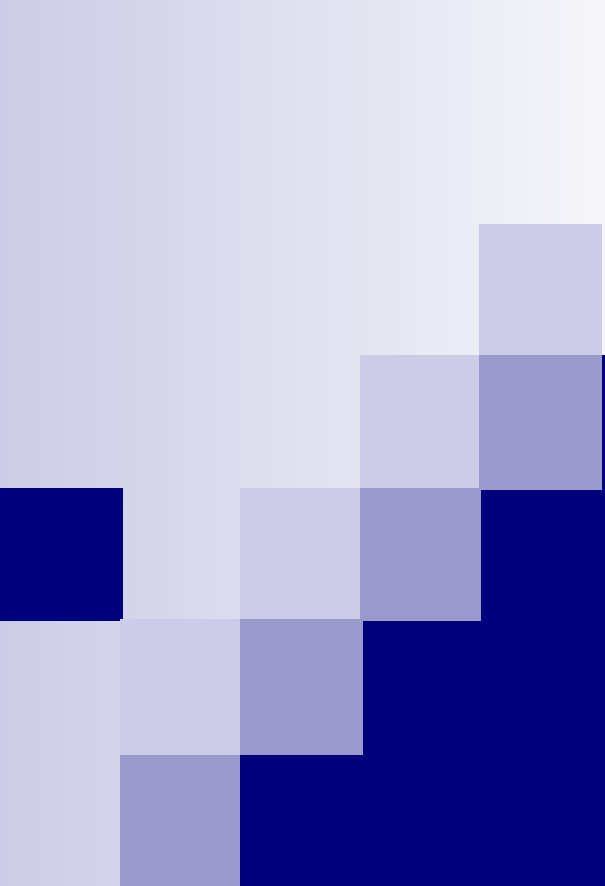


Don't reinvent the wheel!



IBM



ΚΕΦΑΛΑΙΟ 10^ο

ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

{B1} - Κεφ. 10ο (Σελ. 170-184)

{OM} - Κεφ. 5ο (Σελ. 97-121)

ΤΜΗΜΑΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

- Ονομάζεται η τεχνική **σχεδίασης** και **ανάπτυξης** των προγραμμάτων ως ένα **σύνολο** από απλούστερα **τμήματα** προγραμμάτων.
- **Πλεονεκτήματα:**
 - 1) Διευκολύνει την **ανάπτυξη** του **αλγορίθμου** και του αντίστοιχου προγράμματος (**Ανάλυση** σε **υποπροβλήματα** και επίλυση τους).
 - 2) Διευκολύνει την **κατανόηση** και **διόρθωση** του προγράμματος.
 - 3) Μειώνει το **μέγεθος** του προγράμματος, άρα και τον **χρόνο** **ανάπτυξης** του κώδικα.
 - 4) **Επεκτείνει** τις **δυνατότητες** των γλωσσών προγραμματισμού, με την δημιουργία **βιβλιοθηκών** από χρήσιμα υποπρογράμματα (όπως οι **ενσωματωμένες συναρτήσεις** στην «ΓΛΩΣΣΑ»).

7.6 | Συναρτήσεις

Πολλές γνωστές συναρτήσεις από τα μαθηματικά χρησιμοποιούνται συχνά και περιέχονται στη ΓΛΩΣΣΑ. Οι συναρτήσεις αυτές είναι:

HM(X)	Υπολογισμός ημιτόνου
ΣΥΝ(X)	Υπολογισμός συνημιτόνου
ΕΦ(X)	Υπολογισμός εφαπτομένης
T_P(X)	Υπολογισμός τετραγωνικής ρίζας
ΛΟΓ(X)	Υπολογισμός φυσικού λογαρίθμου
E(X)	Υπολογισμός του e^x
A_M(X)	Ακέραιο μέρος του X
A_T(X)	Απόλυτη τιμή του X



Python help() Function

Built-in Functions in Python

abs()	classmethod()	filter()	id()	max()	property()	str()
all()	compile()	float()	input()	memoryview()	range()	sum()
any()	complex()	format()	int()	min()	repr()	super()
ascii()	delattr()	frozenset()	isinstance()	next()	reversed()	tuple()
bin()	dict()	getattr()	issubclass()	object()	round()	type()
bool()	dir()	globals()	iter()	oct()	set()	vars()
bytearray()	divmod()	hasattr()	len()	open()	setattr()	zip()
bytes()	enumerate()	hash()	list()	ord()	slice()	__import__()
callable()	eval()	help()	locals()	pow()	sorted()	
chr()	exec()	hex()	map()	print()	staticmethod()	

ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

- Ονομάζεται κάθε **τμήμα** προγράμματος που επιτελεί **αυτόνομο έργο** και έχει **γραφεί χωριστά** από το υπόλοιπο **κύριο πρόγραμμα**.
- **Ιδιότητες:**
 - 1) Πρέπει να έχει **μία είσοδο** από τη **αρχή** του υποπρογράμματος και **μία έξοδο** από το **τέλος**.
 - 2) Πρέπει να είναι **ανεξάρτητο** από άλλα υποπρογράμματα.
 - 3) Πρέπει να **μην είναι πολύ μεγάλο** σε μέγεθος, ώστε να είναι εύκολα κατανοητό και να μπορεί να ελέγχεται. Γενικά πρέπει να εκτελεί **μόνο μία βασική λειτουργία**.

ΚΑΤΗΓΟΡΙΕΣ ΥΠΟΠΡΟΓΡΑΜΜΑΤΩΝ

■ ΣΥΝΑΡΤΗΣΕΙΣ (FUNCTIONS)

- Υπολογίζουν **μόνο μία τιμή** και την **επιστρέφουν** στο πρόγραμμα που τις καλεί.
- Μοιάζουν με τις μαθηματικές συναρτήσεις { **$HM(X)$, $ΛΟΓ(X)$, $A_T(X)$, $T_P(X)$** }.
- Εκτελούνται με την **εμφάνιση του ονόματος** τους σε οποιαδήποτε έκφραση.

■ ΔΙΑΔΙΚΑΣΙΕΣ (PROCEDURES)

- Εκτελούν **οποιαδήποτε λειτουργία** μπορεί να εκτελέσει και ένα πρόγραμμα.
- Μεταφέρουν τα **αποτελέσματά** τους με την χρήση των **παραμέτρων**.
- Εκτελούνται όταν χρησιμοποιείται η εντολή «**ΚΑΛΕΣΕ**» στο πρόγραμμα.



- ❖ Ταξινόμηση Πίνακα ➤ ΔΙΑΔΙΚΑΣΙΑ
- ❖ Μέγιστη τιμή Πίνακα ➤ ΣΥΝΑΡΤΗΣΗ
- ❖ Υπάρχει το X στον Πίνακα; ➤ ΣΥΝΑΡΤΗΣΗ
- ❖ Αντιμετάθεση δύο τιμών ➤ ΔΙΑΔΙΚΑΣΙΑ

ΔΟΜΗ ΜΙΑΣ ΣΥΝΑΡΤΗΣΗΣ

```
ΣΥΝΑΡΤΗΣΗ Όνομα (<λίστα παραμέτρων>) : <τύπος συνάρτησης>  
  <Τμήμα δηλώσεων>  
ΑΡΧΗ  
  ...  
  Όνομα <- έκφραση  
  ...  
ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
```

- Το όνομα της Συνάρτησης είναι οποιοδήποτε **έγκυρο όνομα** της ΓΛΩΣΣΑΣ.
- Μια Συνάρτηση λοιπόν μπορεί να είναι: **ΠΡΑΓΜΑΤΙΚΗ, ΑΚΕΡΑΙΑ, ΧΑΡΑΚΤΗΡΑΣ, ΛΟΓΙΚΗ**.
- Στις εντολές του σώματος της Συνάρτησης πρέπει **υποχρεωτικά** να υπάρχει μία **εντολή εκχώρησης τιμής στο όνομα** της Συνάρτησης.

ΔΙΑΒΑΣΕ α, β

sum $\leftarrow$ Άθροισμα (α, β)

MO $\leftarrow$ Μέσο_όρο (α, β)

Συνάρτηση
υπολογισμού
αθροίσματος

ΣΥΝΑΡΤΗΣΗ Άθροισμα (x, y): ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: x, y

ΑΡΧΗ

Άθροισμα $\leftarrow x + y$

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

Συνάρτηση
υπολογισμού
μέσου όρου

ΣΥΝΑΡΤΗΣΗ Μέσο_όρο (x, y): ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: x, y

ΑΡΧΗ

Μέσο_όρο $\leftarrow (x + y) / 2$

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

Το <όνομα> της
συνάρτησης έχει δύο
ρόλους:

1. Χρησιμοποιείται για να καλέσουμε τη συνάρτηση στο κύριο πρόγραμμα ή σε άλλο υποπρόγραμμα.
2. Χρησιμοποιείται σαν μεταβλητή μέσα στις εντολές της συνάρτησης όπου καταχωρείται σ' αυτή η τελική τιμή της συνάρτησης την οποία και μεταφέρει εκεί που την καλούμε.

