Μονάδα Δικτύων και Επικοινωνιών ΗΥ
Τομέας Πληροφορικής, Μαθηματικών και Στατιστικής
ΓΕΩΠΟΝΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Εισαγωγή στην Επιστήμη των ΗΥ
Μάθημα-4
url: http://openeclass.aua.gr (AOA101)

Θ. ΤΣΙΛΙΓΚΙΡΙΔΗΣ
ΚΑΘΗΓΗΤΗΣ
ΔΙΚΤΥΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Λογισμικό Υπολογιστή

- ΕΝΟΤΗΤΑ 4.1: Εισαγωγή στα Λειτουργικά Συστήματα (2ΔΩ)
- ΕΝΟΤΗΤΑ 4.2: Ανάπτυξη Λογισμικού (2ΔΩ)
- ΕΝΟΤΗΤΑ 4.3: Γλώσσες Προγραμματισμού (1ΔΩ)
- ΕΝΟΤΗΤΑ 4.4: Αλγόριθμοι - Παραδείγματα (2ΔΩ) – Φροντιστήριο (*)

Διαδικαστικός Προγραμματικός: Αλγόριθμοι

- Αλγόριθμος – Ορισμοί.
- Δομές αλγορίθμων.
- Διαγράμματα ροής.
- Ψευδοκώδικας.
- Κώδικας.
- Δομημένες ροές.
- Τμηματοποίηση – υποαλγόριθμοι.
- Παραδείγματα –Προβλήματα.

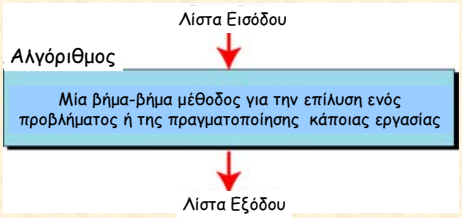
Στόχοι

Στο μάθημα αυτό:

- Θα κατανοήσουμε τι κάνει ένας αλγόριθμος
- Θα ορίσουμε και χρησιμοποιήσουμε δομές που αναπτύσσουν αλγόριθμους, όπως είναι η διαδοχή, η απόφαση και η επανάληψη
- Θα κατανοήσουμε τη χρήση τριών εργαλείων με τα οποία αντιπροσωπεύουμε τους αλγορίθμους, όπως είναι τα διαγράμματα ροής, οι ψευδοκώδικες και οι δομημένες ροές
- Κατανοήσουμε τη σημασία της τμηματοποίησης και των υπο-αλγορίθμων
- Εργαστούμε με κάποιους γνωστούς αλγόριθμους

Τι είναι αλγόριθμος

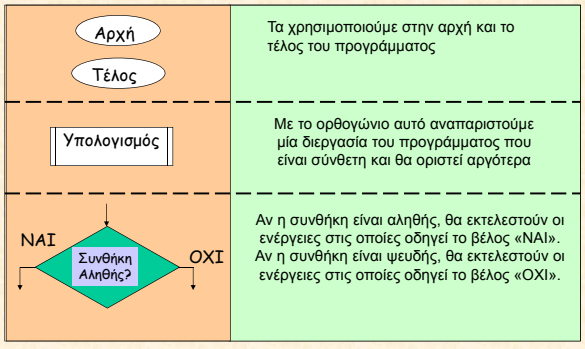
Αλγόριθμος είναι ένας πεπερασμένος αριθμός βημάτων ο οποίος λύνει ένα πρόβλημα.



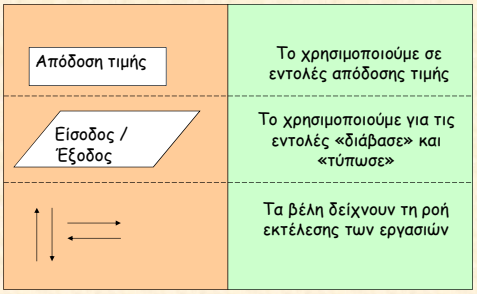
Περιγραφή των βημάτων ενός αλγορίθμου

1. Σε φυσική Γλώσσα
2. Με διάγραμμα ροής
3. Με ψευδοκώδικα
4. Με κώδικα

Διάγραμμα ροής



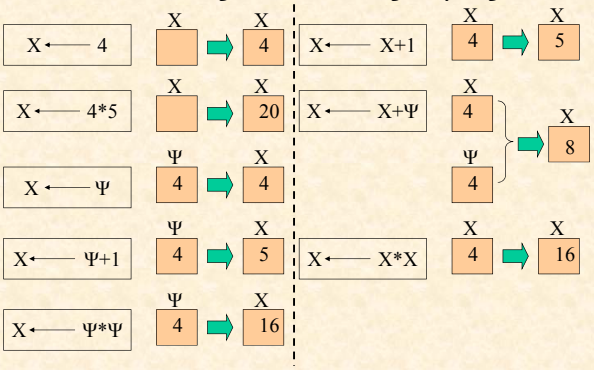
Διάγραμμα ροής



Βασικές κατηγορίες εντολών

- Εντολές Εισόδου/Εξόδου
- Εντολές απόδοσης τιμής
- Εντολές διακλάδωσης
- Εντολές επανάληψης

Εντολές απόδοσης τιμής

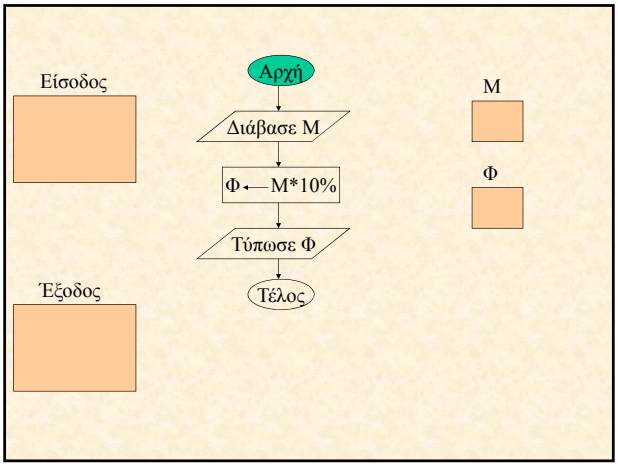


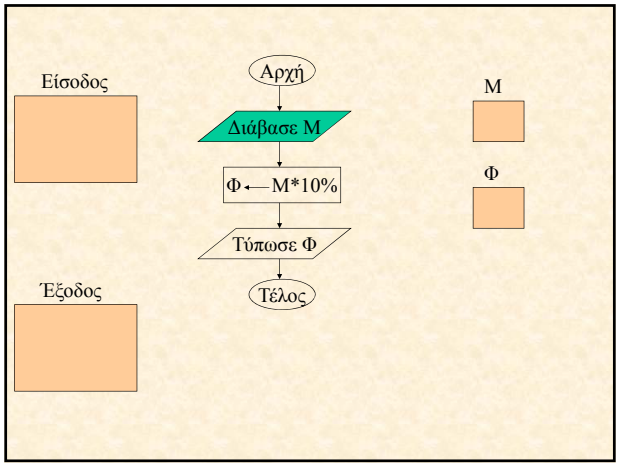
Παράδειγμα

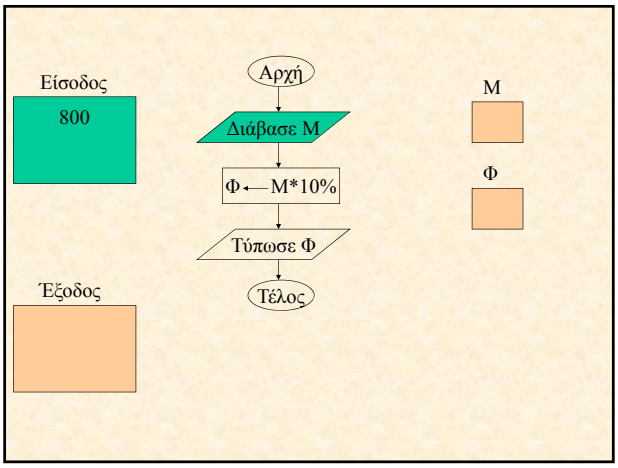
- Ο φόρος ενός εργαζομένου είναι ίσος με το 10% του μισθού του. Να γραφεί πρόγραμμα που δέχεται σαν είσοδο το μισθό ενός εργαζομένου, και εμφανίζει στην έξοδο το φόρο που πρέπει αυτός να πληρώσει.

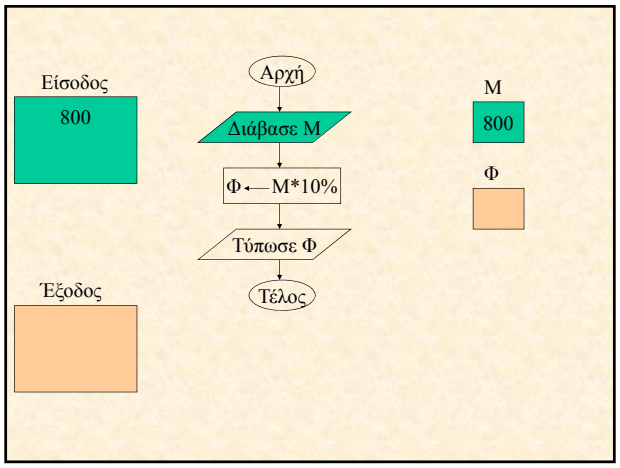
Ο αλγόριθμος σε φυσική γλώσσα

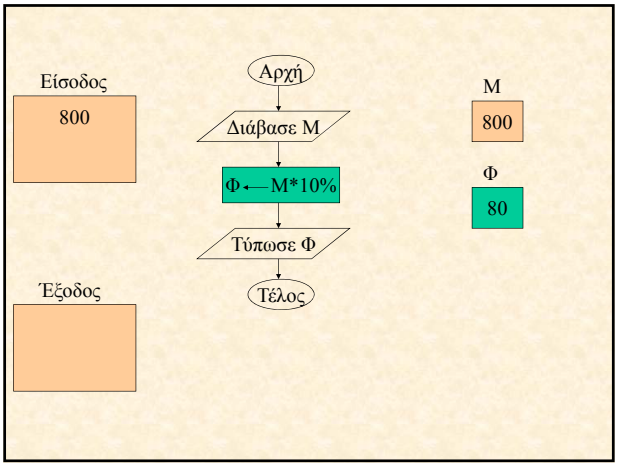
- Πολλαπλασιάζουμε τον αριθμό της εισόδου επί 10, αυτό που βρίσκουμε το διαιρούμε δια 100 και και τυπώνουμε στην έξοδο το αποτέλεσμα της διαίρεσης

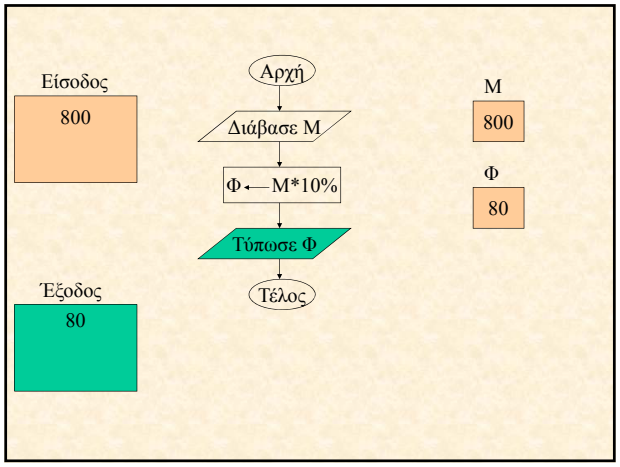

Είσοδος
Αρχή
Διάβασε Μ
Φ ← Μ*10%
Τύπωσε Φ
Τέλος
Έξοδος
Μ
Φ

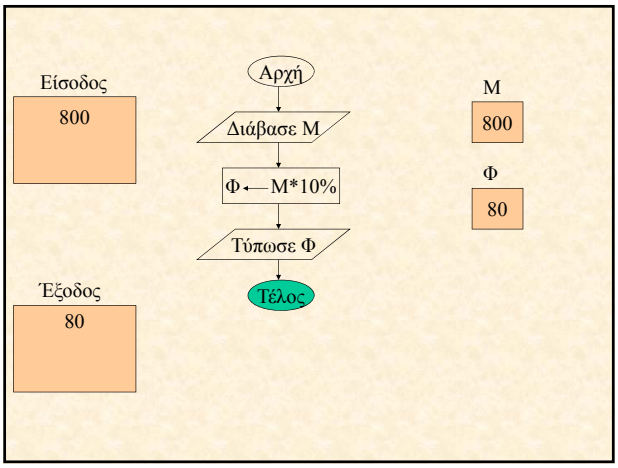

Είσοδος
Αρχή
Διάβασε Μ
Φ ← Μ*10%
Τύπωσε Φ
Τέλος
Έξοδος
Μ
Φ

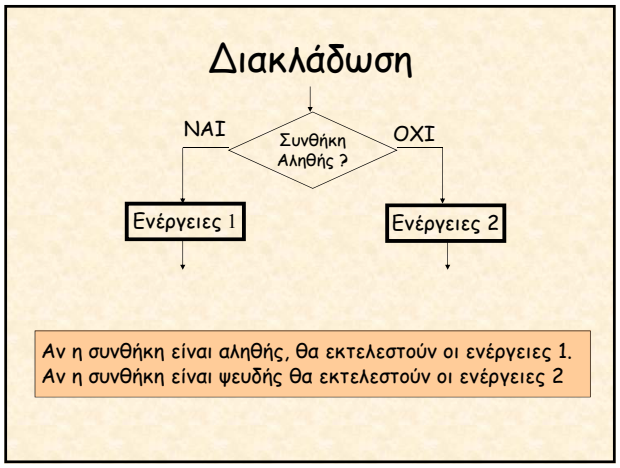

Είσοδος
800
Αρχή
Διάβασε Μ
Φ ← Μ*10%
Τύπωσε Φ
Τέλος
Έξοδος
Μ
Φ


Είσοδος
800
Αρχή
Διάβασε Μ
Φ ← Μ*10%
Τύπωσε Φ
Τέλος
Έξοδος
Μ
800
Φ


Είσοδος
800
Αρχή
Διάβασε Μ
Φ←Μ*10%
Τύπωσε Φ
Τέλος
Έξοδος
Μ
800
Φ
80


Είσοδος
800
Αρχή
Διάβασε Μ
Φ←Μ*10%
Τύπωσε Φ
Τέλος
Έξοδος
80
Μ
800
Φ
80


Είσοδος
800
Αρχή
Διάβασε Μ
Φ←Μ*10%
Τύπωσε Φ
Τέλος
Έξοδος
80
Μ
800
Φ
80


Διακλάδωση
ΝΑΙ
Συνθήκη Αληθής ?
ΟΧΙ
Ενέργειες 1
Ενέργειες 2
Αν η συνθήκη είναι αληθής, θα εκτελεστούν οι ενέργειες 1.
Αν η συνθήκη είναι ψευδής θα εκτελεστούν οι ενέργειες 2

Ένα απλό πρόβλημα

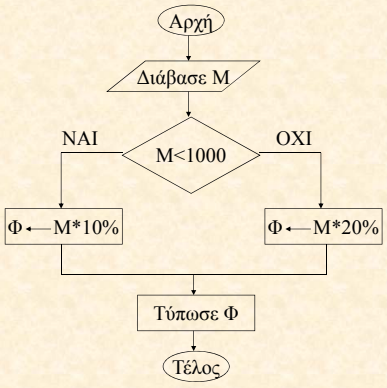
Έστω ότι ο φόρος ενός εργαζομένου προκύπτει ως εξής: Αν ο μισθός του είναι κάτω από 1000€, ο φόρος του είναι 10% του μισθού του. Αλλιώς, ο φόρος του είναι 20% του μισθού του.

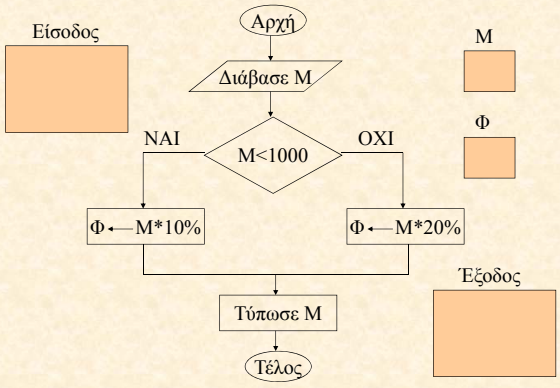
Γράψτε ένα πρόγραμμα που δέχεται σαν είσοδο το μισθό ενός εργαζομένου, και εμφανίζει στην έξοδο το φόρο που πρέπει αυτός να πληρώσει.

