ή αυτού που προσδιορίζεται από την παράμετρο DIM.
SHAPE (όνομα πίνακα)	Επιστρέφει την μορφή του πίνακα.

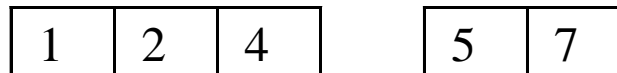
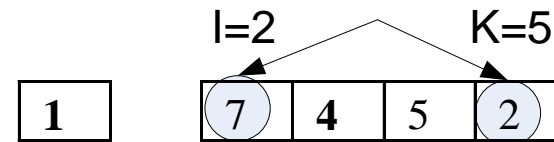
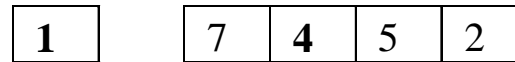
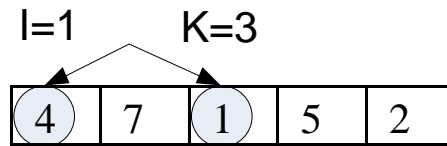


Χρήσιμες συναρτήσεις 2

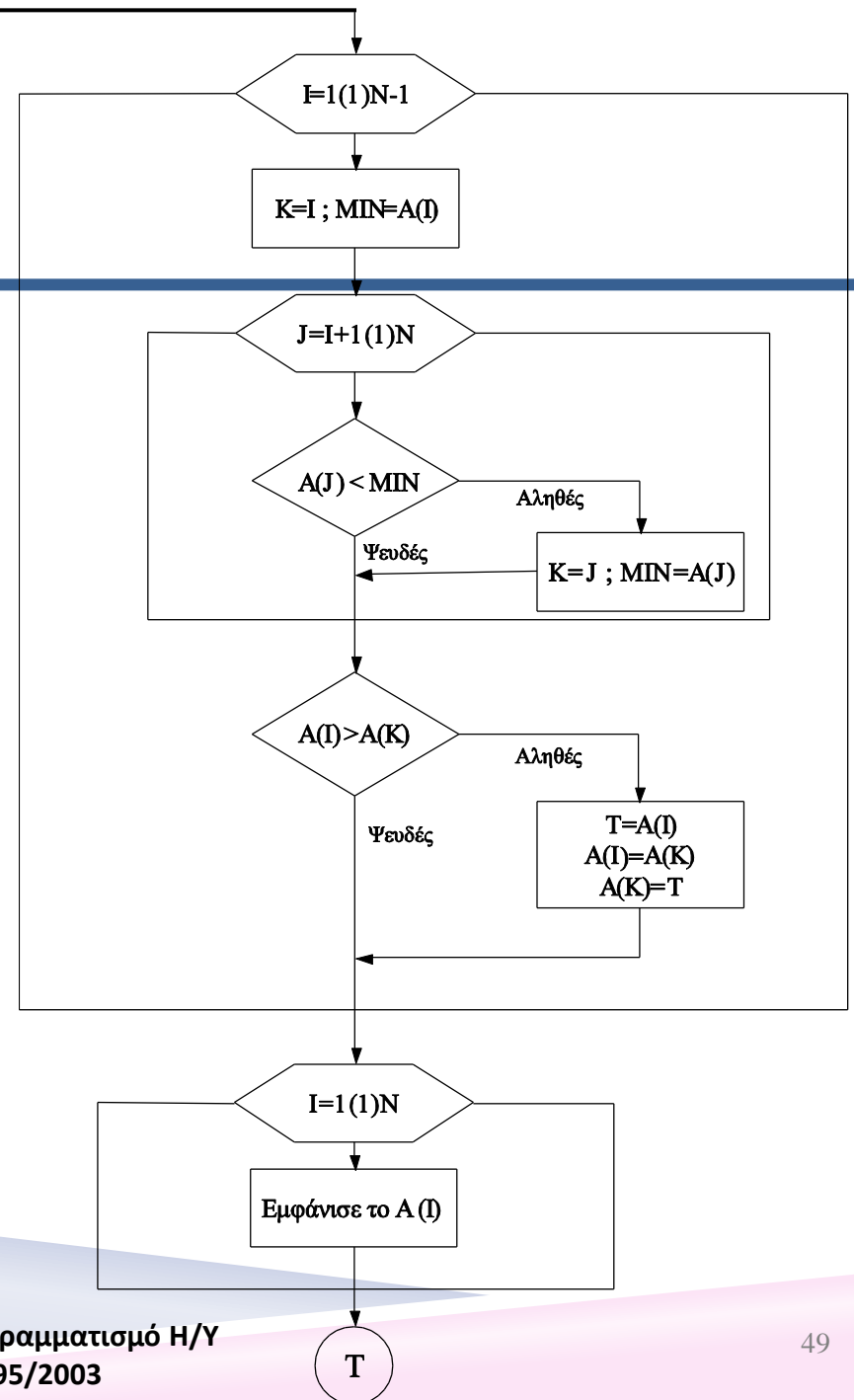
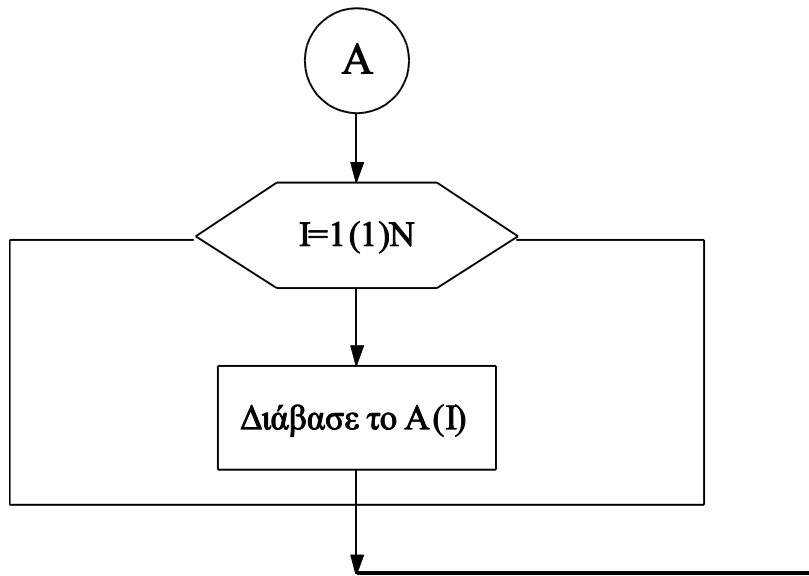
Όνομα	Περιγραφή
SIZE (όνομα πίνακα [,DIM])	Επιστρέφει τον αριθμό των στοιχείων όλου του πίνακα ή αυτού που προσδιορίζεται από την παράμετρο DIM, πρδ. SIZE(A,2) προσδιορίζει το πλήθος των τιμών που μπορεί να πάρει ο δεύτερος δείκτης του πίνακα A.
ALLOCATED (όνομα πίνακα)	Επιστρέφει true αν ο πίνακας έχει δηλωθεί με την εντολή ALLOCATED ή false διαφορετικά.
DOT_PRODUCT (όνομα πίνακα1,όνομα πίνακα2)	Επιστρέφει το εσωτερικό γινόμενο δύο μονοδιάστατων πινάκων.