ΔΟΜΗ ΜΙΑΣ ΔΙΑΔΙΚΑΣΙΑΣ

```
ΔΙΑΔΙΚΑΣΙΑ Όνομα(<λίστα παραμέτρων>)  
    <Τμήμα δηλώσεων>  
ΑΡΧΗ  
    <εντολές>  
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
```

- Το όνομα της Διαδικασίας είναι οποιοδήποτε **έγκυρο όνομα** της ΓΛΩΣΣΑΣ.
- Στο σώμα της Διαδικασίας μπορούν να υπάρχουν **οποιοσδήποτε εντολές** της ΓΛΩΣΣΑΣ, όπως οι εντολές **ΔΙΑΒΑΣΕ** και **ΓΡΑΨΕ**.
- Η **λίστα παραμέτρων** είναι μια λίστα μεταβλητών, των οποίων οι τιμές **μεταβιβάζονται** προς τη **διαδικασία** κατά την **κλήση** ή/και **επιστρέφονται** στο **κύριο πρόγραμμα** μετά το τέλος της διαδικασίας.

ΔΟΜΗ ΜΙΑΣ ΔΙΑΔΙΚΑΣΙΑΣ

ΚΑΛΕΣΕ <Όνομα διαδικασίας> (<λίστα παραμέτρων>)

ΚΑΛΕΣΕ Γέμισμα_Πίνακα (γ)

{
ΔΙΑΔΙΚΑΣΙΑ Γέμισμα_Πίνακα (Α)
ΜΕΤΑΒΛΗΤΕΣ
ΑΚΕΡΑΙΕΣ: Α[100], i
ΑΡΧΗ
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100
ΓΡΑΨΕ 'Δώσε ', i, ' στοιχείο'
ΔΙΑΒΑΣΕ Α[i]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
}

ΟΛΑ ΜΑΖΙ ΣΕ ΕΝΑ ΠΡΟΓΡΑΜΜΑ

ΠΡΟΓΡΑΜΜΑ ΠΑΡΑΔΕΙΓΜΑ ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: X , Ψ

ΑΡΧΗ

ΔΙΑΒΑΣΕ X, Ψ

X ← ΑΠΟΛΥΤΗ_ΤΙΜΗ(X)

Ψ ← ΑΠΟΛΥΤΗ_ΤΙΜΗ(Ψ)

ΚΑΛΕΣΕ ΑΝΤΙΜΕΤΑΘΕΣΕ(X,Ψ)

ΓΡΑΨΕ X , Ψ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΣΥΝΑΡΤΗΣΗ ΑΠΟΛΥΤΗ_ΤΙΜΗ(A) : ΑΚΕΡΑΙΑ ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A

ΑΡΧΗ

ΑΝ A < 0 ΤΟΤΕ

A ← -1 * A

ΤΕΛΟΣ_ΑΝ

ΑΠΟΛΥΤΗ_ΤΙΜΗ ← A

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΔΙΑΔΙΚΑΣΙΑ ΑΝΤΙΜΕΤΑΘΕΣΕ(A,B)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, B, TEMP

ΑΡΧΗ

TEMP ← A

A ← B

B ← TEMP

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Πραγματικές Παράμετροι: X, Ψ

Τυπικές Παράμετροι (Ορίσματα): A, B

ΕΙΣΟΔΟΣ: 15 , -20

ΕΞΟΔΟΣ: 20 , 15

ΠΡΟΓΡΑΜΜΑ ΠΙΝΑΚΕΣ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Α[100] , Β[100] , Γ[100]

ΑΡΧΗ

ΚΑΛΕΣΕ ΕΙΣΑΓΩΓΗ(Α)

ΚΑΛΕΣΕ ΕΙΣΑΓΩΓΗ(Β)

ΚΑΛΕΣΕ ΕΙΣΑΓΩΓΗ(Γ)

ΓΡΑΨΕ ΜΙΝ(Α) , ΜΙΝ(Β) , ΜΙΝ(Γ)

ΚΑΛΕΣΕ ΤΑΞΙΝΟΜΗΣΗ(Α)

ΚΑΛΕΣΕ ΤΑΞΙΝΟΜΗΣΗ(Β)

ΚΑΛΕΣΕ ΤΑΞΙΝΟΜΗΣΗ(Γ)

ΓΡΑΨΕ ΜΟ(Α) , ΜΟ(Β) , ΜΟ(Γ)

ΚΑΛΕΣΕ ΕΜΦΑΝΙΣΗ(Α)

ΚΑΛΕΣΕ ΕΜΦΑΝΙΣΗ(Β)

ΚΑΛΕΣΕ ΕΜΦΑΝΙΣΗ(Γ)