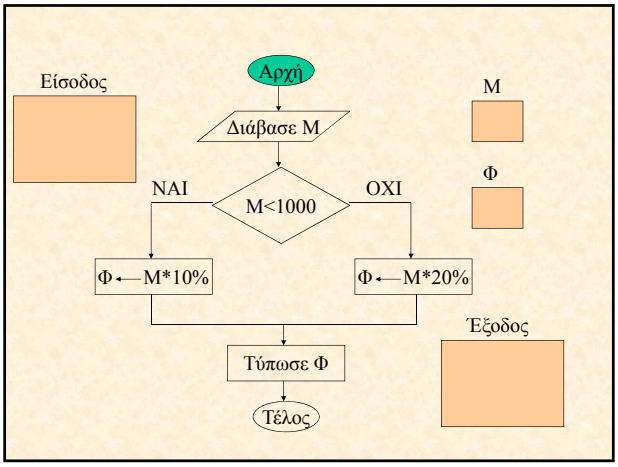
Ο αλγόριθμος σε φυσική γλώσσα

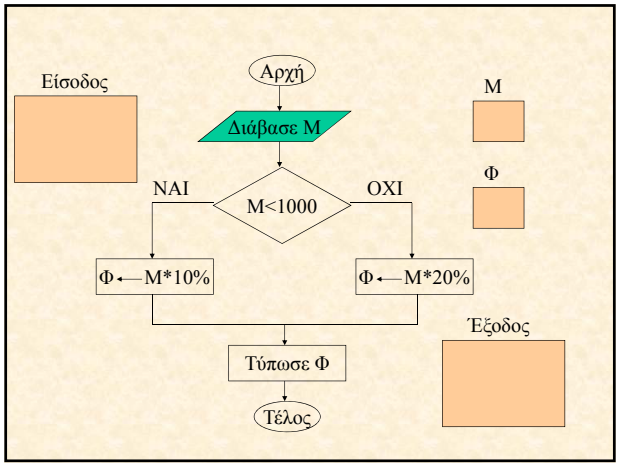
Αν ο μισθός της εισόδου είναι μικρότερος από 1000, τον πολλαπλασιάζουμε επί 10, διαιρούμε το γινόμενο επί 100 και τυπώνουμε το αποτέλεσμα. Αλλιώς, πολλαπλασιάζουμε τον αριθμό της εισόδου επί 20, διαιρούμε το γινόμενο επί 100 και τυπώνουμε το αποτέλεσμα

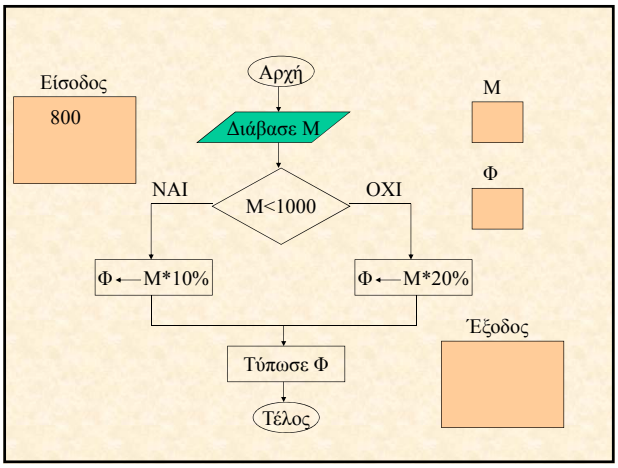
Διάγραμμα ροής

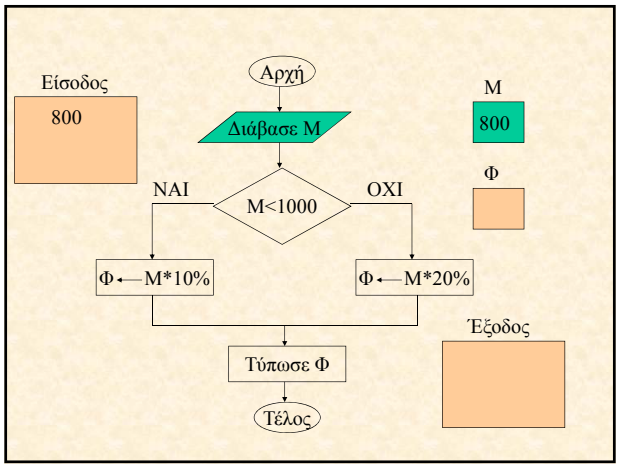


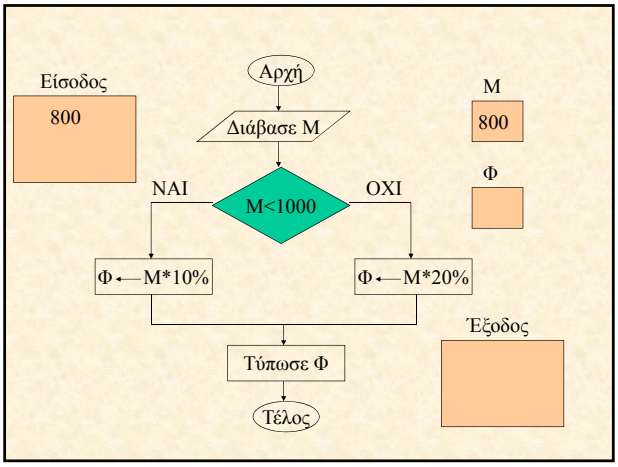


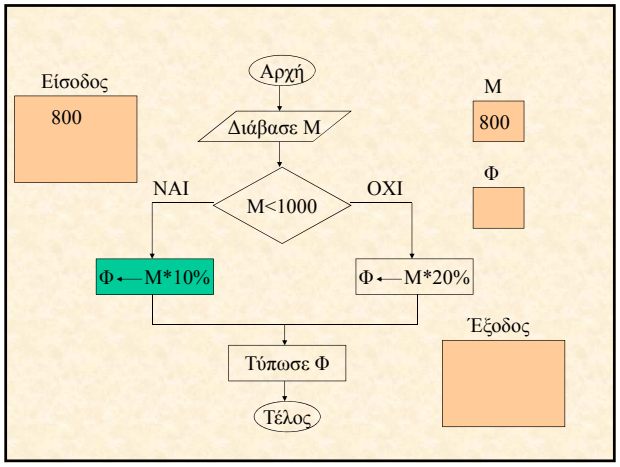

Είσοδος
Αρχή
Διάβασε Μ
Μ
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ←Μ*10%
Φ←Μ*20%
Έξοδος
Τύπωσε Φ
Τέλος

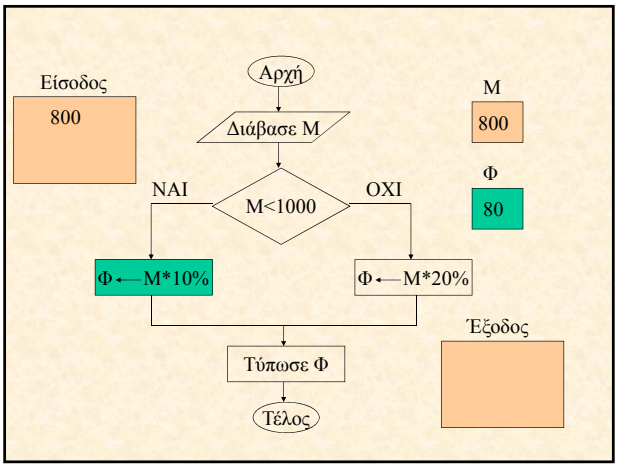

Είσοδος
Αρχή
Διάβασε Μ
Μ
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ←Μ*10%
Φ←Μ*20%
Έξοδος
Τύπωσε Φ
Τέλος

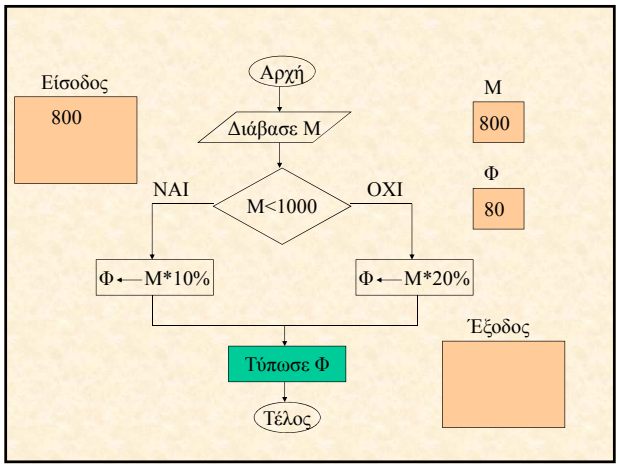

Είσοδος
800
Αρχή
Διάβασε Μ
Μ
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ←Μ*10%
Φ←Μ*20%
Έξοδος
Τύπωσε Φ
Τέλος

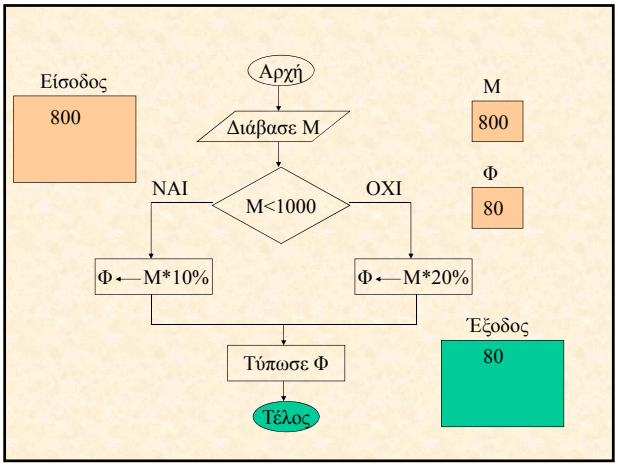

Είσοδος
800
Αρχή
Διάβασε Μ
Μ
800
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ←Μ*10%
Φ←Μ*20%
Έξοδος
Τύπωσε Φ
Τέλος


Είσοδος
800
Αρχή
Διάβασε Μ
Μ
800
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ←Μ*10%
Φ←Μ*20%
Έξοδος
Τύπωσε Φ
Τέλος


Είσοδος
800
Αρχή
Διάβασε Μ
Μ
800
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ←Μ*10%
Φ←Μ*20%
Έξοδος
Τύπωσε Φ
Τέλος


Είσοδος
800
Αρχή
Διάβασε Μ
Μ
800
Φ
80
ΝΑΙ
Μ<1000
ΟΧΙ
Φ←Μ*10%
Φ←Μ*20%
Έξοδος
Τύπωσε Φ
Τέλος

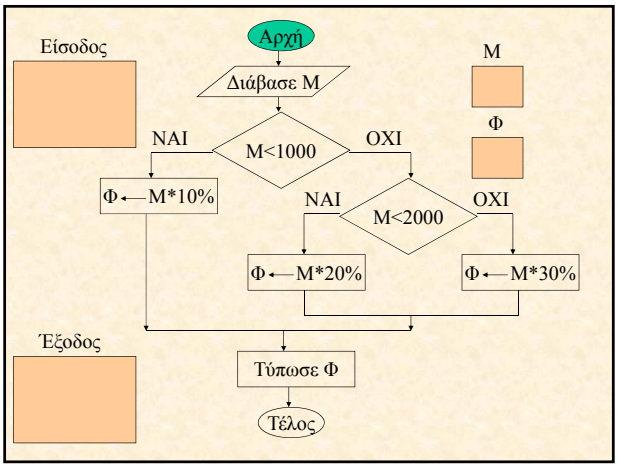

Είσοδος
800
Αρχή
Διάβασε Μ
Μ
800
Φ
80
ΝΑΙ
Μ<1000
ΟΧΙ
Φ←Μ*10%
Φ←Μ*20%
Έξοδος
Τύπωσε Φ
Τέλος

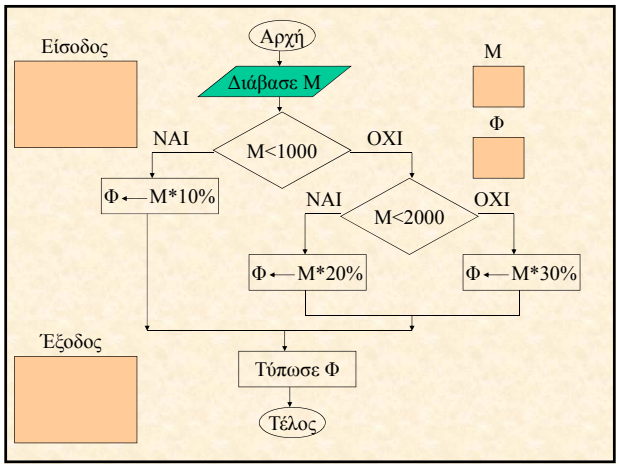


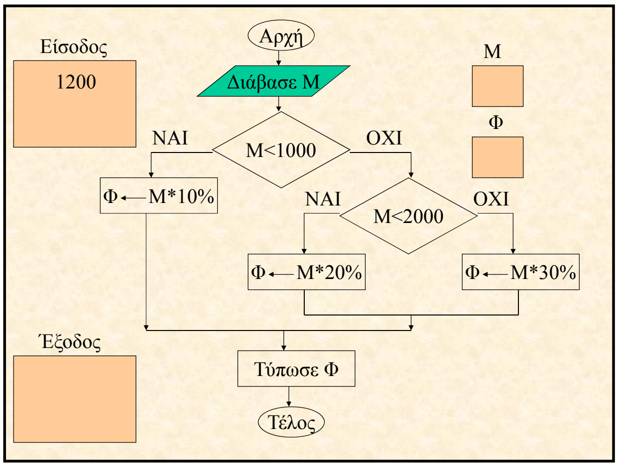
Ένα πιο δύσκολο πρόβλημα

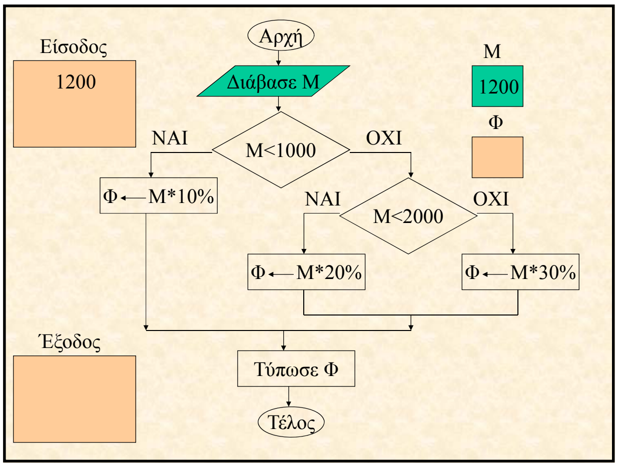
Έστω ότι ο φόρος ενός εργαζομένου προκύπτει ως εξής: Αν ο μισθός του είναι κάτω από 1000€, ο φόρος του είναι 10% του μισθού του. Αν ο μισθός είναι από 1000 € και πάνω και κάτω από 2000 €, ο φόρος του είναι 20% του μισθού του. Αλλιώς, ο φόρος του είναι 30% του μισθού του.

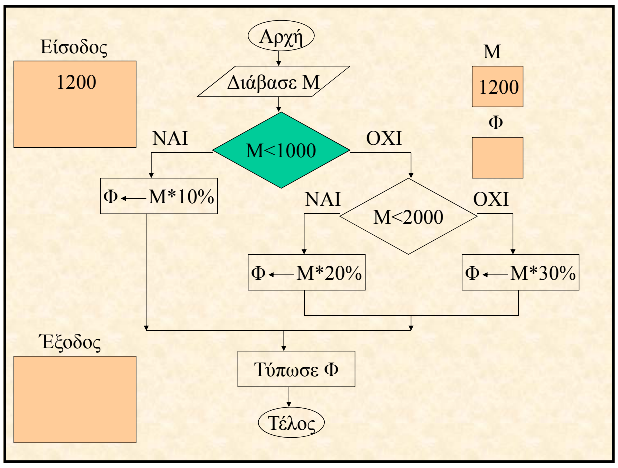
Γράψτε ένα πρόγραμμα που δέχεται σαν είσοδο το μισθό ενός εργαζομένου, και εμφανίζει στην έξοδο το φόρο που πρέπει αυτός να πληρώσει.

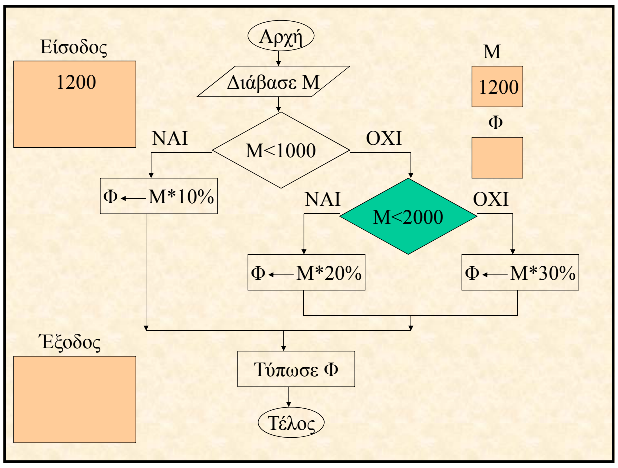


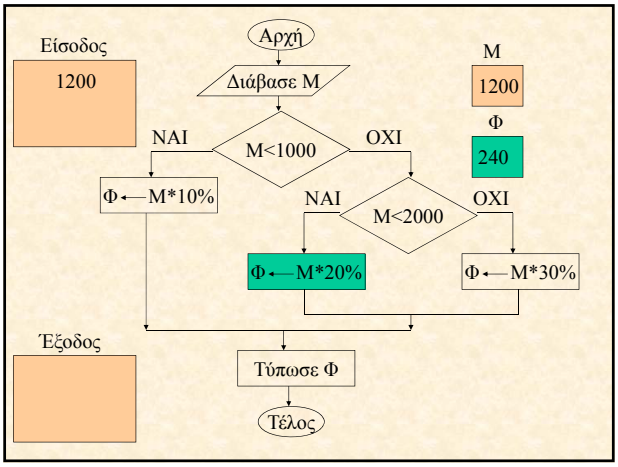


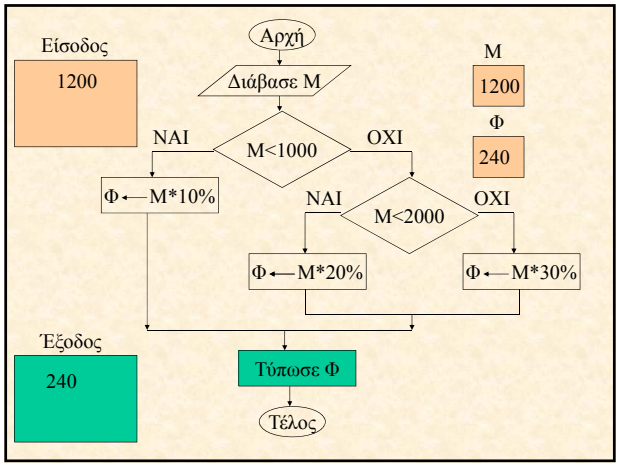

Είσοδος
1200
Αρχή
Διάβασε Μ
Μ
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ ← Μ*10%
ΝΑΙ
Μ<2000
ΟΧΙ
Φ ← Μ*20%
Φ ← Μ*30%
Έξοδος
Τύπωσε Φ
Τέλος