Ταξινόμηση με επιλογή (selection sort)



Αλγόριθμος (Ταξινόμηση με επιλογή)



Πρόγραμμα (Ταξινόμηση με επιλογή)

PROGRAM CONSOLE1

!-----

IMPLICIT NONE

INTEGER :: I,J,K,N

INTEGER, DIMENSION(5) :: A

INTEGER :: T,MIN

A=(/4,7,1,5,2/)

N=5

!-----

DO I=1,N-1

K=I ; MIN=A(I)

DO J=I+1,N

IF (A(J)<MIN) THEN

K=J ; MIN=A(J)

END IF

END DO



Πρόγραμμα (Ταξινόμηση με επιλογή) (...συνέχεια)

```
IF (A(I)>A(K)) THEN  
  T=A(I)  
  A(I)=A(K)  
  A(K)=T  
END IF
```

```
END DO
```

```
!-----
```

```
DO I=1,N
```

```
  PRINT*,A(I)
```

```
END DO
```

```
END PROGRAM CONSOLE1
```



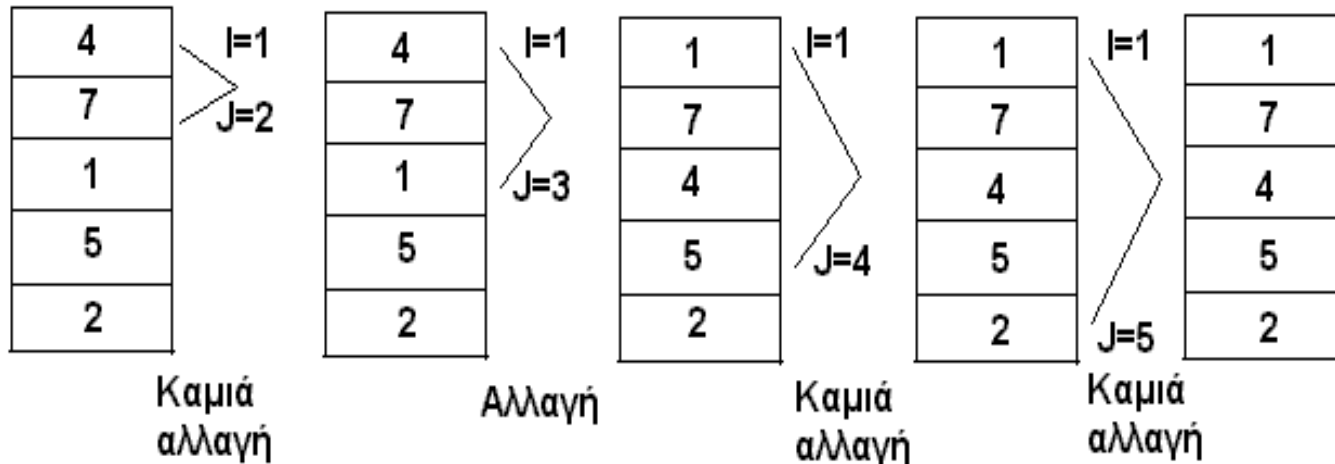
Ταξινόμηση με επιλογή (complexity)

I	Πλήθος συγκρίσεων για εύρεση ελαχίστου στον πίνακα A(I,:)	Πλήθος συγκρίσεων για σύγκριση του στοιχείου I με το στοιχείο K	Σύνολο
1	n-1	1	n
2	n-2	1	n-1
3	n-3	1	n-2
...
n-1	1	1	2
Σύνολο	$1+2+\dots+(n-1)=n(n-1)/2$	n-1	$(n+2)(n-1)/2$