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

**ΣΥΝΑΡΤΗΣΗ ΜΟ(Π) : ΠΡΑΓΜΑΤΙΚΗ
ΜΕΤΑΒΛΗΤΕΣ**

ΑΚΕΡΑΙΕΣ: Π[100], i, Σ

ΑΡΧΗ

Σ ← 0

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

Σ ← Σ + Π[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΟ ← Σ / 100

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

**ΣΥΝΑΡΤΗΣΗ ΜΙΝ(Π) : ΑΚΕΡΑΙΑ
ΜΕΤΑΒΛΗΤΕΣ**

ΑΚΕΡΑΙΕΣ: Π[100], i, m

ΑΡΧΗ

m ← Π[1]

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100

ΑΝ Π[i] < m ΤΟΤΕ

m ← Π[i]

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΜΙΝ ← m

ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

ΔΙΑΔΙΚΑΣΙΑ ΤΑΞΙΝΟΜΗΣΗ(Π)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Π[100], i , j

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 100

ΓΙΑ j ΑΠΟ 100 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1

ΑΝ Π[j-1] > Π[j] ΤΟΤΕ

ΚΑΛΕΣΕ ΑΝΤΙΜΕΤΑΘΕΣΕ(Π[j-1], Π[j])

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΑΝΤΙΜΕΤΑΘΕΣΕ(A,B)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: A, B, TEMP

ΑΡΧΗ

TEMP ← A

A ← B

B ← TEMP

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΕΙΣΑΓΩΓΗ(Π)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Π[100], i

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΔΙΑΒΑΣΕ Π[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΕΜΦΑΝΙΣΗ(Π)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Π[100], i

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 100

ΓΡΑΨΕ Π[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

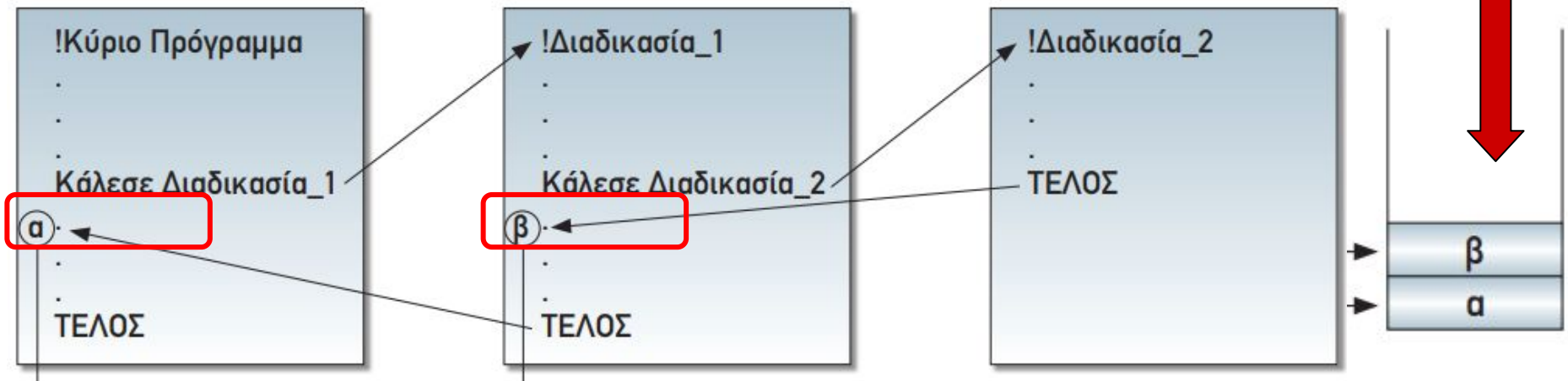


Ιδιαίτερη προσοχή χρειάζονται τα παρακάτω σημεία:

- **Τοπικότητα μεταβλητών.** Οι μεταβλητές του κυρίου προγράμματος, είναι γνωστές μόνο στο πρόγραμμα. Έξω από το πρόγραμμα, κανένα υποπρόγραμμα - διαδικασία δεν τις γνωρίζει και δεν μπορεί να τις χρησιμοποιήσει. Το ίδιο ισχύει και για τις μεταβλητές των υποπρογραμμάτων διαδικασιών. Για αυτό λέμε, ότι οι μεταβλητές έχουν ισχύ τοπική (εντός του προγράμματος-υποπρογράμματος που δηλώνονται).
- **Επικοινωνία υποπρογραμμάτων.** Ο μοναδικός τρόπος επικοινωνίας είναι μέσω των παραμέτρων (πραγματικών-τυπικών). Οι παράμετροι είναι μεταβλητές του κυρίου προγράμματος και των υποπρογραμμάτων.
- **Αντιστοιχία παραμέτρων - κανόνες τυπικών και πραγματικών παραμέτρων.**
 1. Ο αριθμός των πραγματικών και τυπικών παραμέτρων πρέπει να είναι ο ίδιος.
 2. Κάθε πραγματική παράμετρος αντιστοιχεί στην τυπική παράμετρο που βρίσκεται στην αντίστοιχη θέση.
 3. Η τυπική παράμετρος και η αντίστοιχη πραγματική παράμετρος πρέπει να είναι του ίδιου τύπου.