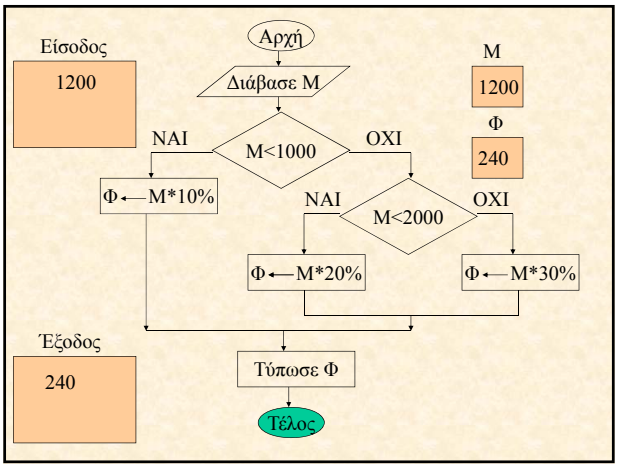

Είσοδος
1200
Αρχή
Διάβασε Μ
Μ
1200
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ ← Μ*10%
ΝΑΙ
Μ<2000
ΟΧΙ
Φ ← Μ*20%
Φ ← Μ*30%
Έξοδος
Τύπωσε Φ
Τέλος


Είσοδος
1200
Αρχή
Διάβασε Μ
Μ
1200
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ ← Μ*10%
ΝΑΙ
Μ<2000
ΟΧΙ
Φ ← Μ*20%
Φ ← Μ*30%
Έξοδος
Τύπωσε Φ
Τέλος


Είσοδος
1200
Αρχή
Διάβασε Μ
Μ
1200
Φ
ΝΑΙ
Μ<1000
ΟΧΙ
Φ ← Μ*10%
ΝΑΙ
Μ<2000
ΟΧΙ
Φ ← Μ*20%
Φ ← Μ*30%
Έξοδος
Τύπωσε Φ
Τέλος



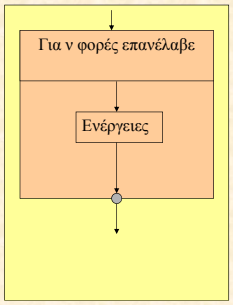




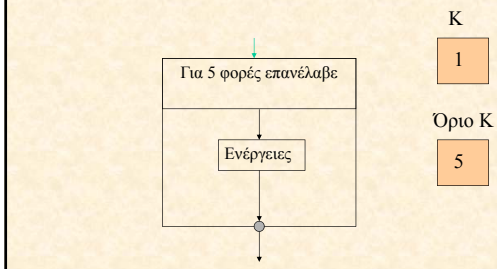
Επανάληψη

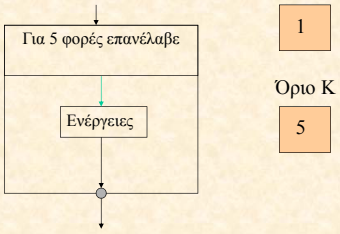
- Για ν φορές επανέλαβε
- Εφόσον η συνθήκη είναι αληθής επανέλαβε
- Επανέλαβε όσο η συνθήκη να γίνει αληθής

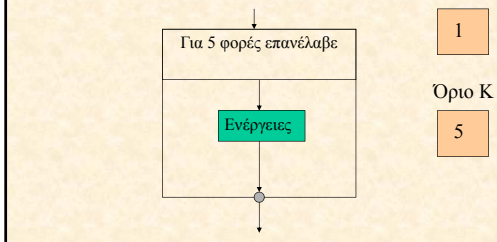
Για ν φορές επανέλαβε

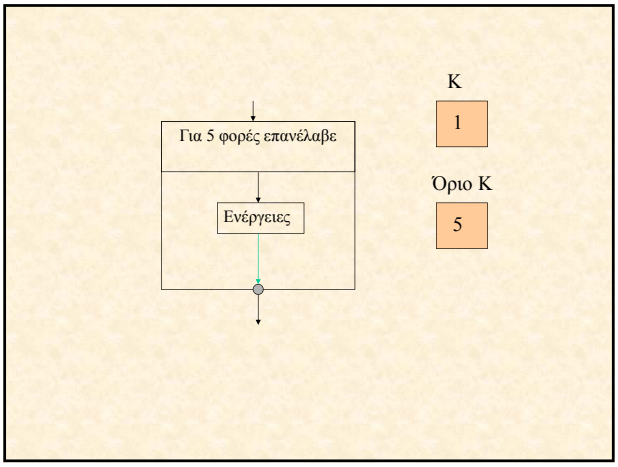


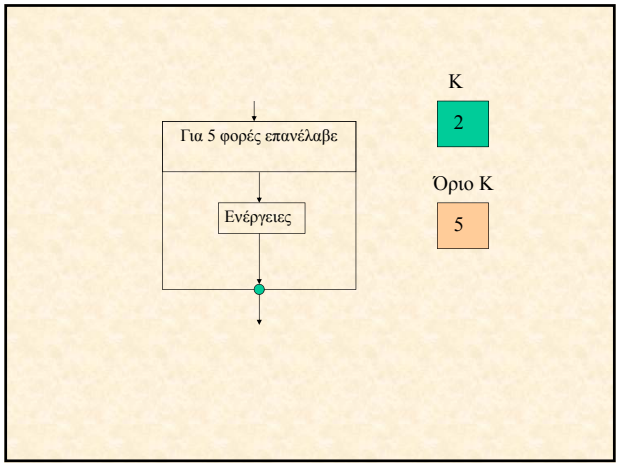
Σε αυτή τη μορφή επανάληψης, οι «ενέργειες» θα εκτελεστούν γνωστό αριθμό φορών, και για το συγκεκριμένο παράδειγμα ν φορές. Η μέτρηση των φορών γίνεται με τη βοήθεια μίας μεταβλητής, που παίρνει μία αρχική τιμή πριν εκτελεστούν για πρώτη φορά οι «ενέργειες» και η τιμή της αλλάζει κάθε φορά που τελειώνει η εκτέλεση των «ενεργειών».

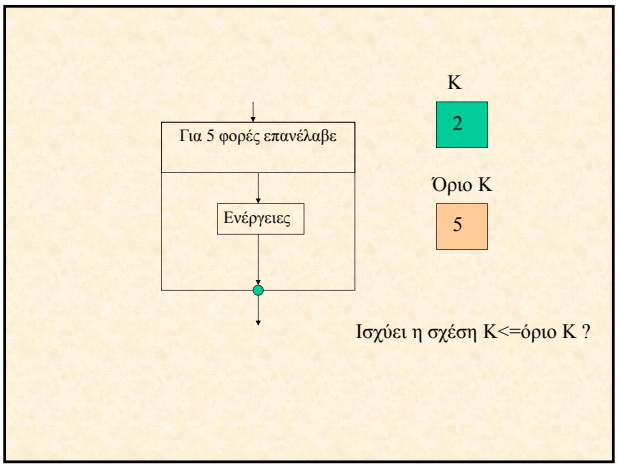


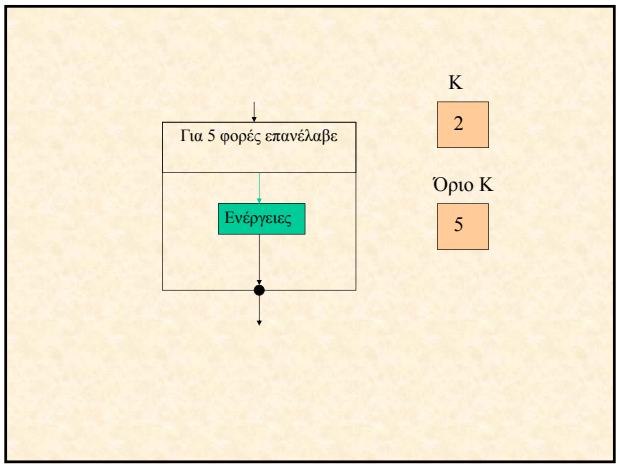


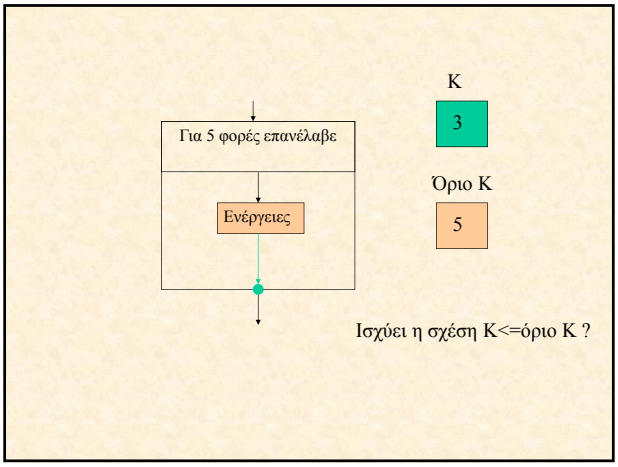


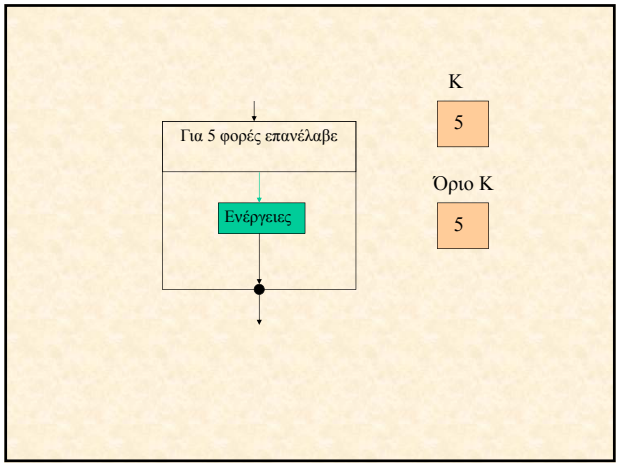

Για 5 φορές επανέλαβε
Ενέργειες
K
1
Όριο K
5

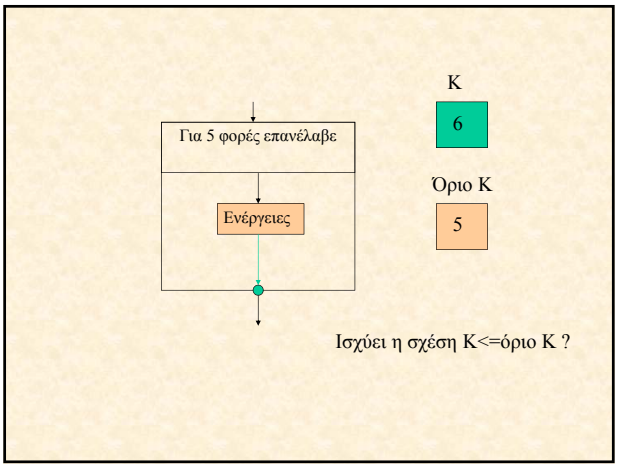

Για 5 φορές επανέλαβε
Ενέργειες
K
2
Όριο K
5

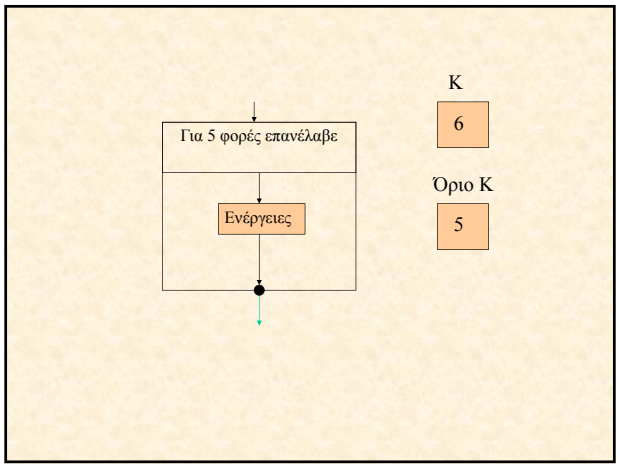

Για 5 φορές επανέλαβε
Ενέργειες
K
2
Όριο K
5
Ισχύει η σχέση K<=όριο K ?


Για 5 φορές επανέλαβε
Ενέργειες
K
2
Όριο K
5


Για 5 φορές επανέλαβε
Ενέργειες
K
3
Όριο K
5
Ισχύει η σχέση K<=όριο K ?


Για 5 φορές επανέλαβε
Ενέργειες
K
5
Όριο K
5


Για 5 φορές επανέλαβε
Ενέργειες
K
6
Όριο K
5
Ισχύει η σχέση K<=όριο K ?

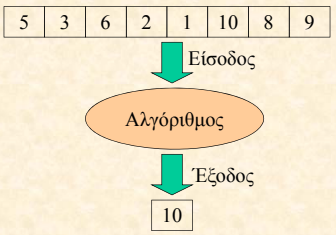

Για 5 φορές επανέλαβε
Ενέργειες
K
6
Όριο K
5

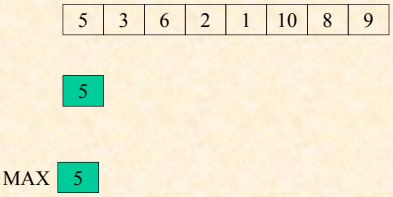
Παράδειγμα 1

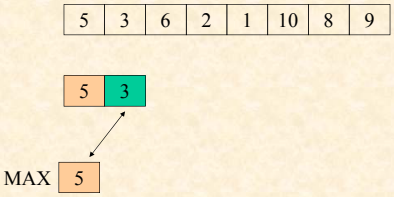
Να γραφεί πρόγραμμα το οποίο δέχεται σαν είσοδο 10 αριθμούς. Το πρόγραμμα εμφανίζει στην έξοδο το μεγαλύτερο από τους αριθμούς που δόθηκαν