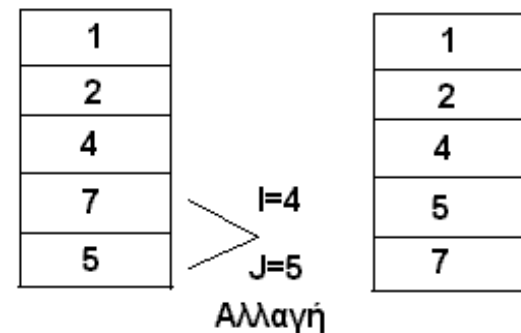
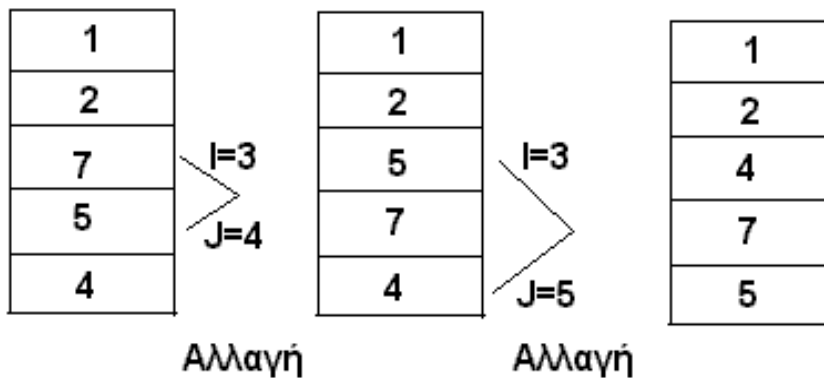
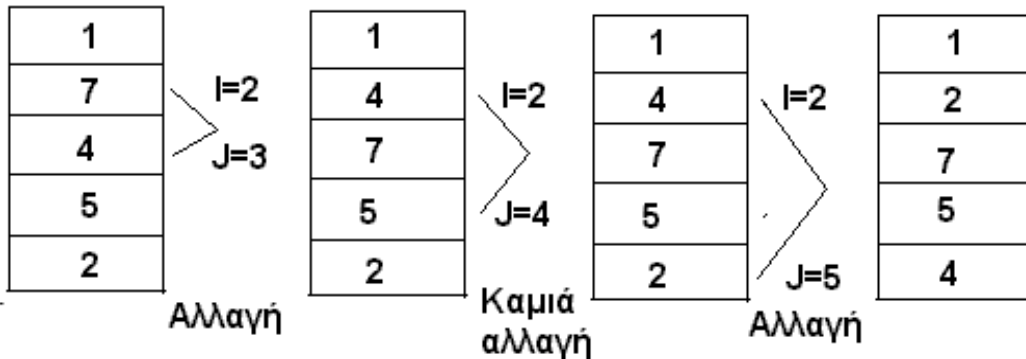


Παραλλαγή της ταξινόμησης με επιλογή

Να γραφεί πρόγραμμα που, αφού τοποθετήσει τους αριθμούς: 4,7,1,5,2 σε ένα πίνακα A, στη συνέχεια θα ταξινομήσει τον πίνακα σε *αύξουσα* σειρά.