Η ΧΡΗΣΗ ΣΤΟΙΒΑΣ ΣΤΗΝ ΚΛΗΣΗ ΔΙΑΔΙΚΑΣΙΩΝ

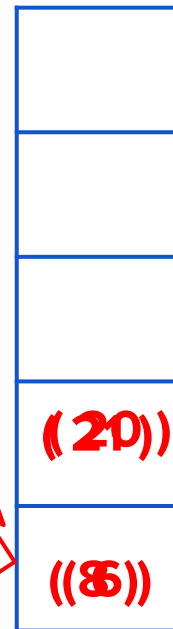
- ❖ Όταν μία Διαδικασία ή Συνάρτηση καλείται από το κύριο πρόγραμμα, τότε η **αμέσως επόμενη διεύθυνση** του κύριου προγράμματος, που ονομάζεται διεύθυνση επιστροφής (**return address**),
- ❖ Αποθηκεύεται από τον **μεταφραστή** (compiler) σε μία στοίβα που ονομάζεται **στοίβα χρόνου εκτέλεσης** (execution time stack)



```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: Α, Β
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(Α,Β)
6   ΓΡΑΨΕ Α , Β
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(Β,Α)
8   ΓΡΑΨΕ Α + Β
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
10
11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(Α,Β)
12   ....
13   Α <- 10
14   Β <- 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
16
17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(Β,Α)
18   ....
19   Β <- ΣΥΝ_01(Α,Β)
20   Α <- ΣΥΝ_02(Β,Α)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```



```

23 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_01(Α,Β):ΑΚΕΡΑΙΑ
24   ....
25   ΣΥΝ_01 <- Α - Β
26 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
27
28
29 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_02(Β,Α):ΑΚΕΡΑΙΑ
30   ....
31   ΣΥΝ_02 <- Α + Β
32 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: Α, Β
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(Α,Β)
6   ΓΡΑΨΕ Α , Β
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(Β,Α)
8   ΓΡΑΨΕ Α + Β
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
10
11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(Α,Β)
12   ....
13   Α <- 10
14   Β <- 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
16
17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(Β,Α)
18   ....
19   Β <- ΣΥΝ_01(Α,Β)
20   Α <- ΣΥΝ_02(Β,Α)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

A	B
ΔΙΑΔ_01	

```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: Α, Β
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(Α,Β)
6   ΓΡΑΨΕ Α , Β
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(Β,Α)
8   ΓΡΑΨΕ Α + Β
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
10
11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(Α,Β)
12   ....
13   Α <- 10
14   Β <- 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
16
17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(Β,Α)
18   ....
19   Β <- ΣΥΝ_01(Α,Β)
20   Α <- ΣΥΝ_02(Β,Α)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```



A	B
10	20
ΔΙΑΔ_01	

```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: A, B
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(A,B)
6   ΓΡΑΨΕ A , B
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(B,A)
8   ΓΡΑΨΕ A + B
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

ΟΘΟΝΗ

10 , 20

```

10
11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(A,B)
12   ....
13   A ← 10
14   B ← 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
16
17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(B,A)
18   ....
19   B ← ΣΥΝ_01(A,B)
20   A ← ΣΥΝ_02(B,A)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

A B

10 20

ΔΙΑΔ_01

```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: Α, Β
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(Α,Β)
6   ΓΡΑΨΕ Α , Β
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(Β,Α)
8   ΓΡΑΨΕ Α + Β
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

```

11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(Α,Β)
12   ....
13   Α ← 10
14   Β ← 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

```

17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(Β,Α)
18   ....
19   Β ← ΣΥΝ_01(Α,Β)
20   Α ← ΣΥΝ_02(Β,Α)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

```

23 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_01(Α,Β):ΑΚΕΡΑΙΑ
24   ....
25   ΣΥΝ_01 ← Α - Β
26 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
27
28
29 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_02(Β,Α):ΑΚΕΡΑΙΑ
30   ....
31   ΣΥΝ_02 ← Α + Β
32 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

A	B
10	20
ΔΙΑΔ_01	



B	A
ΔΙΑΔ_02	