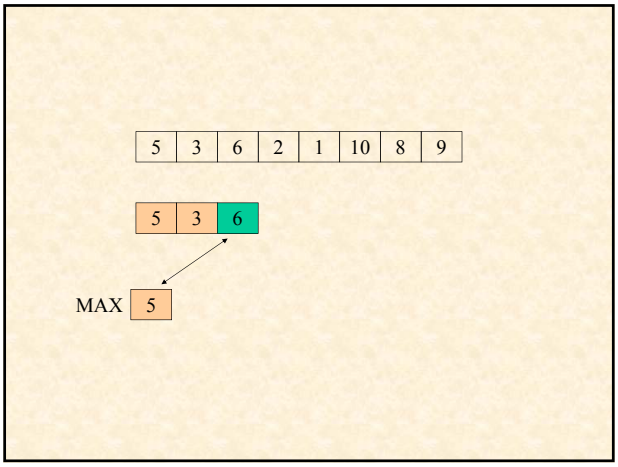
Αλγόριθμος εύρεσης Μέγιστου

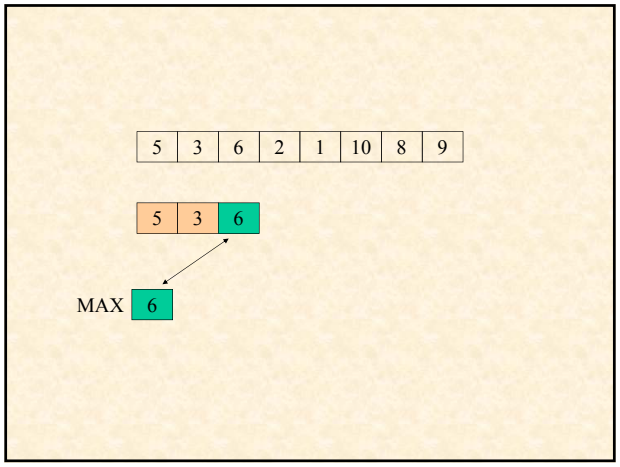
Παράδειγμα

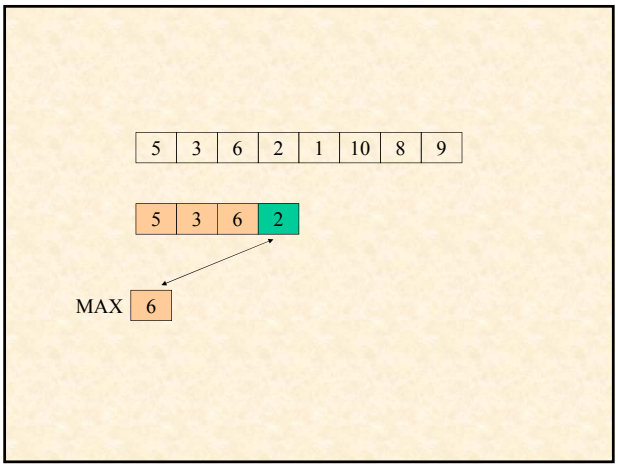


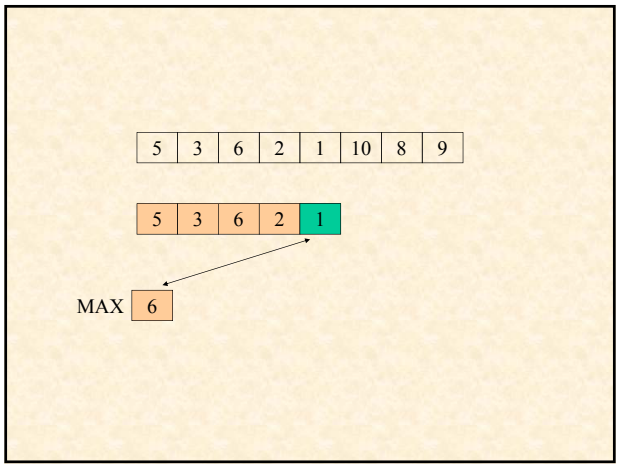


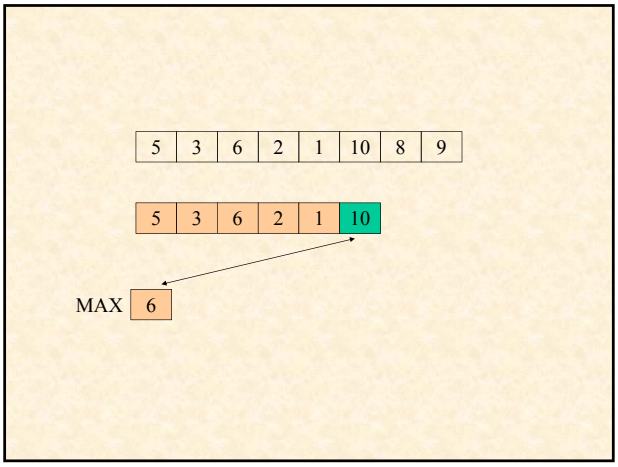


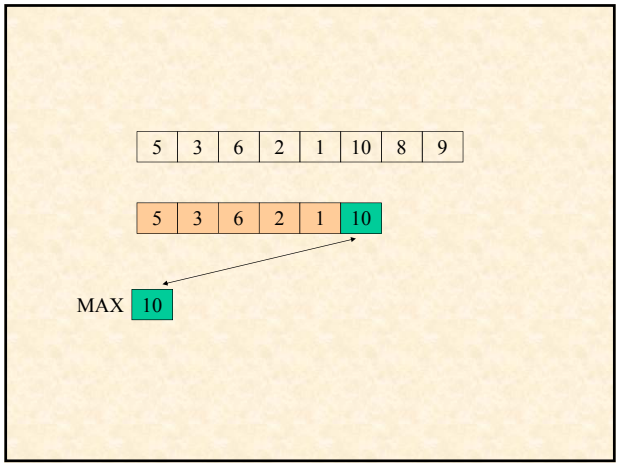

5 3 6 2 1 10 8 9
5 3 6
MAX 5

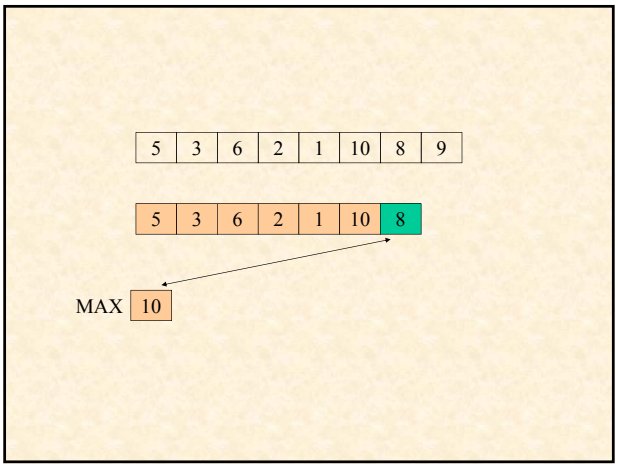

5 3 6 2 1 10 8 9
5 3 6
MAX 6

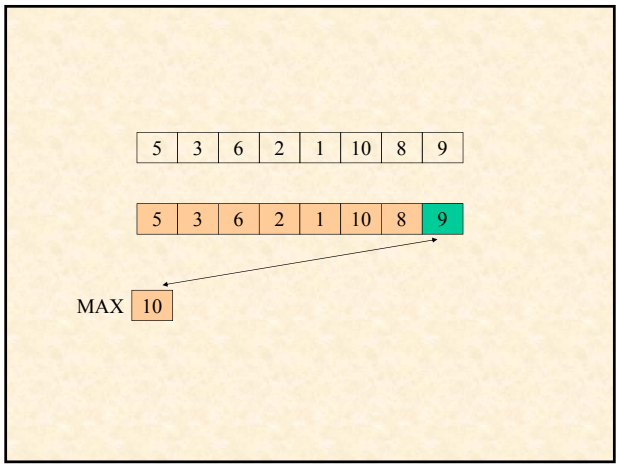

5 3 6 2 1 10 8 9
5 3 6 2
MAX 6


5 3 6 2 1 10 8 9
5 3 6 2 1
MAX 6


5 3 6 2 1 10 8 9
5 3 6 2 1 10
MAX 6

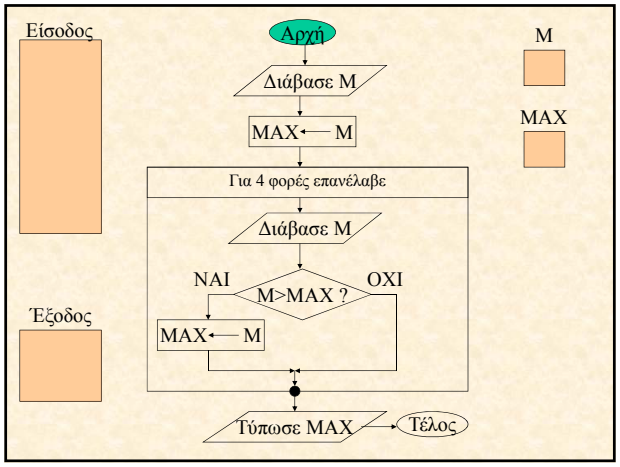

5 3 6 2 1 10 8 9
5 3 6 2 1 10
MAX 10

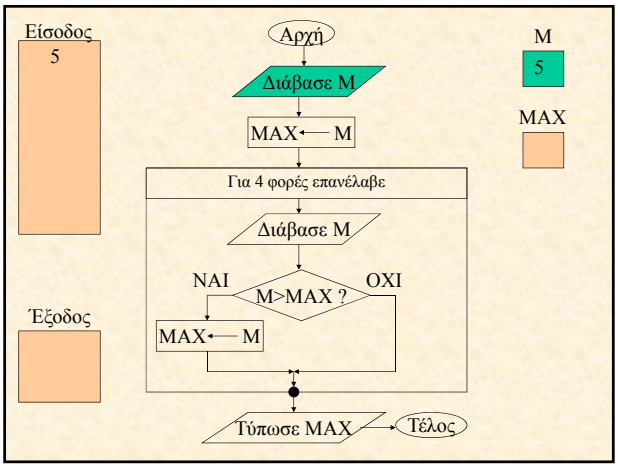

5 3 6 2 1 10 8 9
5 3 6 2 1 10 8
MAX 10

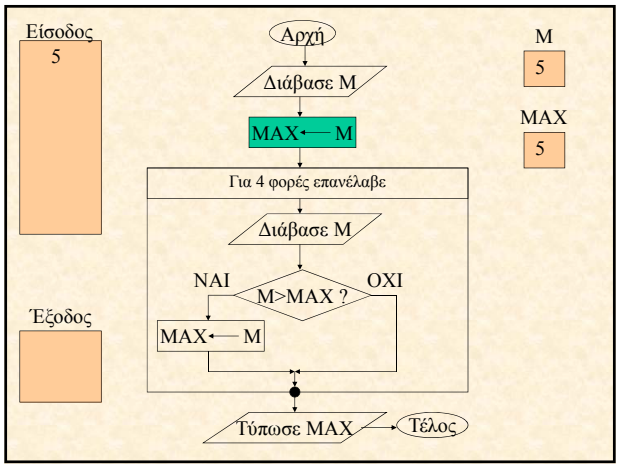

5 3 6 2 1 10 8 9
5 3 6 2 1 10 8 9
MAX 10

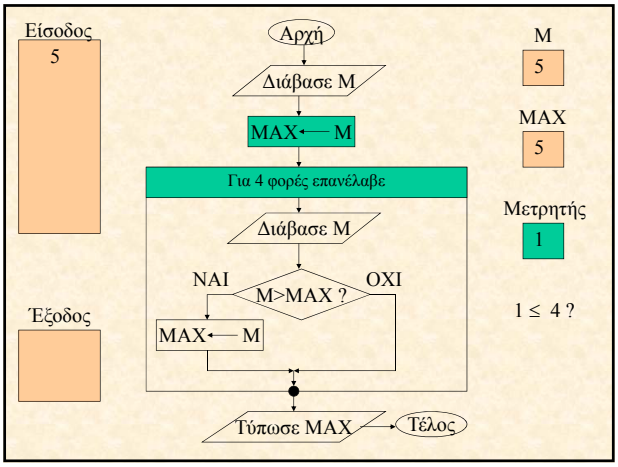
Αλγόριθμος σε φυσική γλώσσα

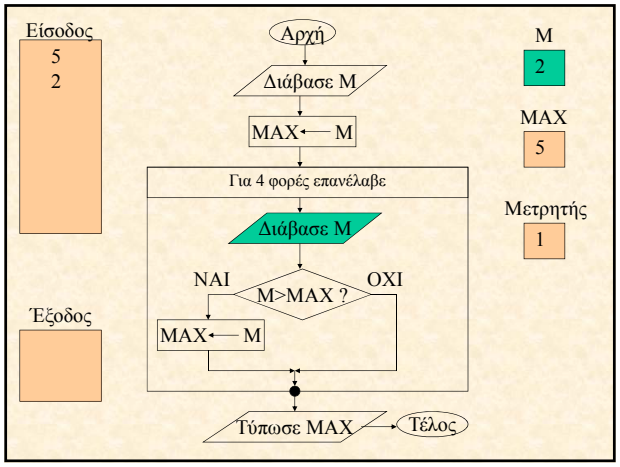
Αποθηκεύουμε τον πρώτο αριθμό σα μέγιστο. Στη συνέχεια, συγκρίνουμε κάθε αριθμό που δίνεται, με το μέγιστο. Αν ο αριθμός είναι μεγαλύτερος από το μέγιστο, αποθηκεύουμε αυτόν στη θέση του μέγιστου.

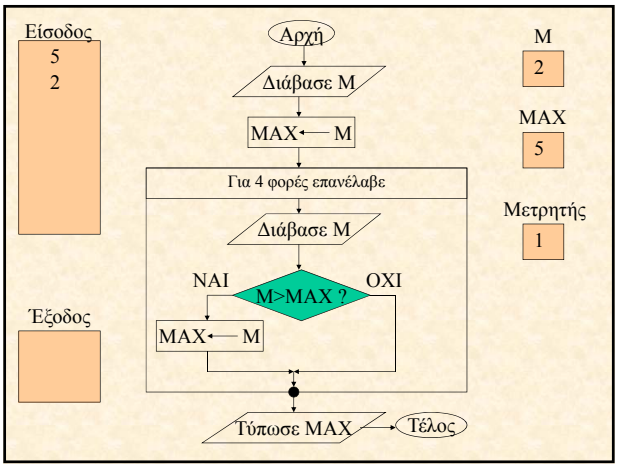


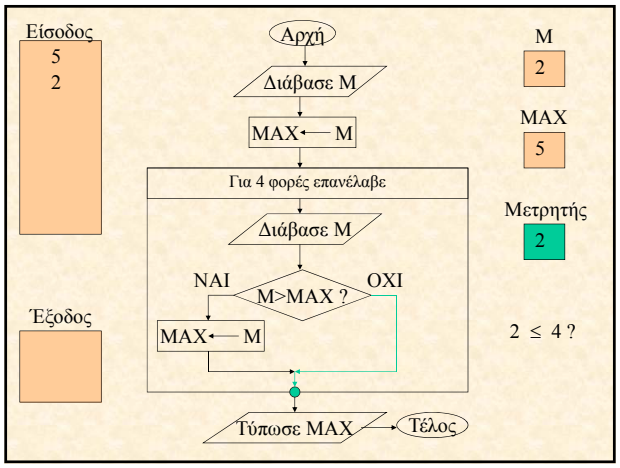


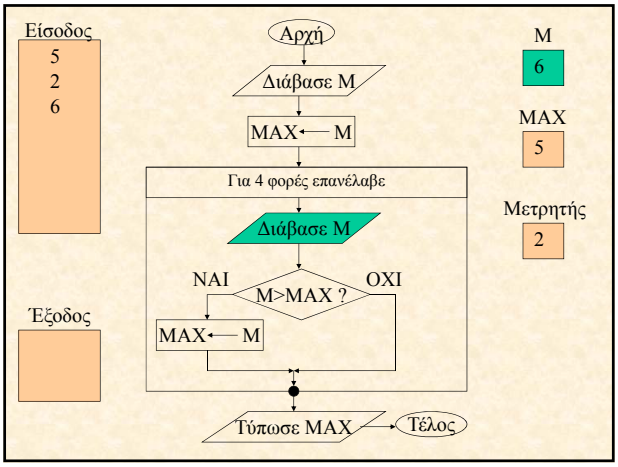


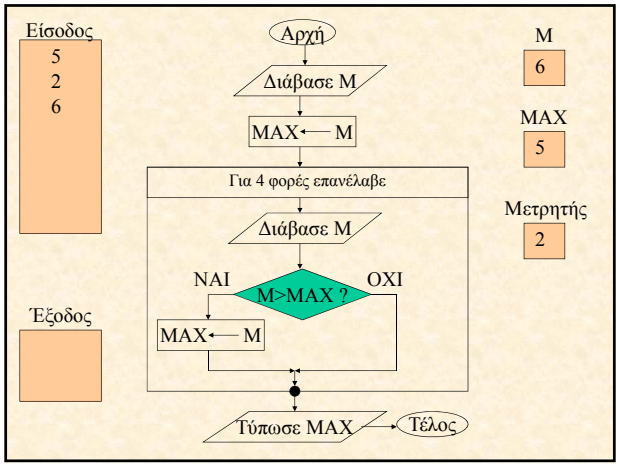

Είσοδος
5
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
5
MAX
5
Μετρητής
1
1 ≤ 4 ?

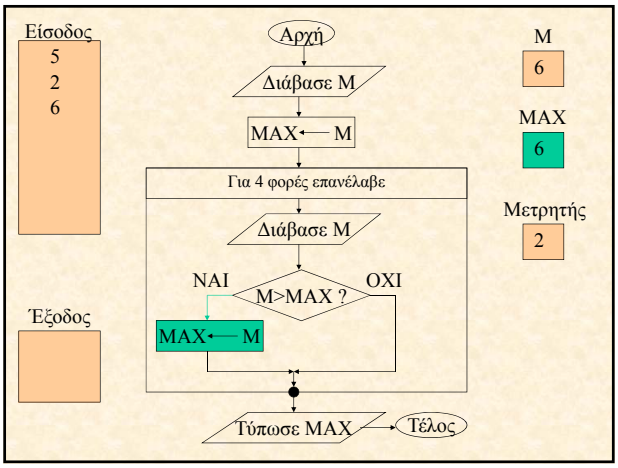

Είσοδος
5
2
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
2
MAX
5
Μετρητής
1

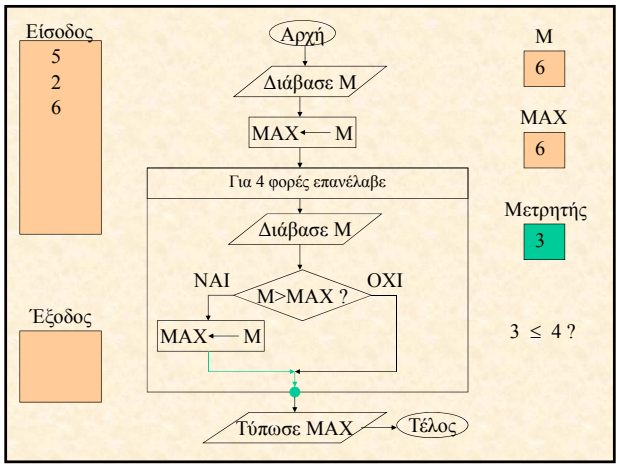

Είσοδος
5
2
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
2
MAX
5
Μετρητής
1

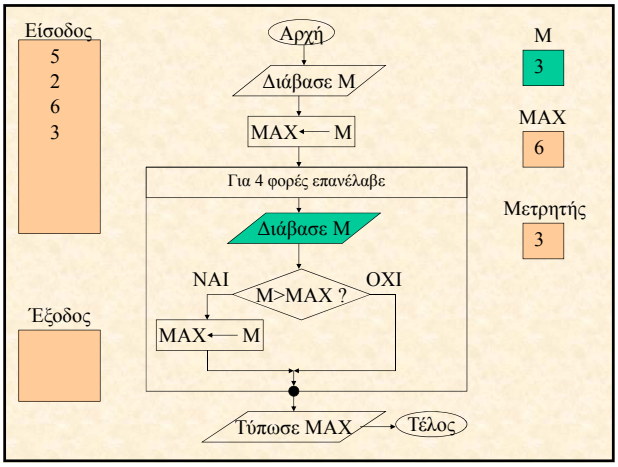

Είσοδος
5
2
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
2
MAX
5
Μετρητής
2
2 ≤ 4 ?