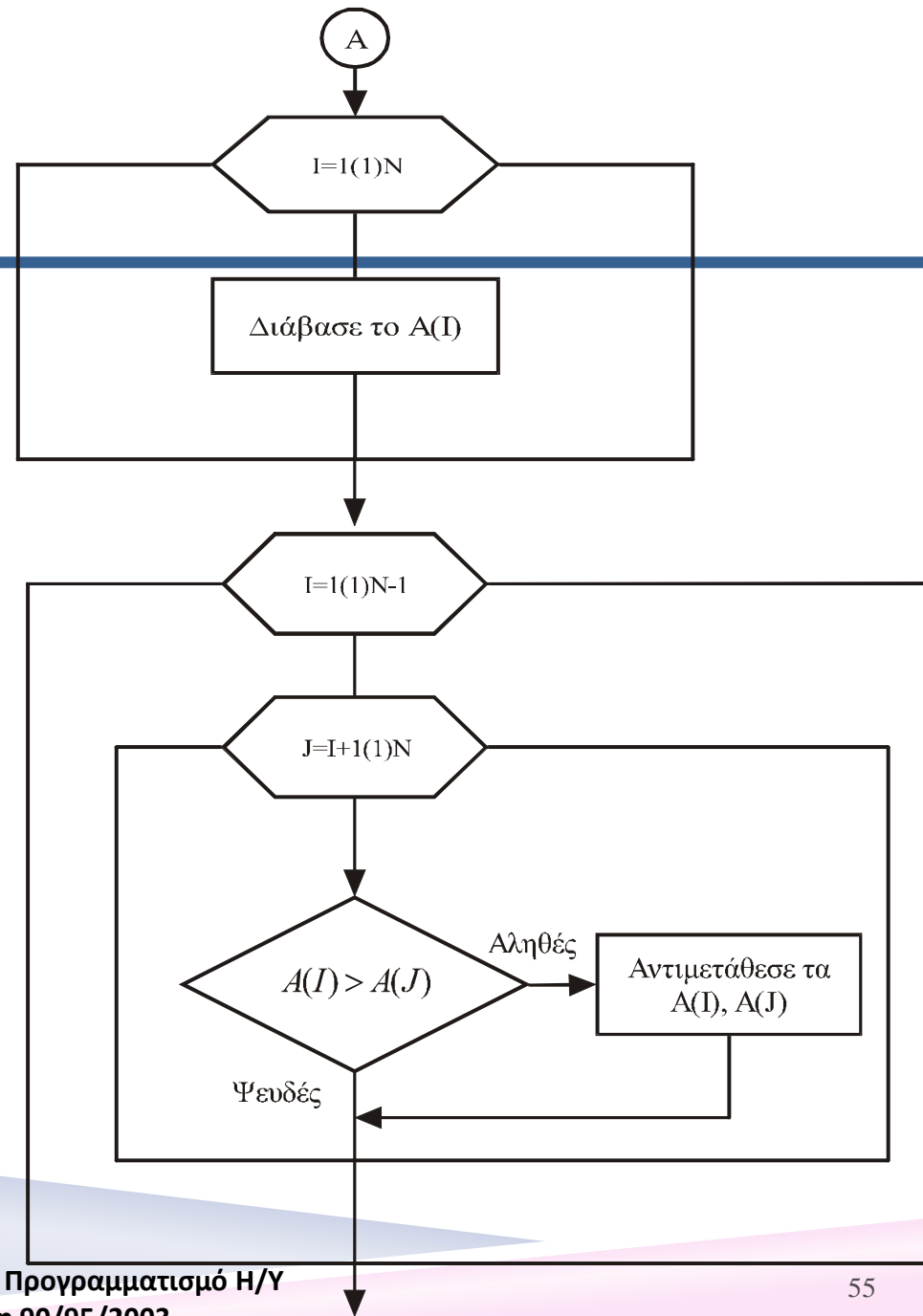


Παράδειγμα με το χέρι



Αλγόριθμος

(Ταξινόμηση σε αύξουσα σειρά)



Πρόγραμμα (Ταξινόμηση σε αύξουσα σειρά)

```
.....  
DO I=1,N-1  
    DO J=I+1,N  
        IF (A(I)>A(J)) THEN  
            K=A(I)  
            A(I)=A(J)  
            A(J)=K  
        END IF  
    END DO  
END DO
```

* Θα μπορούσαμε να χρησιμοποιήσουμε και την συνάρτηση της IMSL βιβλιοθήκης SVRGN(N,RA,RV) όπου N=πλήθος στοιχείων, RA=πίνακας που θα ταξινομηθεί, RV=ο ταξινομημένος πίνακας π.χ. CALL SVRGN (N, A, A).



Ταξινόμηση σε φθίνουσα σειρά

Σε φθίνουσα σειρά

```
DO I=1,N-1
    DO J=I+1,N
        IF (A(I)<A(J)) THEN
            K=A(I)
            A(I)=A(J)
            A(J)=K
        END IF
    END DO
END DO
```



1. Ταξινόμηση με αντιμετάθεση (bubble sort)

Συγκρίσεις	Πίνακας				
Αρχικός Πίνακας	4	7	1	5	2
A(1) με A(2)	4	7	1	5	2
A(2) με A(3)	4	1	7	5	2
A(3) με A(4)	4	1	5	7	2
A(4) με A(5)	4	1	5	2	7



2. Ταξινόμηση με αντιμετάθεση (bubble sort)

Συγκρίσεις	Πίνακας				
Αρχικός Πίνακας	4	1	5	2	7
A(1) με A(2)	1	4	5	2	7
A(2) με A(3)	1	4	5	2	7
A(3) με A(4)	1	4	2	5	7



3. Ταξινόμηση με αντιμετάθεση (bubble sort)

Συγκρίσεις	Πίνακας				
Αρχικός Πίνακας	1	4	2	5	7
A(1) με A(2)	1	4	2	5	7
A(2) με A(3)	1	2	4	5	7