```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: Α, Β
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(Α,Β)
6   ΓΡΑΨΕ Α , Β
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(Β,Α)
8   ΓΡΑΨΕ Α + Β
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

```

11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(Α,Β)
12   ....
13   Α <- 10
14   Β <- 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
16
17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(Β,Α)
18   ....
19   Β <- ΣΥΝ_01(Α,Β)
20   Α <- ΣΥΝ_02(Β,Α)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

```

23 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_01(Α,Β):ΑΚΕΡΑΙΑ
24   ....
25   ΣΥΝ_01 <- Α - Β
26 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
27
28
29 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_02(Β,Α):ΑΚΕΡΑΙΑ
30   ....
31   ΣΥΝ_02 <- Α + Β
32 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

A	B
10	20
ΔΙΑΔ_01	

A	B
10	20
-10	
ΣΥΝ_01	

B	A
ΔΙΑΔ_02	



```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: Α, Β
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(Α,Β)
6   ΓΡΑΨΕ Α , Β
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(Β,Α)
8   ΓΡΑΨΕ Α + Β
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

```

11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(Α,Β)
12   ....
13   Α <- 10
14   Β <- 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ
16
17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(Β,Α)
18   ....
19   Β <- ΣΥΝ_01(Α,Β)
20   Α <- ΣΥΝ_02(Β,Α)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

```

23 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_01(Α,Β):ΑΚΕΡΑΙΑ
24   ....
25   ΣΥΝ_01 <- Α - Β
26 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
27
28
29 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_02(Β,Α):ΑΚΕΡΑΙΑ
30   ....
31   ΣΥΝ_02 <- Α + Β
32 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

A	B
10	20
ΔΙΑΔ_01	

A	B
10	20
-10	
ΣΥΝ_01	



B	A
-10	
ΔΙΑΔ_02	

```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: Α, Β
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(Α,Β)
6   ΓΡΑΨΕ Α , Β
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(Β,Α)
8   ΓΡΑΨΕ Α + Β
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

```

11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(Α,Β)
12   ....
13   Α <- 10
14   Β <- 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

```

17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(Β,Α)
18   ....
19   Β <- ΣΥΝ_01(Α,Β)
20   Α <- ΣΥΝ_02(Β,Α)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

```

23 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_01(Α,Β):ΑΚΕΡΑΙΑ
24   ....
25   ΣΥΝ_01 <- Α - Β
26 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ
27
29 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_02(Β,Α):ΑΚΕΡΑΙΑ
30   ....
31   ΣΥΝ_02 <- Α + Β
32 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

A	B
10	20
ΔΙΑΔ_01	

B	A
-10	10
0	
ΣΥΝ_02	

B	A
-10	
ΔΙΑΔ_02	



```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: Α, Β
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(Α,Β)
6   ΓΡΑΨΕ Α , Β
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(Β,Α)
8   ΓΡΑΨΕ Α + Β
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

```

11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(Α,Β)
12   ....
13   Α <- 10
14   Β <- 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

```

17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(Β,Α)
18   ....
19   Β <- ΣΥΝ_01(Α,Β)
20   Α <- ΣΥΝ_02(Β,Α)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

```

23 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_01(Α,Β):ΑΚΕΡΑΙΑ
24   ....
25   ΣΥΝ_01 <- Α - Β
26 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

```

27
28
29 ΣΥΝΑΡΤΗΣΗ ΣΥΝ_02(Β,Α):ΑΚΕΡΑΙΑ
30   ....
31   ΣΥΝ_02 <- Α + Β
32 ΤΕΛΟΣ_ΣΥΝΑΡΤΗΣΗΣ

```

A	B
10	20
ΔΙΑΔ_01	

B	A
-10	10
0	
ΣΥΝ_02	

B	A
-10	0
ΔΙΑΔ_02	



```

1 ΠΡΟΓΡΑΜΜΑ ΣΤΟΙΒΑ_ΕΚΤΕΛΕΣΗΣ
2 ΜΕΤΑΒΛΗΤΕΣ
3   ΑΚΕΡΑΙΕΣ: A, B
4 ΑΡΧΗ
5   ΚΑΛΕΣΕ ΔΙΑΔ_01(A,B)
6   ΓΡΑΨΕ A , B
7   ΚΑΛΕΣΕ ΔΙΑΔ_02(B,A)
8   ΓΡΑΨΕ A + B
9 ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

ΟΘΟΝΗ
10, 20
-10

```

11 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_01(A,B)
12   ....
13   A <- 10
14   B <- 20
15 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

A	B
10	20
ΔΙΑΔ_01	