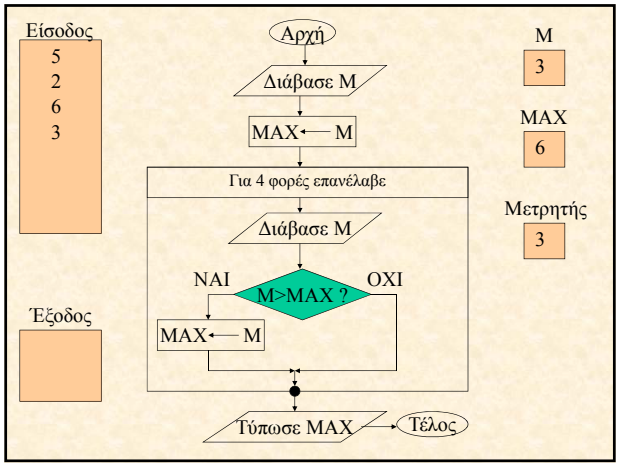

Είσοδος
5
2
6
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
6
MAX
5
Μετρητής
2

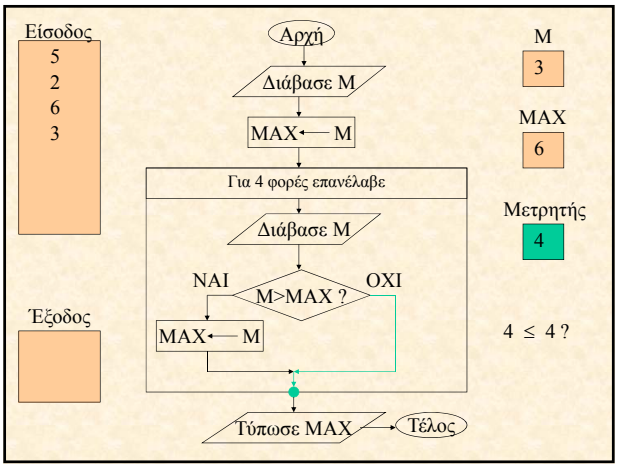

Είσοδος
5
2
6
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
6
MAX
5
Μετρητής
2

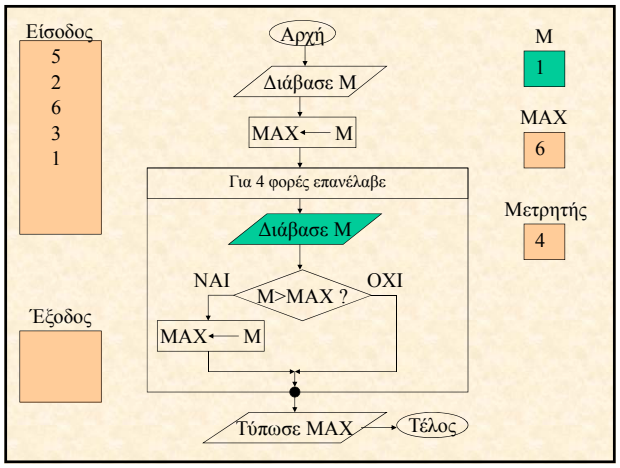

Είσοδος
5
2
6
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
6
MAX
6
Μετρητής
2

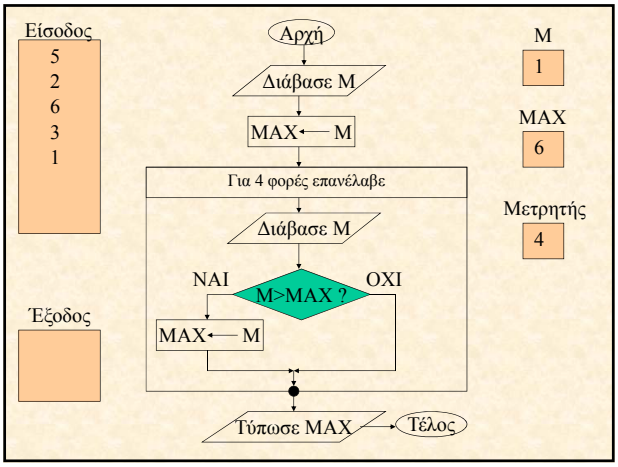

Είσοδος
5
2
6
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
6
MAX
6
Μετρητής
3
3 ≤ 4 ?

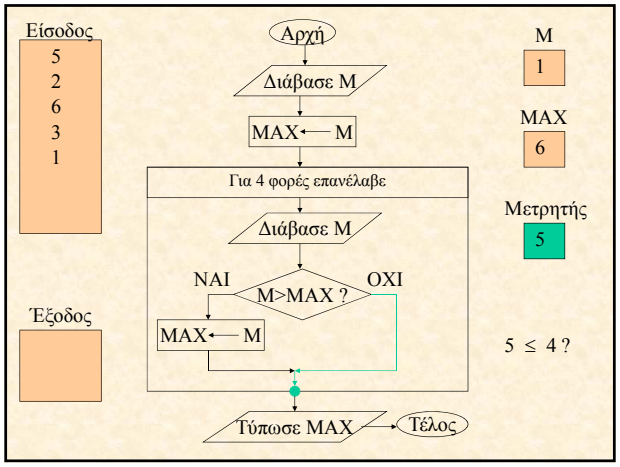

Είσοδος
5
2
6
3
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
3
MAX
6
Μετρητής
3

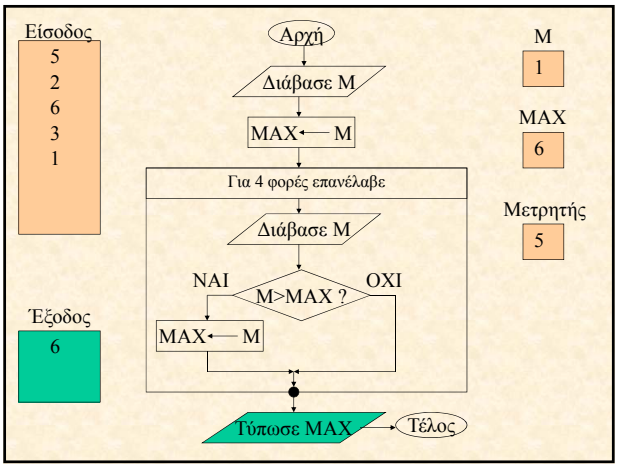

Είσοδος
5
2
6
3
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
3
MAX
6
Μετρητής
3

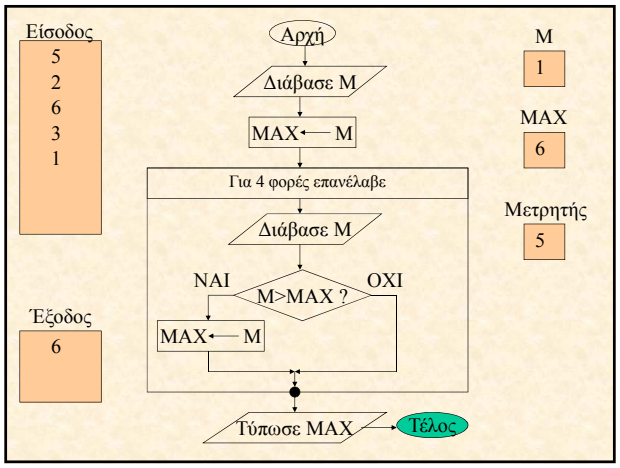

Είσοδος
5
2
6
3
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
3
MAX
6
Μετρητής
4
4 ≤ 4 ?


Είσοδος
5
2
6
3
1
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
1
MAX
6
Μετρητής
4


Είσοδος
5
2
6
3
1
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
1
MAX
6
Μετρητής
4


Είσοδος
5
2
6
3
1
Έξοδος
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
1
MAX
6
Μετρητής
5
5 ≤ 4 ?

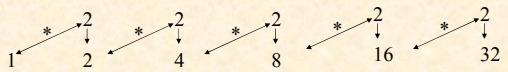

Είσοδος
5
2
6
3
1
Έξοδος
6
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
1
MAX
6
Μετρητής
5

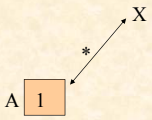

Είσοδος
5
2
6
3
1
Έξοδος
6
Αρχή
Διάβασε Μ
MAX ← Μ
Για 4 φορές επανέλαβε
Διάβασε Μ
ΝΑΙ
Μ>MAX ?
ΟΧΙ
MAX ← Μ
Τύπωσε MAX
Τέλος
Μ
1
MAX
6
Μετρητής
5

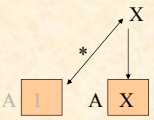
Παράδειγμα 2

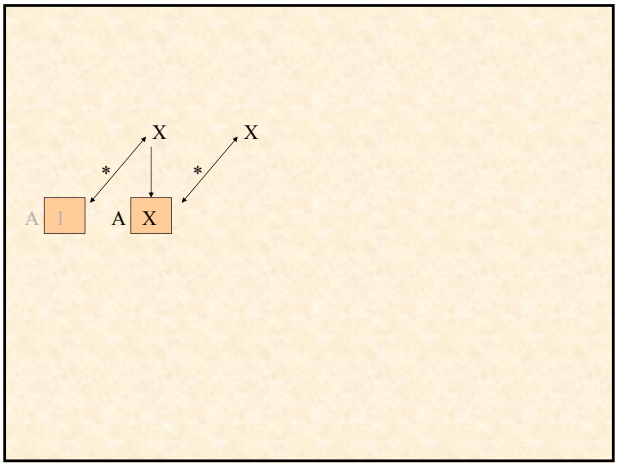
Γράψτε ένα πρόγραμμα που δέχεται σαν είσοδο δύο αριθμούς, έστω χ και ψ, και εμφανίζει στην έξοδο τον αριθμό χ^{ψ}

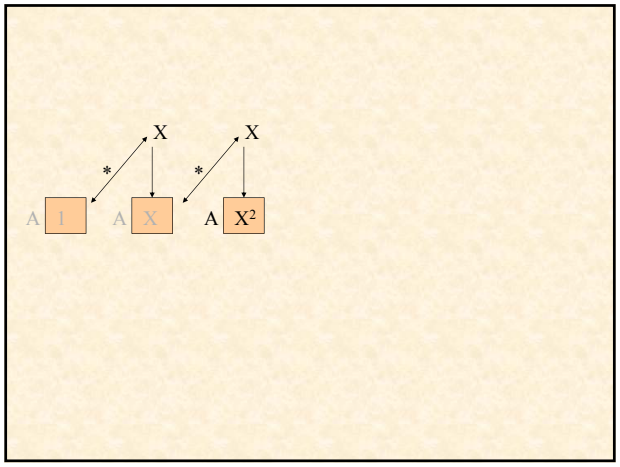
Βάση =2 Εκθέτης = 5

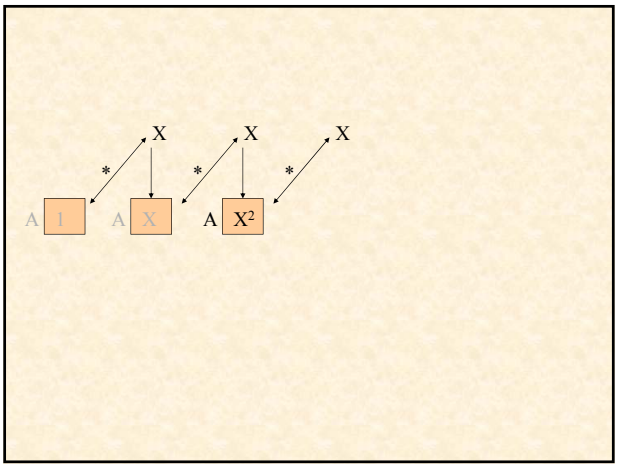


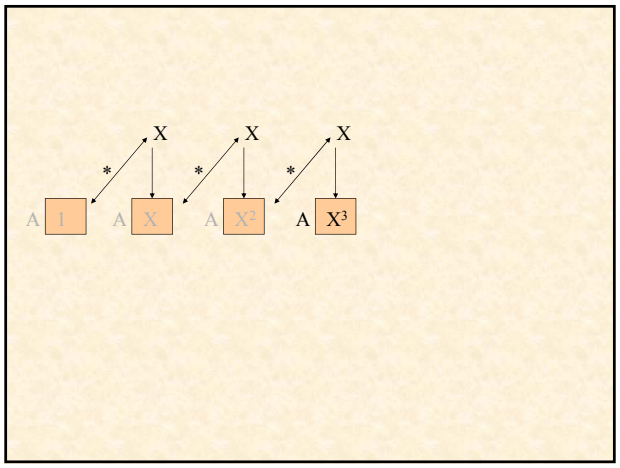


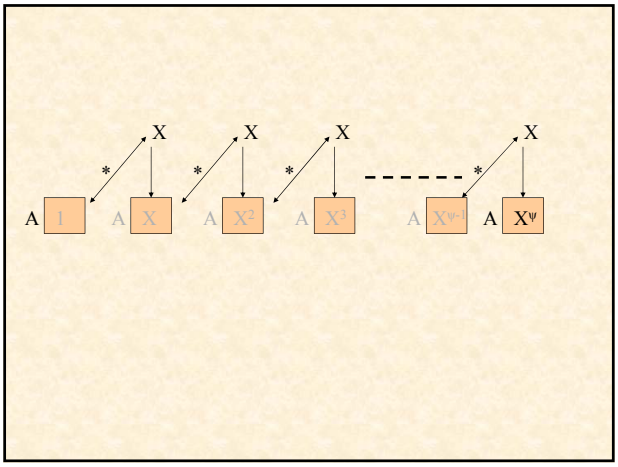


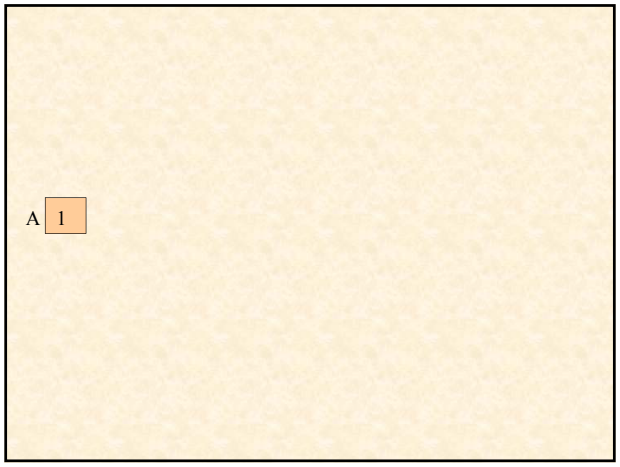

X
X
*
*
A
1
A
X

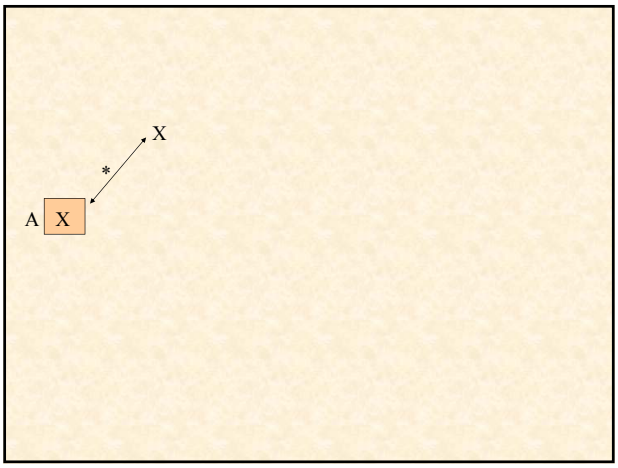

X
X
*
*
A
1
A
X
A
X²

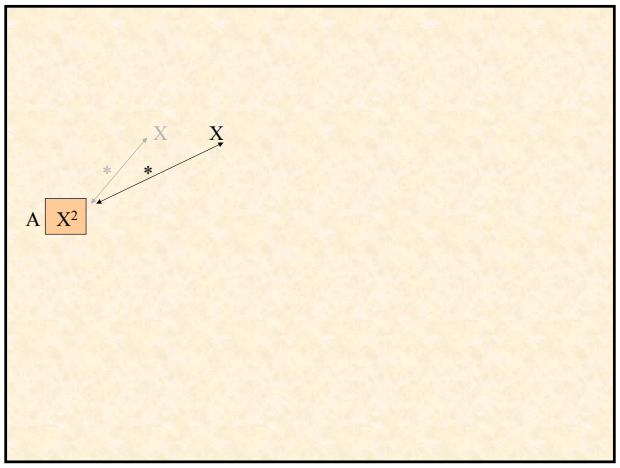

X
X
X
*
*
*
A
1
A
X
A
X²

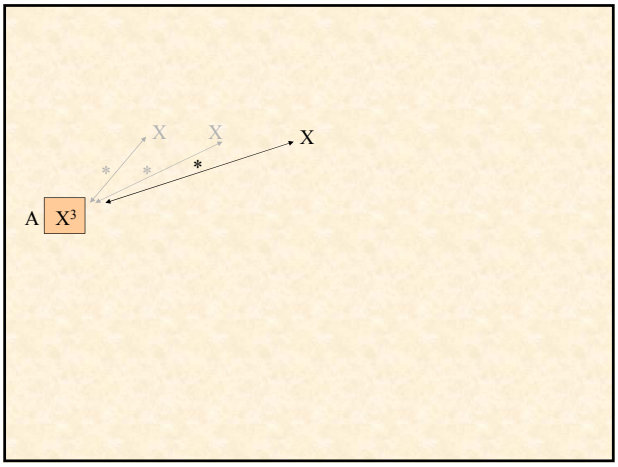

X
X
X
*
*
*
A
1
A
X
A
X²
A
X³

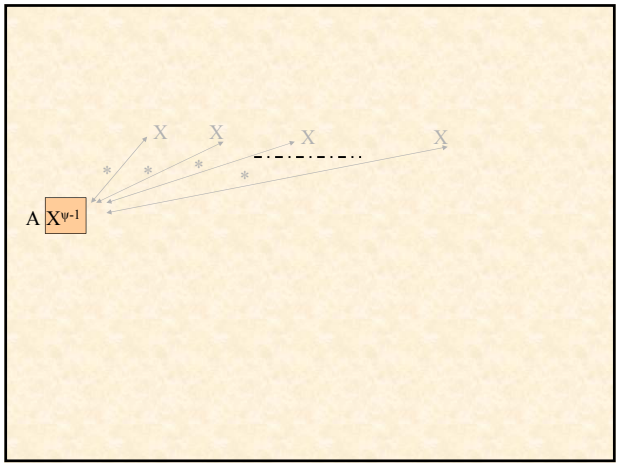

X
X
X
X
*
*
*
*
A 1
A X
A X2
A X3
A Xψ-1
A Xψ

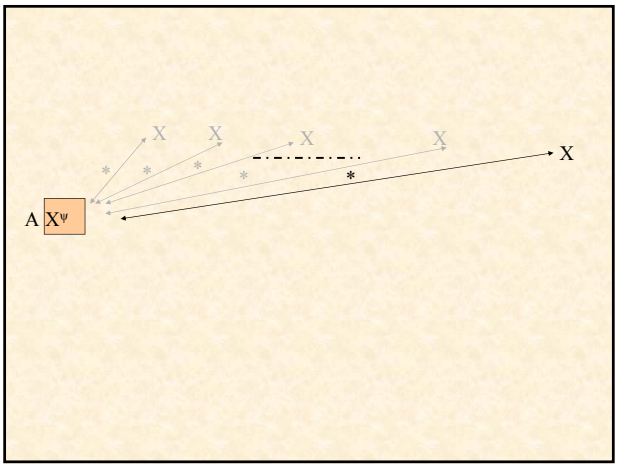

A 1

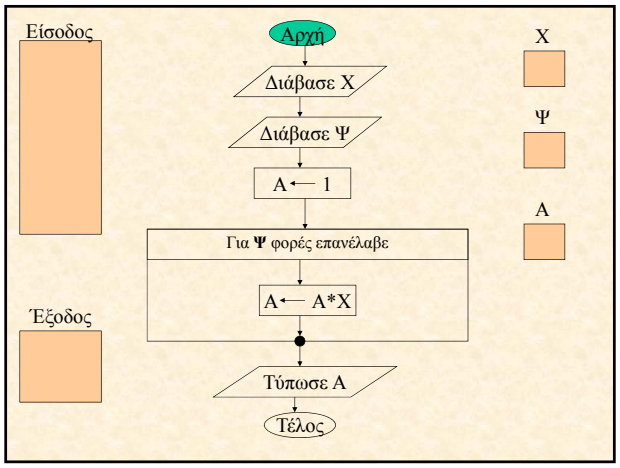

X
*
A X


X
X
*
*
A X2

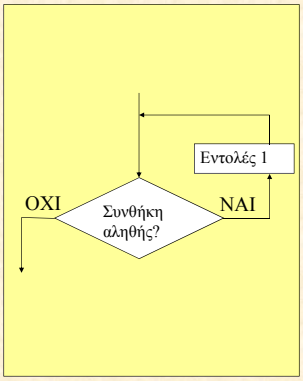

X
X
X
*
A
X³


X
X
X
X
A
Xψ-1


X
X
X
X
X
*
A
Xψ


Είσοδος
Αρχή
Διάβασε Χ
Διάβασε Ψ
Α ← 1
Για Ψ φορές επανέλαβε
Α ← Α*Χ
Έξοδος
Τύπωσε Α
Τέλος
Χ
Ψ
Α