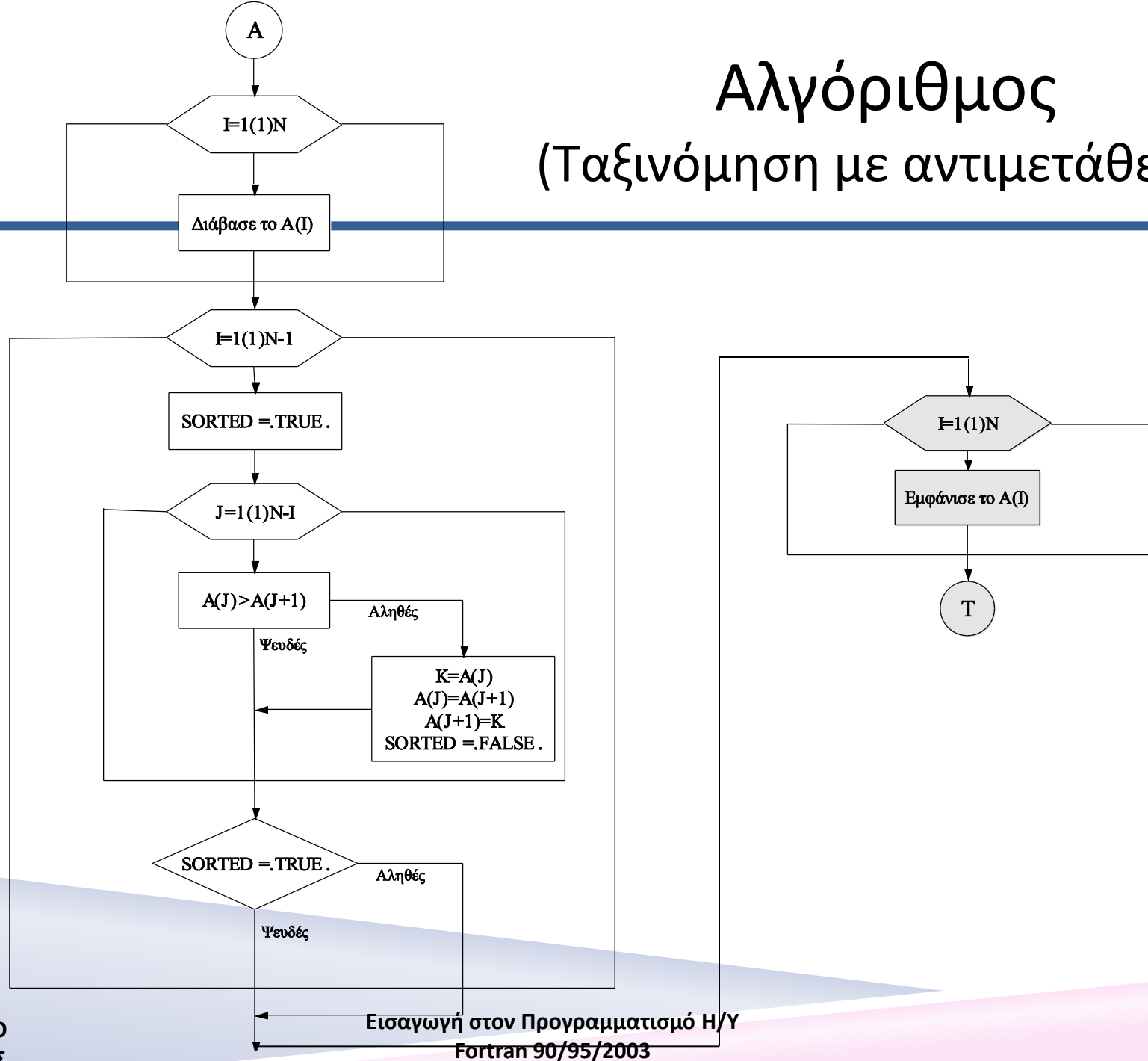


4. Ταξινόμηση με αντιμετάθεση (bubble sort)

Συγκρίσεις	Πίνακας				
Αρχικός Πίνακας	1	2	4	5	7
A(1) με A(2)	1	2	4	5	7



Αλγόριθμος (Ταξινόμηση με αντιμετάθεση)



Πρόγραμμα (Ταξινόμηση με αντιμετάθεση)

PROGRAM CONSOLE2

!-----

IMPLICIT NONE

INTEGER :: I, J, N

LOGICAL :: SORTED

REAL, DIMENSION(5) :: A

REAL :: T

A=(/4,7,1,5,2/)

N=5

!-----

DO I=1,N-1

SORTED=.TRUE.

DO J=1,N-I



Πρόγραμμα

(Ταξινόμηση με αντιμετάθεση) (...συνέχεια)

```
IF (A(J)>A(J+1)) THEN
    T=A(J)
    A(J)=A(J+1)
    A(J+1)=T
    SORTED=.FALSE.
END IF
END DO
IF (SORTED) EXIT
END DO
!-----
DO I=1,N
    PRINT*,A(I)
END DO
END PROGRAM CONSOLE2
```



Ταξινόμηση με αντιμετάθεση (complexity)

I	Πλήθος Συγκρίσεων του A(J) με το A(J+1) στον πίνακα A(1:N-I)
1	n-1
2	n-2
3	n-3
...	...
n-1	1
Σύνολο	$1+2+\dots+(n-1)=n(n-1)/2$