```

17 ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΔ_02(B,A)
18   ....
19   B <- ΣΥΝ_01(A,B)
20   A <- ΣΥΝ_02(B,A)
21 ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

```

B	A
-10	0
ΔΙΑΔ_02	

10.6 | Εμβέλεια μεταβλητών-σταθερών

ΟΡΙΣΜΟΣ

Το τμήμα του προγράμματος που ισχύουν οι μεταβλητές λέγεται **εμβέλεια** (*scope*) μεταβλητών.

Περιορισμένη εμβέλεια



Όλες οι μεταβλητές (και οι σταθερές) είναι **τοπικές**, ισχύουν δηλαδή μόνο για το υποπρόγραμμα στο οποίο δηλώθηκαν.

Στη **ΓΛΩΣΣΑ** έχουμε περιορισμένη εμβέλεια.

Απεριόριστη εμβέλεια

- ❖ Σύμφωνα με αυτή την αρχή όλες οι μεταβλητές και όλες οι σταθερές είναι **Καθολικές**, δηλαδή είναι γνωστές και μπορούν να χρησιμοποιούνται σε οποιοδήποτε τμήμα του προγράμματος, άσχετα που δηλώθηκαν.
- ❖ Η απεριόριστη εμβέλεια καταστρατηγεί την αρχή της **αυτονομίας** των υποπρογραμμάτων, δημιουργεί πολλά προβλήματα και τελικά είναι αδύνατη για μεγάλα προγράμματα με πολλά υποπρογράμματα, αφού ο καθένας που γράφει κάποιο υποπρόγραμμα πρέπει **να γνωρίζει τα ονόματα όλων των μεταβλητών** που χρησιμοποιούνται στα υπόλοιπα υποπρογράμματα.

Μερικώς περιορισμένη εμβέλεια

- ❖ Σύμφωνα με αυτή την αρχή άλλες μεταβλητές είναι **τοπικές** και άλλες **καθολικές**.

- ❖ Η μερικώς περιορισμένη εμβέλεια προσφέρει μερικά **πλεονεκτήματα στον πεπειραμένο** προγραμματιστή, αλλά για τον αρχάριο περιπλέκει το πρόγραμμα δυσκολεύοντας την ανάπτυξή του.