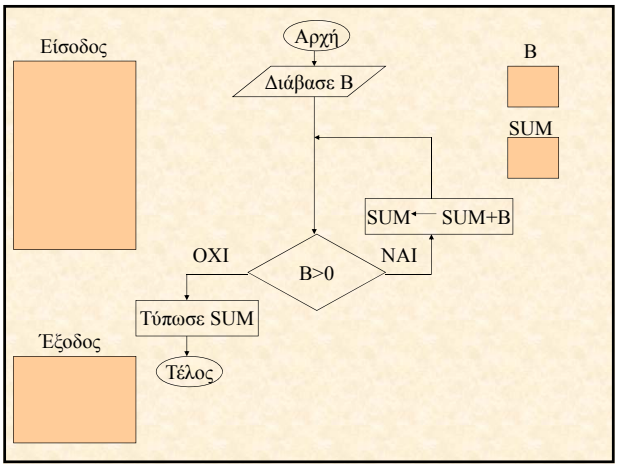
Εφόσον η συνθήκη είναι αληθής επανέλαβε

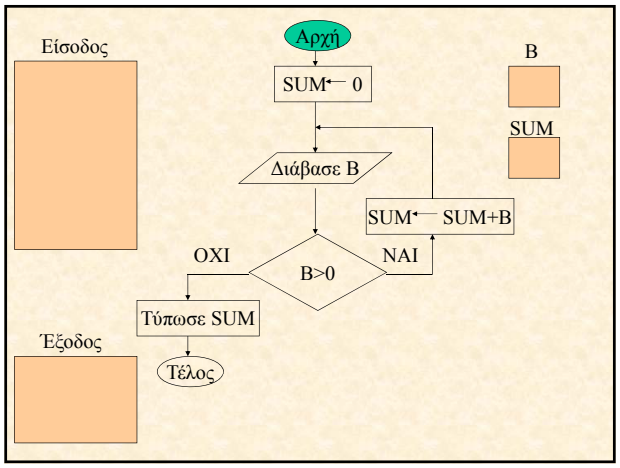


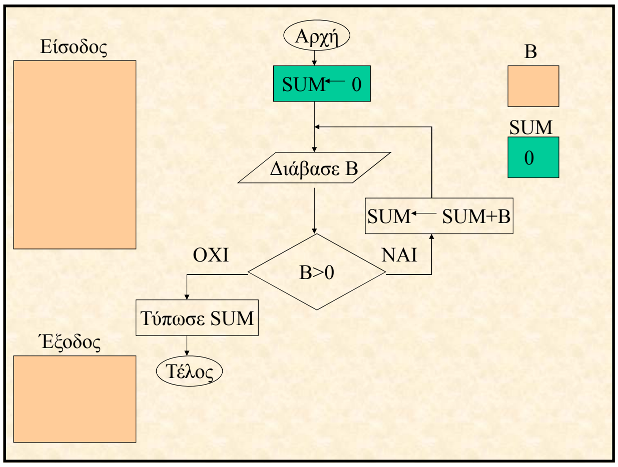
Σε αυτή τη μορφή επανάληψης, οι εντολές 1 θα εκτελεστούν **<u>άγνωστο αριθμό φορών</u>**.
Οι εντολές 1 θα εκτελούνται **<u>για όσο η συνθήκη είναι αληθής</u>**. Πρώτα εξετάζεται η συνθήκη και μετά, αν αυτή είναι αληθής, εκτελούνται οι εντολές 1. Δηλαδή οι εντολές 1 μπορεί να μην εκτελεστούν και καμία φορά.
<u>Προσοχή</u>: Μέσα στις εντολές 1, θα πρέπει να υπάρχει και μία εντολή που να «ανανεώνει» τη συνθήκη. Αλλιώς η συνθήκη είναι πάντα η ίδια, οπότε ή οι εντολές 1 δε θα εκτελεστούν ποτέ (αν η συνθήκη είναι ψευδής), ή θα εκτελούνται για πάντα (αν η συνθήκη είναι αληθής).

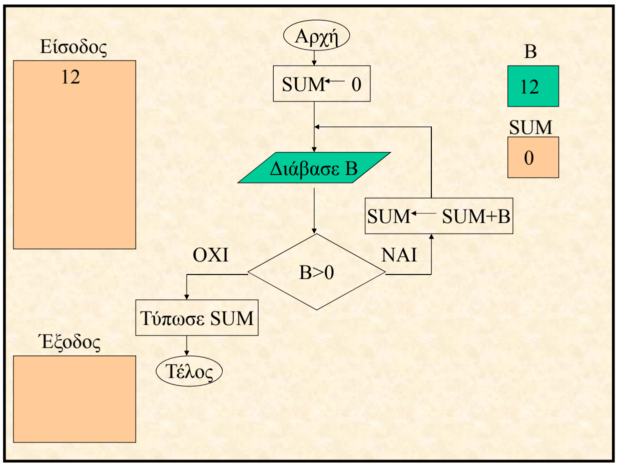
Παράδειγμα 1

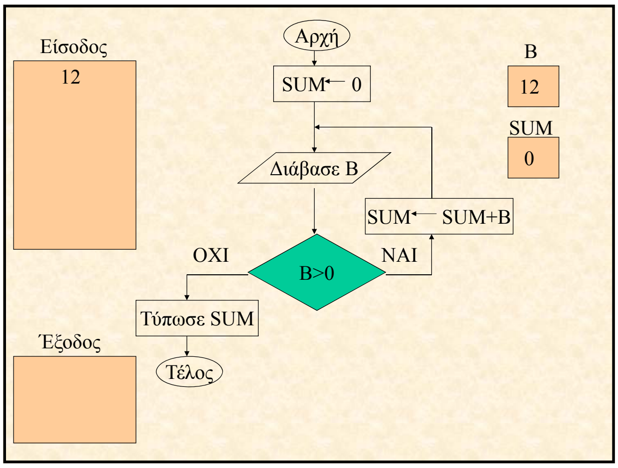
Να γραφεί πρόγραμμα που να επιλύει το ακόλουθο πρόβλημα. Δίδονται θετικοί αριθμοί. Όταν δοθεί ένας αρνητικός αριθμός, το πρόγραμμα εμφανίζει στην έξοδο το άθροισμα όλων των αριθμών, χωρίς να λάβει υπόψη τον τελευταίο αρνητικό αριθμό (ο οποίος έπαιξε το ρόλο της εντολής τερματισμού)

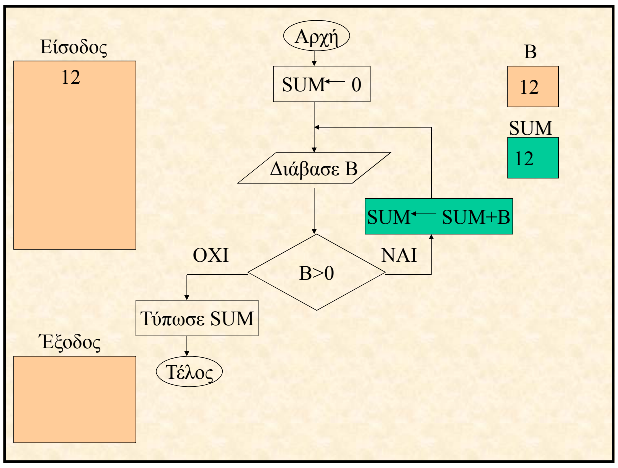


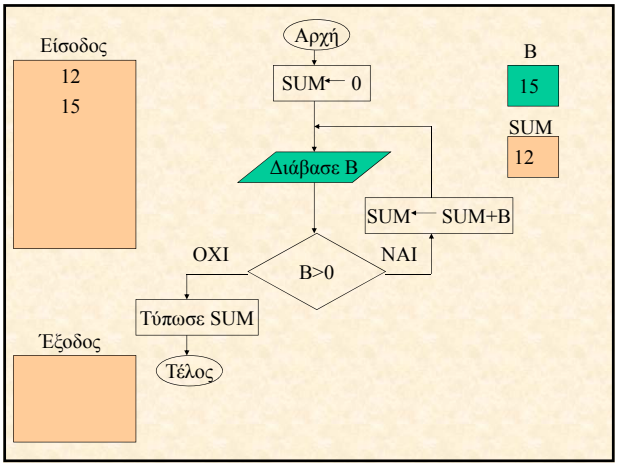


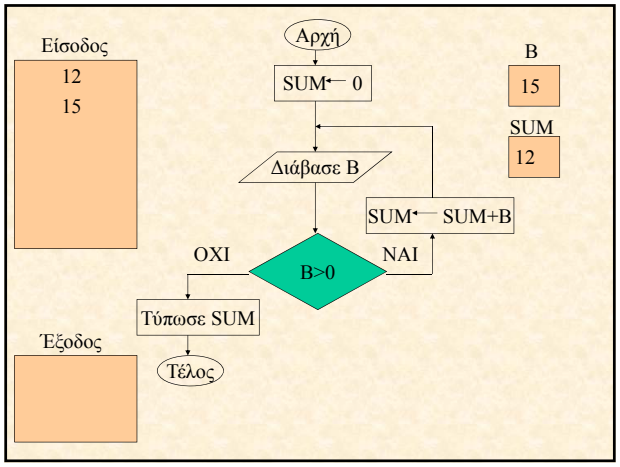

Είσοδος
Αρχή
SUM ← 0
B
SUM
0
Διάβασε B
SUM ← SUM+B
ΟΧΙ
ΝΑΙ
B>0
Τύπωσε SUM
Έξοδος
Τέλος

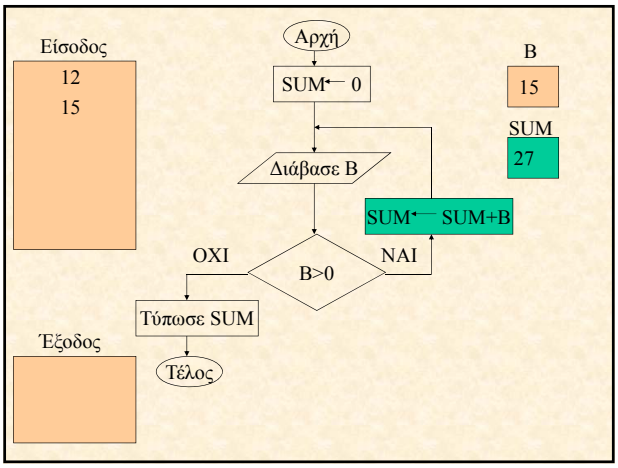

Είσοδος
12
Αρχή
SUM ← 0
B
12
SUM
0
Διάβασε B
SUM ← SUM+B
ΟΧΙ
ΝΑΙ
B>0
Τύπωσε SUM
Έξοδος
Τέλος

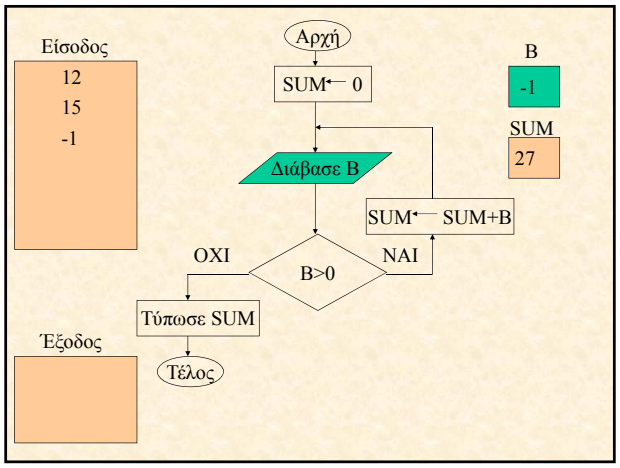

Είσοδος
12
Αρχή
SUM ← 0
B
12
SUM
0
Διάβασε B
SUM ← SUM+B
ΟΧΙ
ΝΑΙ
B>0
Τύπωσε SUM
Έξοδος
Τέλος

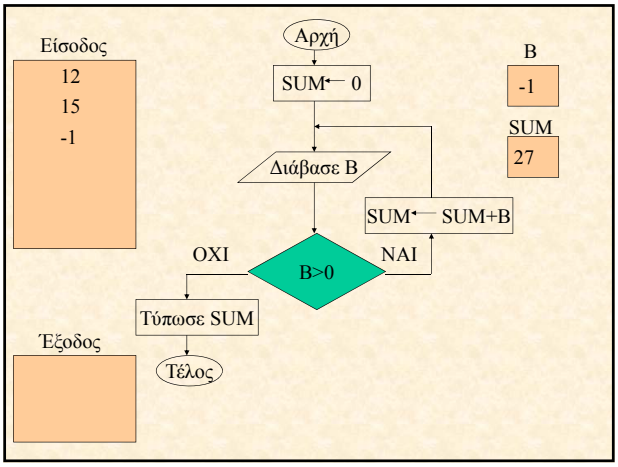

Είσοδος
12
Αρχή
SUM ← 0
B
12
SUM
12
Διάβασε B
SUM ← SUM+B
ΟΧΙ
ΝΑΙ
B>0
Τύπωσε SUM
Έξοδος
Τέλος

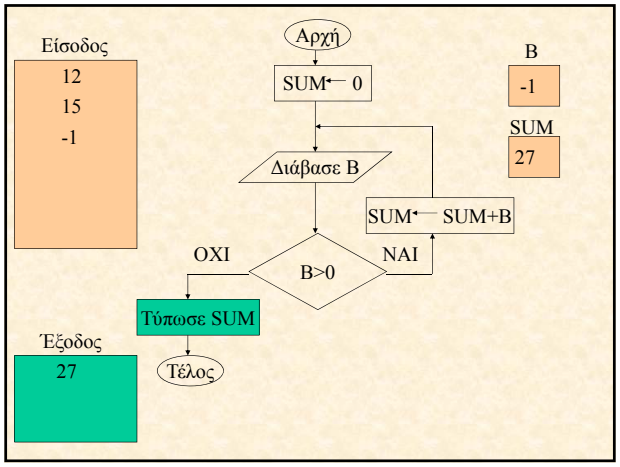

Είσοδος
12
15
Αρχή
SUM← 0
B
15
SUM
12
Διάβασε B
SUM← SUM+B
OXI
B>0
NAI
Τύπωσε SUM
Έξοδος
Τέλος

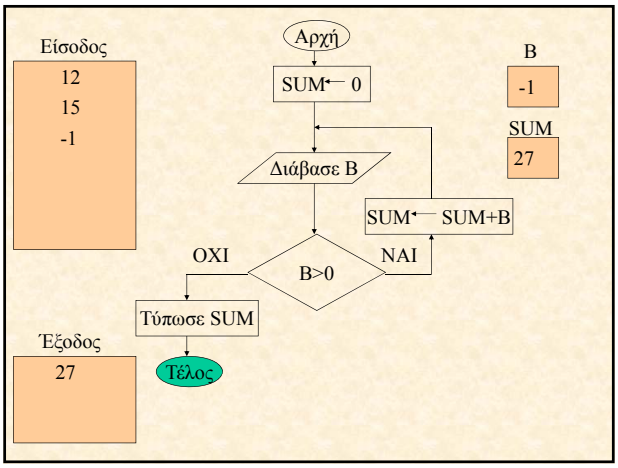

Είσοδος
12
15
Αρχή
SUM← 0
B
15
SUM
12
Διάβασε B
SUM← SUM+B
OXI
B>0
NAI
Τύπωσε SUM
Έξοδος
Τέλος


Είσοδος
12
15
Αρχή
SUM← 0
B
15
SUM
27
Διάβασε B
SUM← SUM+B
OXI
B>0
NAI
Τύπωσε SUM
Έξοδος
Τέλος


Είσοδος
12
15
-1
Αρχή
SUM← 0
B
-1
SUM
27
Διάβασε B
SUM← SUM+B
OXI
B>0
NAI
Τύπωσε SUM
Έξοδος
Τέλος