Αναζήτηση στοιχείου σε έναν ήδη ταξινομημένο πίνακα

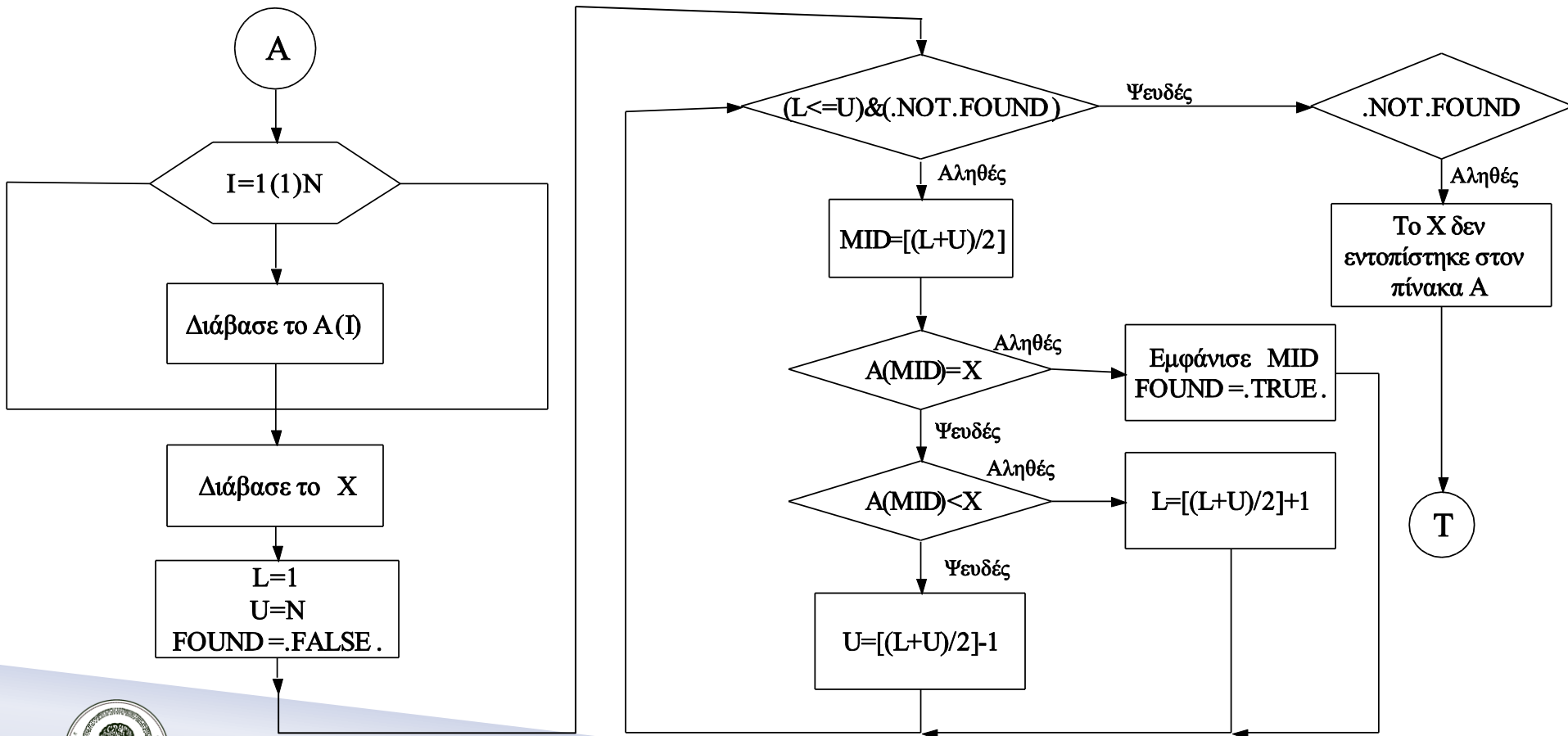
Έστω ο παρακάτω πίνακας

1	5	9	10	12	14	22	35	37	40
---	---	---	----	----	----	----	----	----	----

και ο αριθμός X είναι ο 14. Τότε ελέγχουμε αν ο αριθμός που ψάχνουμε βρίσκεται στην θέση $\left[\frac{1+10}{2} \right] = 5$. Παρατηρούμε ότι $A(5) = 12 < 14$. Συνεπώς ο αριθμός που ψάχνω βρίσκεται στις θέσεις 6 έως και 10 του πίνακα. Ελέγχουμε αν ο αριθμός που ψάχνουμε βρίσκεται στην θέση $\left[\frac{6+10}{2} \right] = 8$. Παρατηρούμε ότι $A(8) = 35 > 14$. Συνεπώς ο αριθμός που ψάχνω βρίσκεται στις θέσεις 6 έως και 7 του πίνακα. Ελέγχουμε αν ο αριθμός που ψάχνουμε βρίσκεται στην θέση $\left[\frac{6+7}{2} \right] = 6$. Παρατηρούμε ότι $A(6) = 14$ και συνεπώς η θέση του αριθμού X είναι η 6^η



Αλγόριθμος (αναζήτησης στοιχείου)



Πρόγραμμα (αναζήτησης στοιχείου)

```
PROGRAM BINARY_SEARCH
```

```
!-----
```

```
IMPLICIT NONE
```

```
INTEGER :: N,L,U,MID
```

```
REAL, DIMENSION(:), ALLOCATABLE :: A
```

```
REAL :: X
```

```
LOGICAL :: FOUND
```

```
!-----
```

```
PRINT*,"DIMENSION OF A ="
```

```
READ*,N
```

```
ALLOCATE (A(N))
```

```
PRINT*,"ELEMENTS OF A ="
```

```
READ*,A
```

```
PRINT*,"X="
```

```
READ*,X
```



Πρόγραμμα (αναζήτησης στοιχείου) (...συνέχεια)

```
L=1
U=N
FOUND=.FALSE.
DO WHILE ((L<=U).AND.(.NOT.FOUND))
  MID=(L+U)/2
  IF (A(MID)==X) THEN
    PRINT*,"H THESH TOY X EINAI :",MID
    FOUND=.TRUE.
  ELSE IF (A(MID)<X) THEN
    L=(L+U)/2+1
  ELSE
    U=(L+U)/2-1
  END IF
END DO
IF (.NOT.FOUND) THEN
  PRINT*,"TO X DEN ENTOPISTHKE STON PINAKA A"
END IF
END PROGRAM BINARY_SEARCH
```



Αναζήτηση στοιχείου σε ταξινομημένο πίνακα (complexity)

N	$T(n) = \lceil \log_2 n \rceil + 1$
10	4
100	7
1000	10
10.000	14
100.000	17
1.000.000	20
10.000.000	24



Συγχώνευση δύο ταξινομημένων πινάκων

Έστω $A=\{1,5,9,13,24\}$, $B=\{2,6,8,14,22,23,26,29\}$, $C=\{\}$.

Συγκρίνουμε το πρώτο στοιχείο του πίνακα A ($A(1)=1, I=1$) με το πρώτο στοιχείο του πίνακα B ($B(1)=2, J=1$).

Επειδή $A(1)<B(1)$ ($A(I)<B(J)$) τοποθετούμε το στοιχείο $A(1)$ ως πρώτο στην λίστα C ($C(1)=A(1)$ ή $C(K)=A(I)$ όπου $K=1$) και προχωρούμε στο επόμενο στοιχείο του πίνακα A ($I=I+1$) αλλά και του πίνακα C ($K=K+1$).