```
1
2
3 var global = 10;
4
5 function fun() {
6
7     var local = 5;
8
9 }
10
11
12
13
```

global variable

local variable

2019 Επαναληπτικές - Θέμα Β2' (Πίνακας Τιμών)

B2. Δίνεται το παρακάτω πρόγραμμα και ένα υποπρόγραμμα:

Πρόγραμμα Θέμα_B2 Μεταβλητές Ακέραιες: a,b Αρχή $a \leftarrow 1$ $b \leftarrow 3$ Όσο $a < 35$ επανάλαβε <u>Κάλεσε Διαδ(a,b)</u> <u>Γράψε b</u> Τέλος_επανάληψης Τέλος_Προγράμματος	Διαδικασία Διαδ(a,b) Μεταβλητές Ακέραιες: a,b Αρχή $b \leftarrow b+a$ $a \leftarrow a+8$ <u>Γράψε a</u> Τέλος_Διαδικασίας
---	--

Να γράψετε στο τετράδιό σας τις τιμές που θα εμφανιστούν κατά την εκτέλεση του προγράμματος με τη σειρά που θα εμφανιστούν.

Μονάδες 10

B2. Δίνεται το παρακάτω πρόγραμμα και ένα υποπρόγραμμα:

Πρόγραμμα Θέμα_B2
Μεταβλητές

ΔΙΑΔ(a,b)

a	b	ΟΘΟΝΗ
1	3	<i>Γράψε a</i>
9	4	9
17	13	17
25	30	25
34	55	34
42	89	42

Διαδικασία Διαδ(a,b)

Μεταβλητές

Ακέραιες: a,b

Αρχή

$b \leftarrow b+a$

$a \leftarrow a+8$

Γράψε a

Τέλος_Διαδικασίας

τιμές που θα εμφανιστούν κατά την
σειρά που θα εμφανιστούν.

Μονάδες 10

B2. Δίνεται το παρακάτω πρόγραμμα και ένα υποπρόγραμμα:

Πρόγραμμα Θέμα_B2

Μεταβλητές

Ακέραιες: a,b

Αρχή

$a \leftarrow 1$

$b \leftarrow 3$

Όσο $a < 35$ επανάλαβε

Καλέσσε Διαδ(a,b)

Γράψε b

Τέλος_επανάληψης

Διαδικασία Διαδ(a,b)

ΔΙΑΔ(a,b)

a	b	ΟΘΟΝΗ
1	3	Γράψε a
9	4	9
17	13	17
25	30	25
34	55	34
42	89	42

ν κατά την

μονάδες 10

ΠΡΟΓΡΑΜΜΑ

a	b	ΟΘΟΝΗ
1	3	Γράψε b

9 → 4 → 17 → 13 →

25 → 30 → 34 → 55 → 42 → 89 →

Στο άθλημα των 100 μέτρων, στους δύο ημιτελικούς συμμετέχουν 16 αθλητές (8+8). Στον τελικό προκρίνεται ο πρώτος κάθε ημιτελικού, καθώς και οι έξι (6) καλύτεροι χρόνοι από όλους τους υπολοίπους. Να γράψετε πρόγραμμα το οποίο: Α) περιλαμβάνει τμήμα δηλώσεων, Β) καλεί την διαδικασία ΕΙΣΟΔΟΣ για κάθε ημιτελικό, η οποία διαβάζει τα ονόματα των αθλητών και τον χρόνο τους, Γ) καλεί την διαδικασία ΤΑΞΙΝΟΜΗΣΗ για κάθε ημιτελικό, η οποία ταξινομεί τους αθλητές ως προς τον χρόνο τους σε αύξουσα σειρά, Δ) δημιουργεί και εμφανίζει τον πίνακα **ON** με τα ονόματα και τον πίνακα **XP** με τους χρόνους των αθλητών που προκρίθηκαν στον τελικό.

ON1[8] - XP1[8]

AΘ1	9,86
AΘ2	9,92
AΘ3	9,95
AΘ4	10,2
AΘ5	10,3
AΘ6	9,90
AΘ7	10,4
AΘ8	9,75

ON2[8] - XP2[8]

AΘ9	9,87
AΘ10	9,93
AΘ11	9,96
AΘ12	10,25
AΘ13	10,35
AΘ14	9,97
AΘ15	10,45
AΘ16	9,78



ON[8] - XP[8]

AΘ8	9,75
AΘ16	9,78
AΘ1	9,86
AΘ9	9,87
AΘ6	9,90
AΘ2	9,92
AΘ10	9,93
AΘ3	9,95

2008 Επαναληπτικές - Θέμα Δ'

ΠΡΟΓΡΑΜΜΑ ΣΤΙΒΟΣ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, A, B

ΠΡΑΓΜΑΤΙΚΕΣ: XR1[8], XR2[8], XR[8]

ΧΑΡΑΚΤΗΡΕΣ: ON1[8], ON2[8], ON[8]

ΑΡΧΗ

ΚΑΛΕΣΕ ΕΙΣΟΔΟΣ(ON1, XR1)

ΚΑΛΕΣΕ ΕΙΣΟΔΟΣ(ON2, XR2)

ΚΑΛΕΣΕ ΤΑΞΙΝΟΜΗΣΗ(ON1, XR1)

ΚΑΛΕΣΕ ΤΑΞΙΝΟΜΗΣΗ(ON2, XR2)

ON[1] ← ON1[1] *!Ο πρώτος στον Α' ημιτ*

XR[1] ← XR1[1]

ON[2] ← ON2[1] *!Ο πρώτος στον Β' ημιτ*

XR[2] ← XR2[1]

A ← 2 *! Δείκτης για τον Α' Ημιτελικό*

B ← 2 *! Δείκτης για τον Β' Ημιτελικό*

ΓΙΑ i ΑΠΟ 3 ΜΕΧΡΙ 8

ΑΝ XR1[A] < XR2[B] **ΤΟΤΕ**

ON[i] ← ON1[A]

XR[i] ← XR1[A]

A ← A + 1

ΑΛΛΙΩΣ

ON[i] ← ON2[B]

XR[i] ← XR2[B]

B ← B + 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 8

ΓΡΑΨΕ ON[i], XR[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΤΑΞΙΝΟΜΗΣΗ(ΟΝ, ΧΡ)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, j,

ΠΡΑΓΜΑΤΙΚΕΣ: ΧΡ[8], t1

ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[8], t2

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ 8

ΓΙΑ j ΑΠΟ 8 ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1

ΑΝ ΧΡ[j-1] > ΧΡ[j] ΤΟΤΕ

t1 ← ΧΡ[j-1]

ΧΡ[j-1] ← ΧΡ[j]

ΧΡ[j] ← t1

t2 ← ΟΝ[j-1]

ΟΝ[j-1] ← ΟΝ[j]

ΟΝ[j] ← t2

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΕΙΣΟΔΟΣ(ΟΝ, ΧΡ)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i

ΠΡΑΓΜΑΤΙΚΕΣ: ΧΡ[8]

ΧΑΡΑΚΤΗΡΕΣ: ΟΝ[8]

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ 8

ΔΙΑΒΑΣΕ ΟΝ[i], ΧΡ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