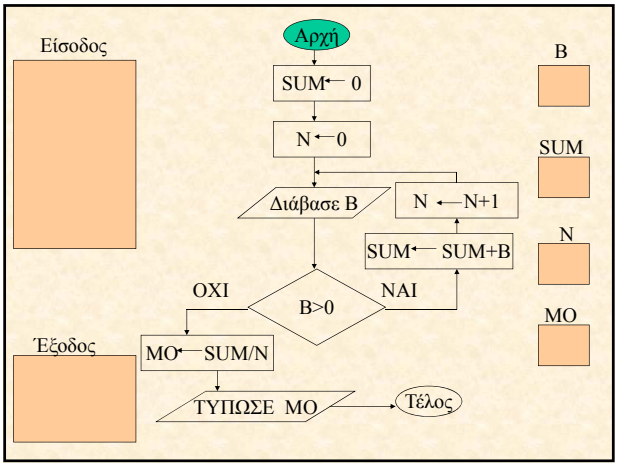


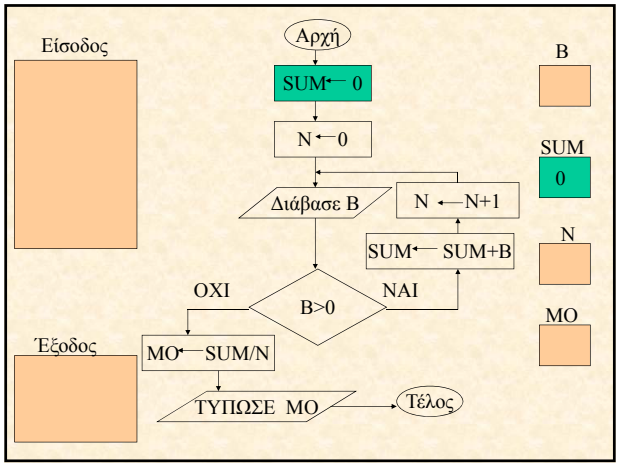


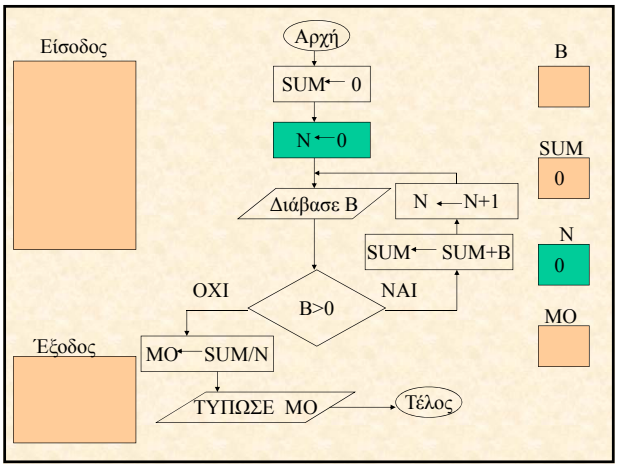


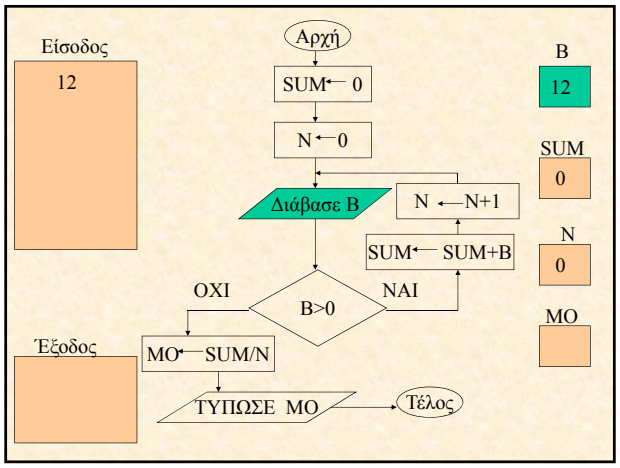
Παράδειγμα 2

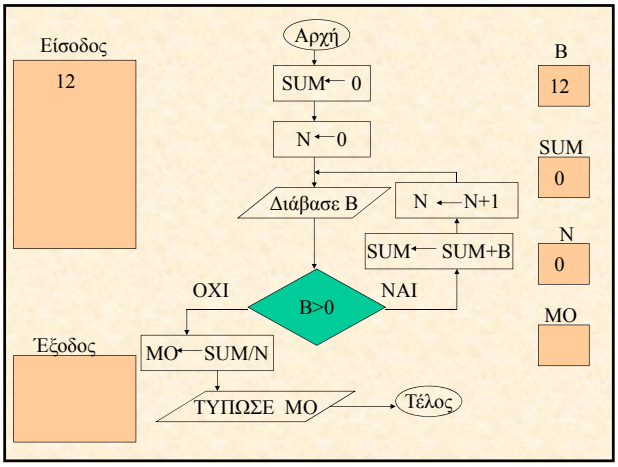
Να γραφεί πρόγραμμα που να επιλύει το ακόλουθο πρόβλημα. Δίδονται θετικοί αριθμοί. Όταν δοθεί ένας αρνητικός αριθμός, το πρόγραμμα εμφανίζει στην έξοδο το μέσο όρο όλων των αριθμών, χωρίς να λάβει υπόψη τον τελευταίο αρνητικό αριθμό (ο οποίος έπαιξε το ρόλο της εντολής τερματισμού)

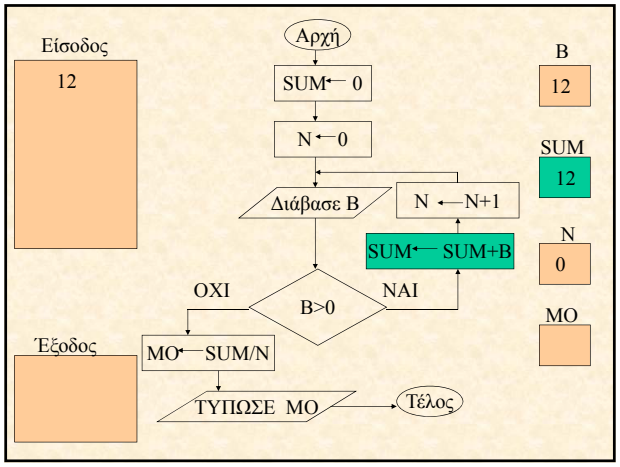

Είσοδος
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+B
Β>0
ΟΧΙ
ΝΑΙ
ΜΟ←SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
SUM
Ν
ΜΟ

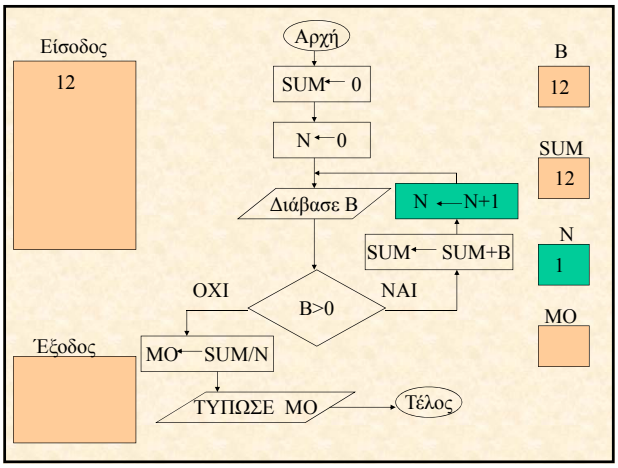

Είσοδος
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+B
Β>0
ΟΧΙ
ΝΑΙ
ΜΟ←SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
SUM
0
Ν
ΜΟ

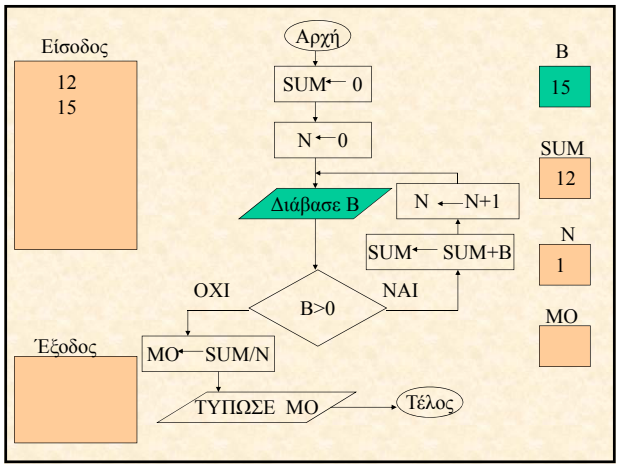

Είσοδος
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+B
Β>0
ΟΧΙ
ΝΑΙ
ΜΟ←SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
SUM
0
Ν
0
ΜΟ

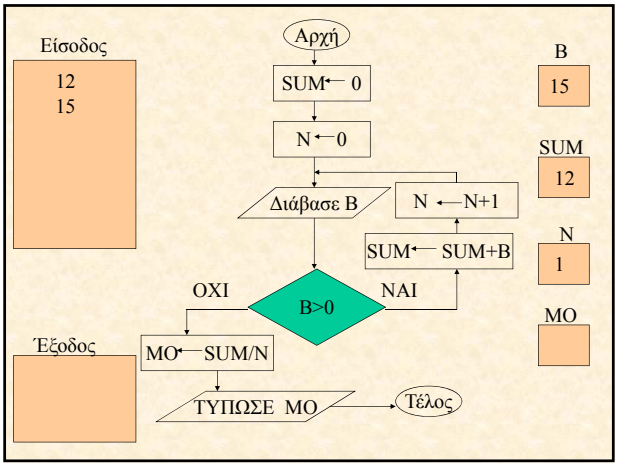

Είσοδος
12
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+B
Β>0
ΟΧΙ
ΝΑΙ
ΜΟ←SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
12
SUM
0
Ν
0
ΜΟ

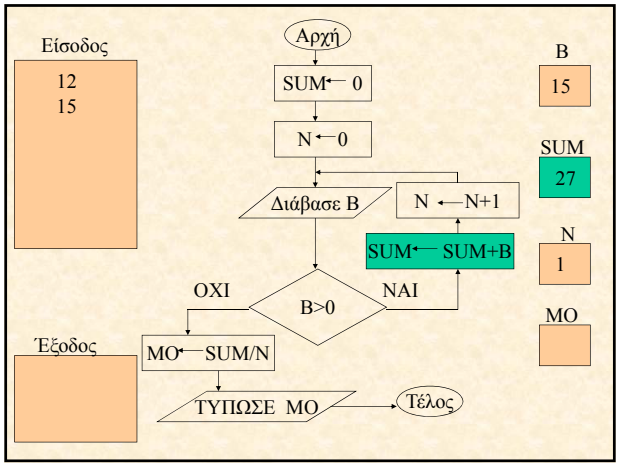

Είσοδος
12
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+B
ΟΧΙ
Β>0
ΝΑΙ
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
B
12
SUM
0
N
0
MO

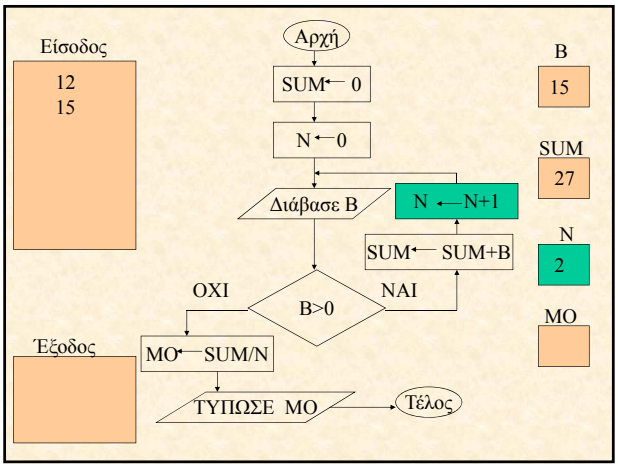

Είσοδος
12
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+B
ΟΧΙ
Β>0
ΝΑΙ
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
B
12
SUM
12
N
0
MO

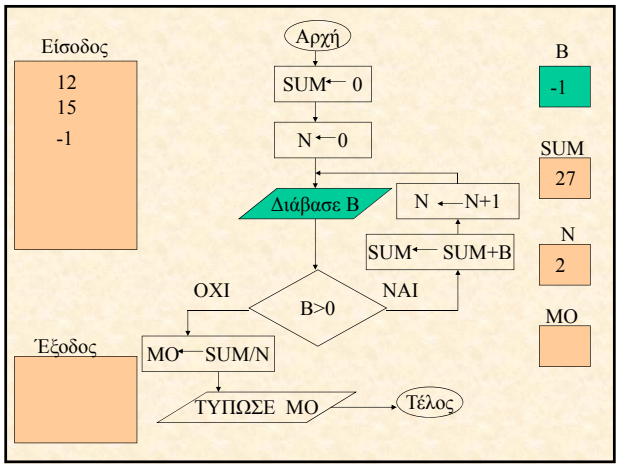

Είσοδος
12
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+B
ΟΧΙ
Β>0
ΝΑΙ
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
B
12
SUM
12
N
1
MO

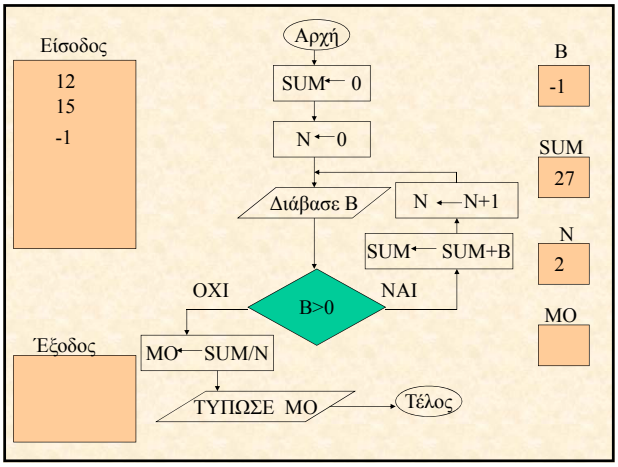

Είσοδος
12
15
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+B
ΟΧΙ
Β>0
ΝΑΙ
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
B
15
SUM
12
N
1
MO

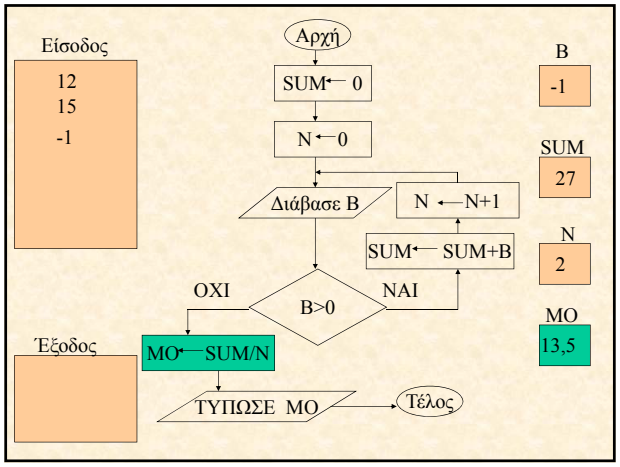

Είσοδος
12
15
Αρχή
SUM← 0
N←0
Διάβασε B
N ←N+1
SUM← SUM+B
B>0
ΟΧΙ
ΝΑΙ
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
Έξοδος
B
15
SUM
12
N
1
MO

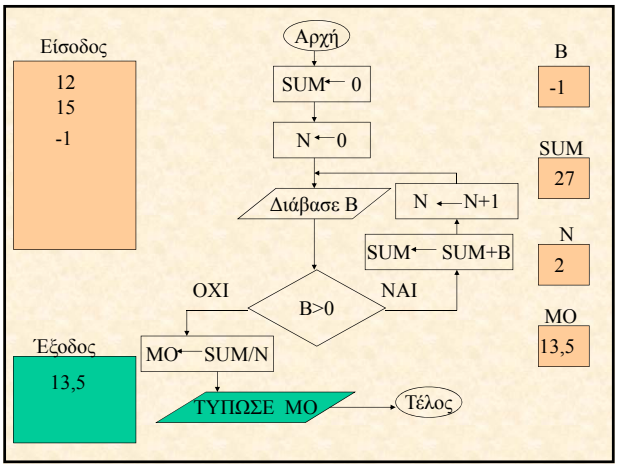

Είσοδος
12
15
Αρχή
SUM← 0
N←0
Διάβασε B
N ←N+1
SUM← SUM+B
B>0
ΟΧΙ
ΝΑΙ
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
Έξοδος
B
15
SUM
27
N
1
MO

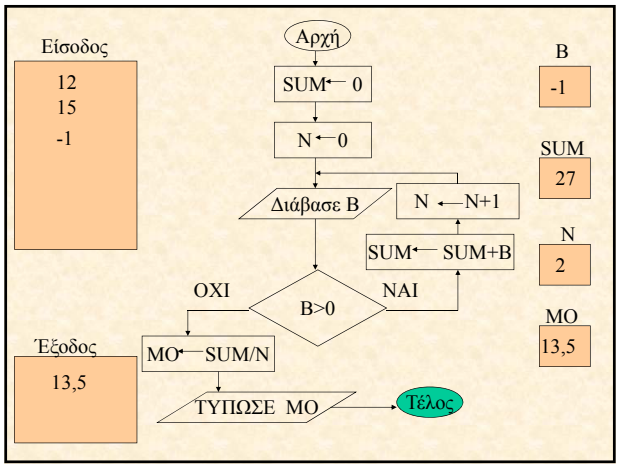

Είσοδος
12
15
Αρχή
SUM← 0
N←0
Διάβασε B
N ←N+1
SUM← SUM+B
B>0
ΟΧΙ
ΝΑΙ
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
Έξοδος
B
15
SUM
27
N
2
MO


Είσοδος
12
15
-1
Αρχή
SUM← 0
N←0
Διάβασε B
N ←N+1
SUM← SUM+B
B>0
ΟΧΙ
ΝΑΙ
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
Έξοδος
B
-1
SUM
27
N
2
MO


Είσοδος
12
15
-1
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+Β
ΟΧΙ
Β>0
ΝΑΙ
ΜΟ← SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
-1
SUM
27
N
2
ΜΟ

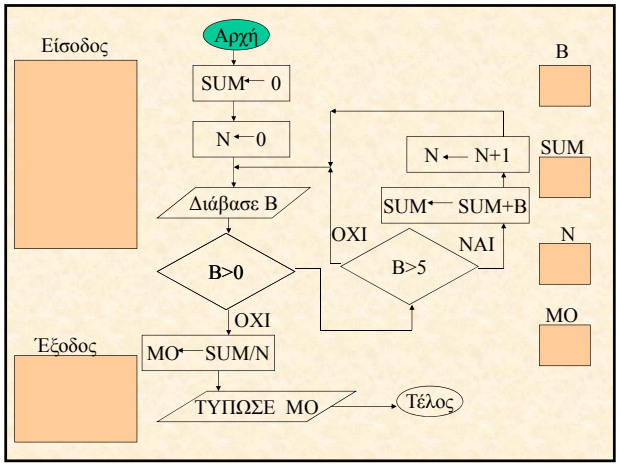

Είσοδος
12
15
-1
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+Β
ΟΧΙ
Β>0
ΝΑΙ
ΜΟ← SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
-1
SUM
27
N
2
ΜΟ
13,5