$A=\{5,9,13,24\}$, $B=\{2,6,8,14,22,23,26,29\}$, $C=\{1\}$

Στη συνέχεια συνεχίζουμε με τους δύο νέους πίνακες A και B έως ότου τελειώσουν τα στοιχεία του ενός από τους δύο πίνακες ($I>N$ ή $J>M$) (όσο δηλαδή $I\leq N$ και $J\leq M$).

$A=\{5,9,13,24\}$, $B=\{6,8,14,22,23,26,29\}$, $C=\{1,2\}$

...

$A=\{\}$, $B=\{26,29\}$, $C=\{1,2,5,6,8,9,13,14,22,23,24\}$.



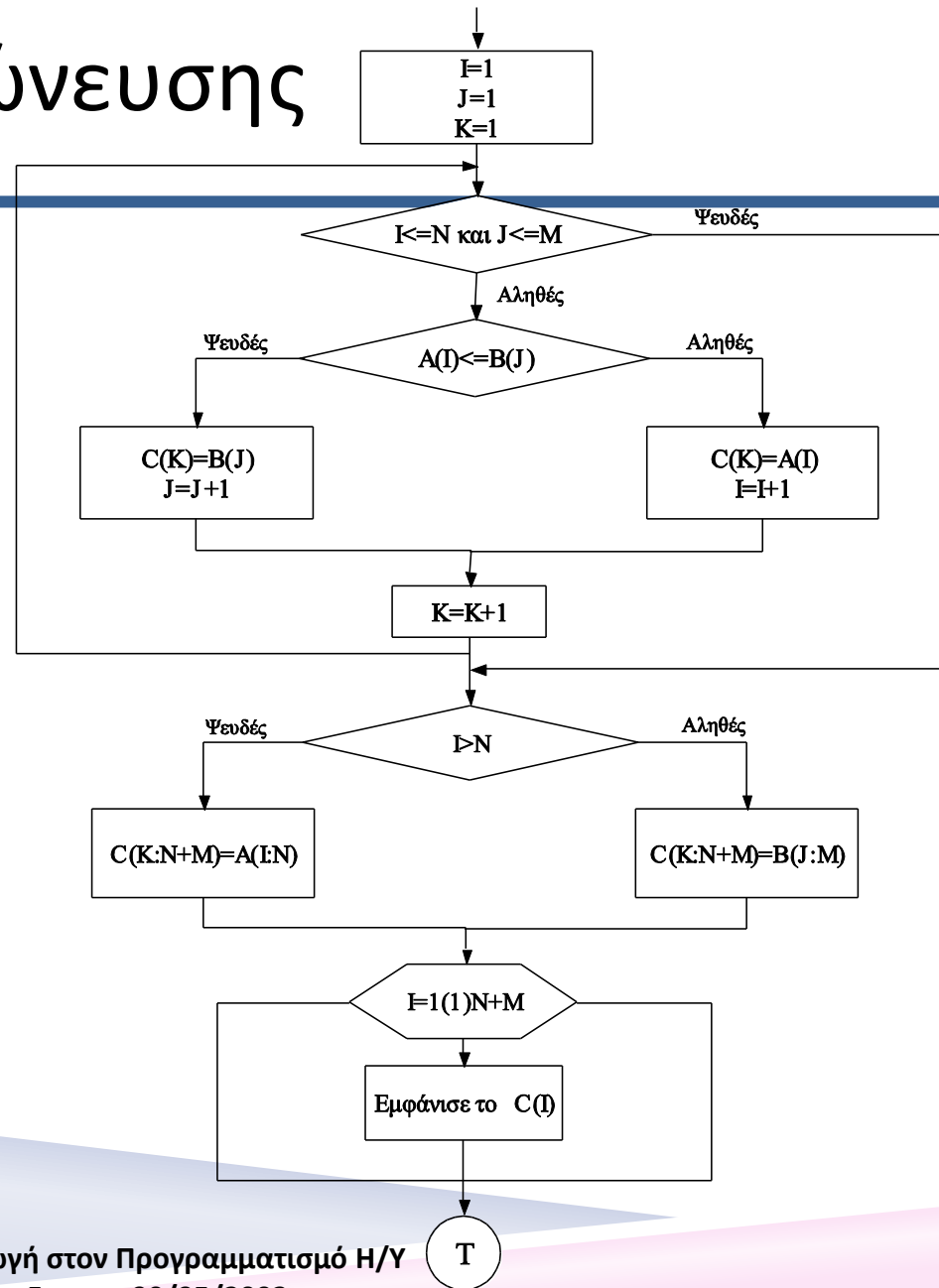
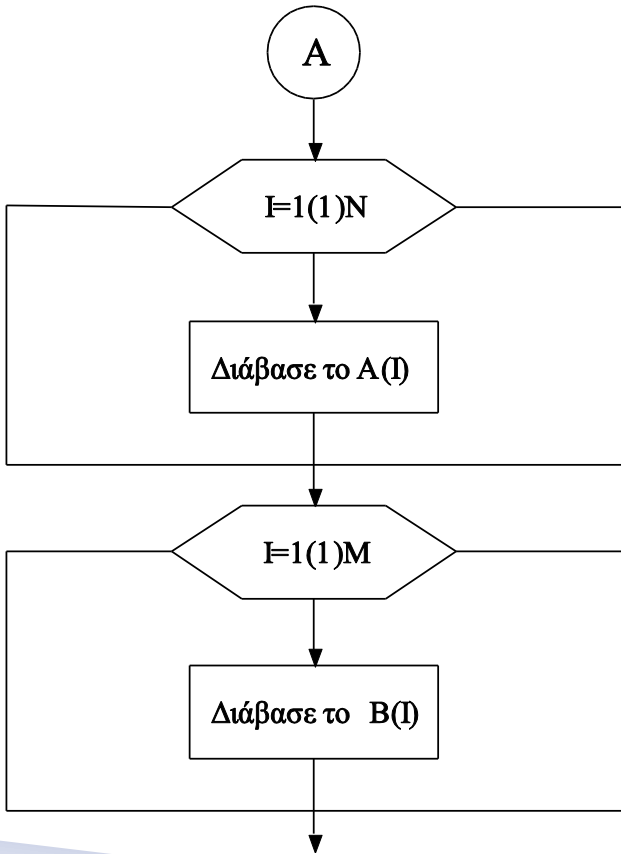
Συνέχεια συγχώνευσης

Εφόσον τελείωσαν τα στοιχεία της λίστας A, μεταφέρουμε όλα τα στοιχεία του πίνακα B στον πίνακα C($C(K:M)=B(J:M)$).

$A=\{\}, B=\{\}, C=\{1,2,5,6,8,9,13,14,22,23,24,26,29\}$.



Αλγόριθμος συγχώνευσης



Πρόγραμμα συγχώνευσης

```
PROGRAM MERGE_MATRICES
  IMPLICIT NONE
  ! STEP 1
  INTEGER :: N,M,K,I,J
  REAL, DIMENSION(:), ALLOCATABLE :: A,B,C
  ! STEP 2
  PRINT*,"DIMENSION OF A="
  READ*,N
  ALLOCATE (A(N))
  PRINT*,"DOSE TA STOIXEIA TOY PINAKA A : "
  READ*,A
  PRINT*,"DIMENSION OF B="
  READ*,M
  ALLOCATE (B(M))
  PRINT*,"DOSE TA STOIXEIA TOY PINAKA B : "
  READ*,B
  ALLOCATE (C(N+M))
```



Πρόγραμμα συγχώνευσης (...συνέχεια – 1)

! STEP 3

I=1

J=1

K=1

! STEP 4

DO WHILE ((I<=N).AND.(J<=M))

IF (A(I)<B(J)) THEN

C(K)=A(I)

I=I+1

ELSE

C(K)=B(J)

J=J+1

END IF

K=K+1

END DO



Πρόγραμμα συγχώνευσης (...συνέχεια – 2)

! STEP 5

IF (I>N) THEN

C(K:N+M)=B(J:M)

ELSE

C(K:N+M)=A(I:N)

END IF

! STEP 6

PRINT*, "-----"

PRINT*, "C="

PRINT*, C(1:N+M)

END PROGRAM MERGE_MATRICES



Επεξεργασία μονοδιάστατων πινάκων – Επεξεργασία στοιχείων πίνακα με καινούριο τύπο δεδομένων

Να διαβασθούν τα ονόματα των 12 μηνών και οι αντίστοιχες μέσες μηνιαίες θερμοκρασίες που είχαμε το έτος 2001. Στη συνέχεια να υπολογιστεί η ελάχιστη/μέγιστη μέση τιμή θερμοκρασίας καθώς και ο μήνας που εμφανίσθηκε.



Ορισμός του νέου τύπου δεδομένων

PROGRAM EXAMPLE5

```
!-----  
      IMPLICIT NONE  
      TYPE TEMPS  
          CHARACTER (LEN=20)           :: MONTH  
          REAL                       :: TEMP  
      END TYPE TEMPS  
      TYPE (TEMPS), DIMENSION(12)     :: MONTEMP  
      REAL                           :: MIN,MAX  
      INTEGER                         :: I  
!-----
```



Ορισμός του νέου τύπου δεδομένων (...συνέχεια – 1)

```
DO I=1,12
    PRINT*,I,"MONTH="
    READ*,MONTEMP(I).MONTH
    PRINT*,I,"TEMPERATURE="
    READ*,MONTEMP(I).TEMP
END DO
!-----
    MAX=MONTEMP(1).TEMP
    MIN=MONTEMP(1).TEMP
!-----
    DO I=2,12
        IF ((MONTEMP(I).TEMP)>MAX) MAX= MONTEMP(I).TEMP
        IF ((MONTEMP(I).TEMP)<MIN) MIN= MONTEMP(I).TEMP
    END DO
```



Ορισμός του νέου τύπου δεδομένων (...συνέχεια – 2)

```
PRINT*,"Η μέγιστη θερμοκρασία είναι =",MAX
DO I=1,12
    IF ((MONTEMP(I).TEMP)==MAX) THEN
        PRINT*, MONTEMP(I).MONTH
    END IF
END DO
!-----
PRINT*,"Η ελάχιστη θερμοκρασία είναι =",MIN
DO I=1,12
    IF ((MONTEMP(I).TEMP)==MIN) THEN
        PRINT*, MONTEMP(I).MONTH
    END IF
END DO
END PROGRAM EXAMPLE5
```



Βιβλιογραφία

Ν. Καραμπετάκης, *Εισαγωγή στη Fortran 90/95/2003*, 2^η Έκδοση, Θεσσαλονίκη: Εκδόσεις Ζήτη, 2011.



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Καραμπετάκης Νικόλαος. «Εισαγωγή στον Προγραμματισμό Η/Υ (Fortran 90/95/2003). Μονοδιάστατοι Πίνακες». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://eclass.auth.gr/courses/OCRS145/>.



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, 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-sa/4.0/>





Τέλος Ενότητας

Επεξεργασία: Αναστασία Γ. Γρηγοριάδου
Θεσσαλονίκη, Χειμερινό Εξάμηνο 2013-2014