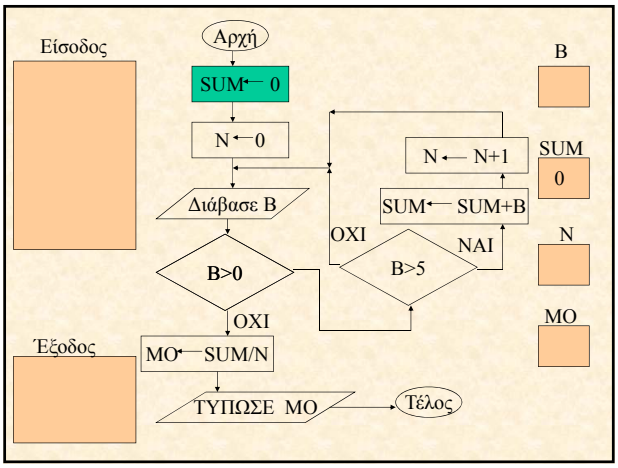

Είσοδος
12
15
-1
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+Β
ΟΧΙ
Β>0
ΝΑΙ
ΜΟ← SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
13,5
Β
-1
SUM
27
N
2
ΜΟ
13,5

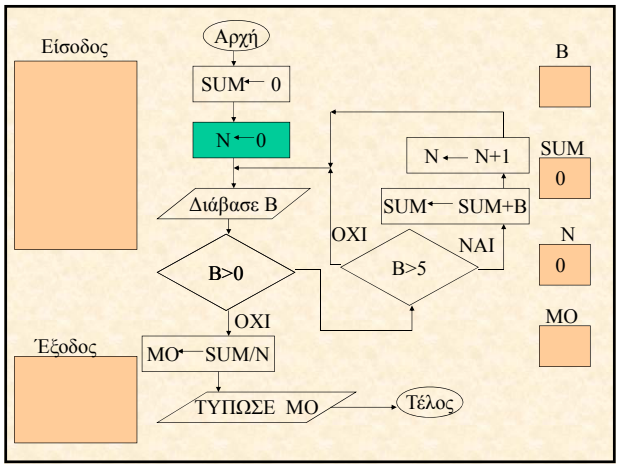

Είσοδος
12
15
-1
Αρχή
SUM← 0
N←0
Διάβασε Β
N ←N+1
SUM← SUM+Β
ΟΧΙ
Β>0
ΝΑΙ
ΜΟ← SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
13,5
Β
-1
SUM
27
N
2
ΜΟ
13,5

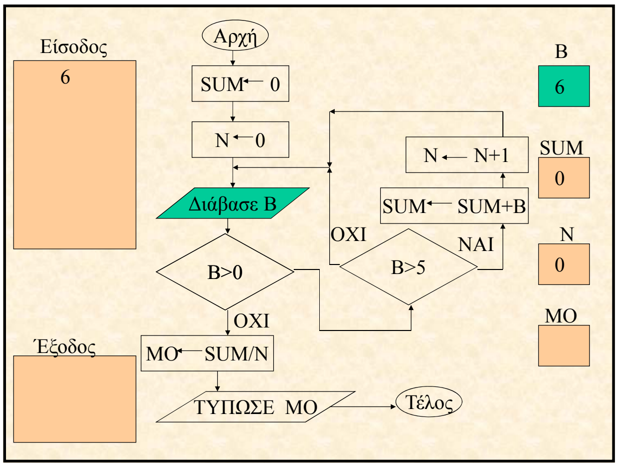
Παράδειγμα 3

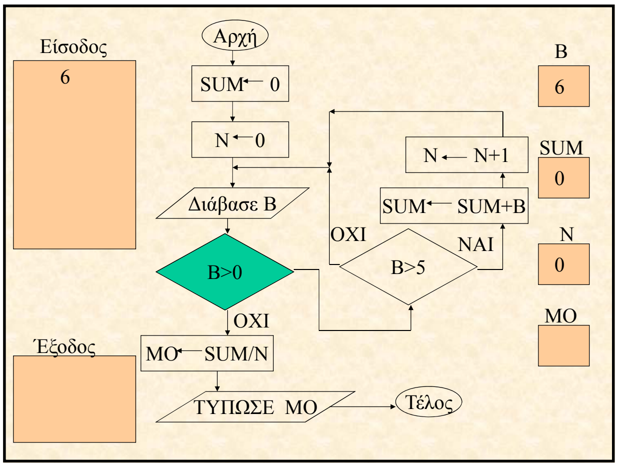
Να γραφεί πρόγραμμα που να επιλύει το ακόλουθο πρόβλημα. Δίδονται θετικοί αριθμοί. Όταν δοθεί ένας αρνητικός αριθμός, το πρόγραμμα εμφανίζει στην έξοδο το μέσο όρο όλων των αριθμών που ήταν μεγαλύτεροι του 5.

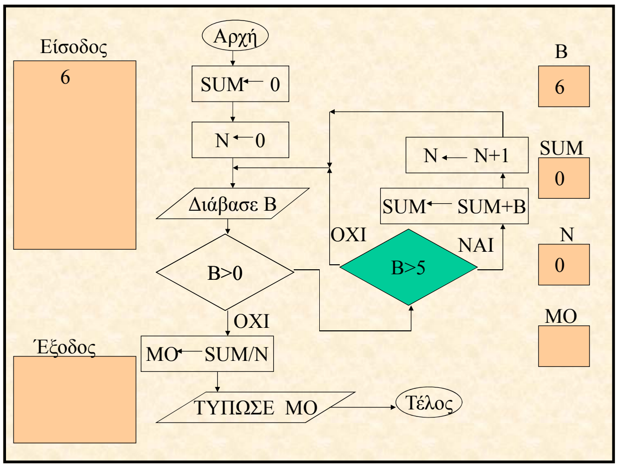


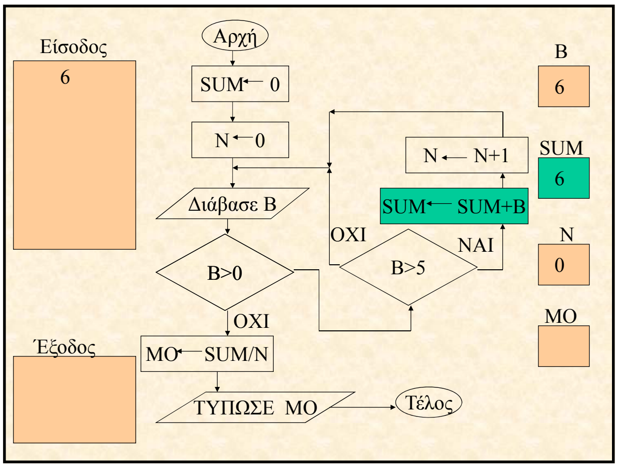


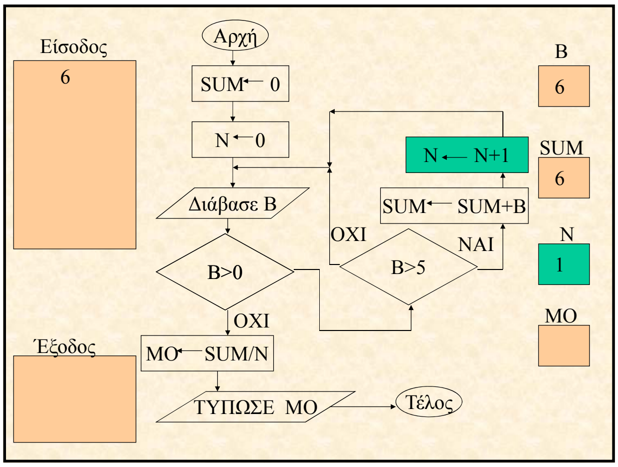


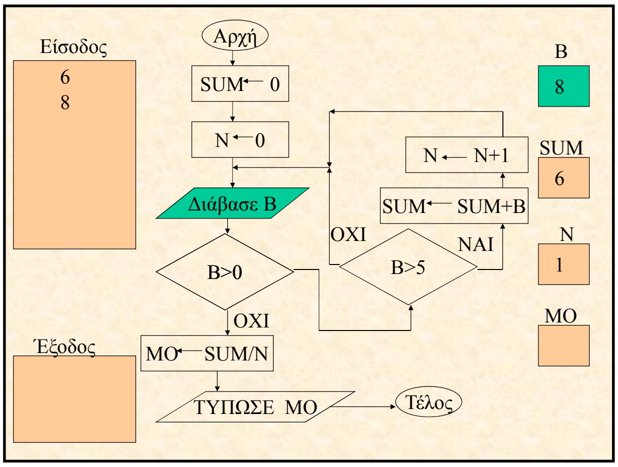

Είσοδος
6
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N← N+1
SUM← SUM+B
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
B
6
SUM
0
N
0
MO

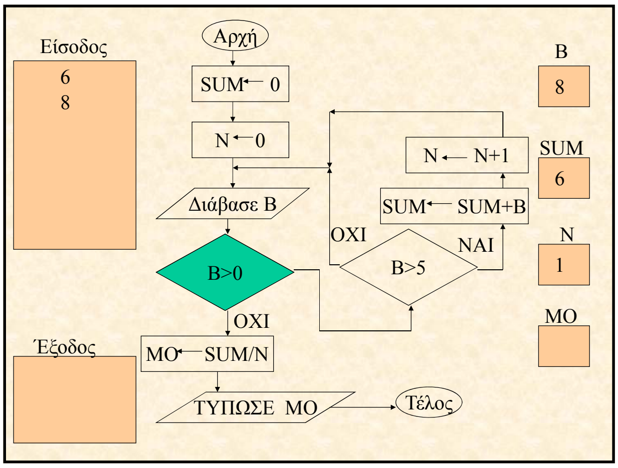

Είσοδος
6
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N← N+1
SUM← SUM+B
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
B
6
SUM
0
N
0
MO

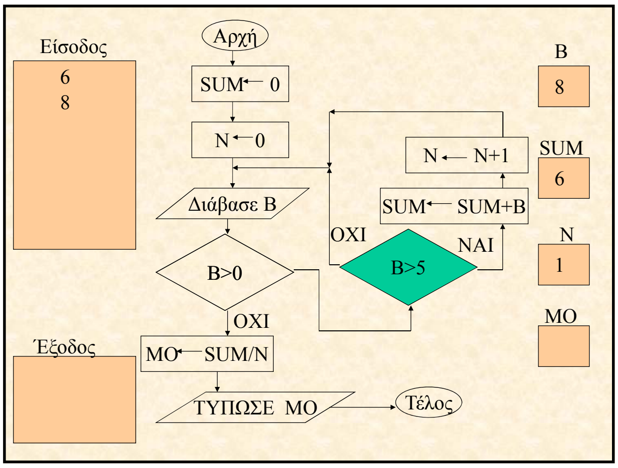

Είσοδος
6
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N← N+1
SUM← SUM+B
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
B
6
SUM
0
N
0
MO

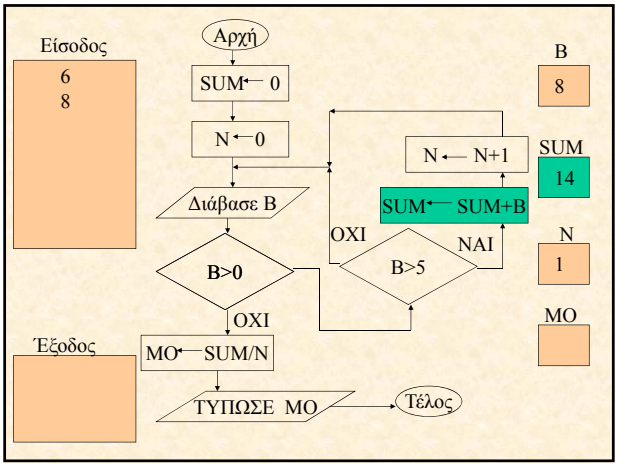

Είσοδος
6
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N← N+1
SUM← SUM+B
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
B
6
SUM
6
N
0
MO

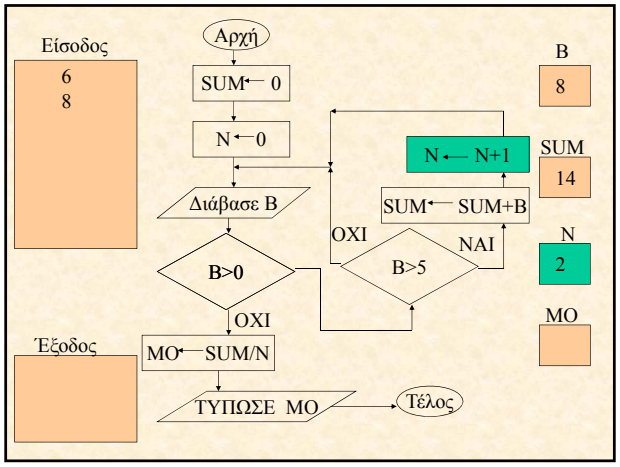

Είσοδος
6
Αρχή
SUM← 0
N ← 0
Διάβασε B
B>0
ΟΧΙ
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
N ← N+1
SUM← SUM+B
ΟΧΙ
B>5
ΝΑΙ
B
6
SUM
6
N
1
MO

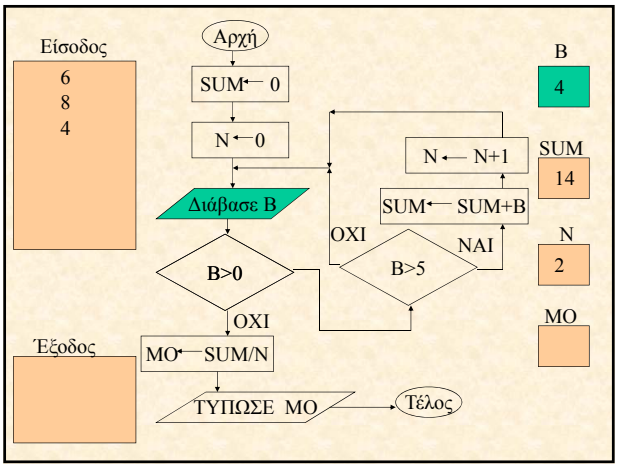

Είσοδος
6
8
Αρχή
SUM← 0
N ← 0
Διάβασε B
B>0
ΟΧΙ
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
N ← N+1
SUM← SUM+B
ΟΧΙ
B>5
ΝΑΙ
B
8
SUM
6
N
1
MO

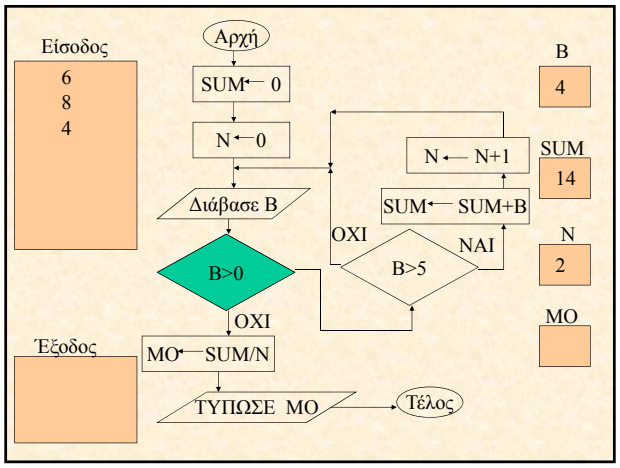

Είσοδος
6
8
Αρχή
SUM← 0
N ← 0
Διάβασε B
B>0
ΟΧΙ
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
N ← N+1
SUM← SUM+B
ΟΧΙ
B>5
ΝΑΙ
B
8
SUM
6
N
1
MO

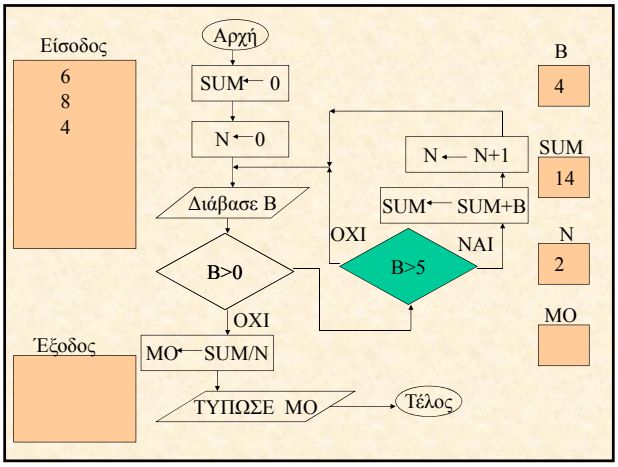

Είσοδος
6
8
Αρχή
SUM← 0
N ← 0
Διάβασε B
B>0
ΟΧΙ
Έξοδος
MO← SUM/N
ΤΥΠΩΣΕ MO
Τέλος
N ← N+1
SUM← SUM+B
ΟΧΙ
B>5
ΝΑΙ
B
8
SUM
6
N
1
MO

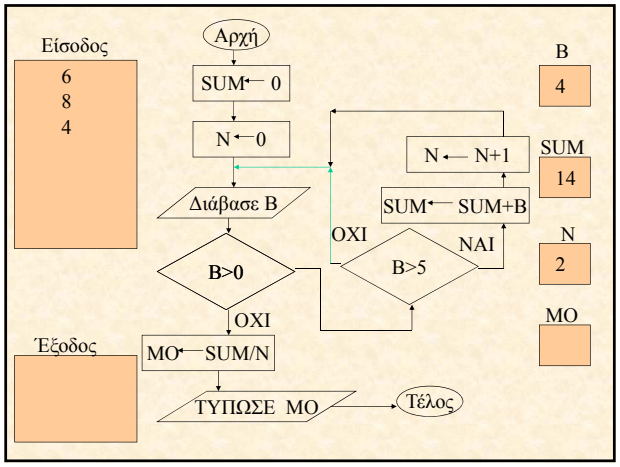

Είσοδος
6
8
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N ← N+1
SUM← SUM+Β
ΟΧΙ
ΜΟ←SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
8
SUM
14
N
1
ΜΟ

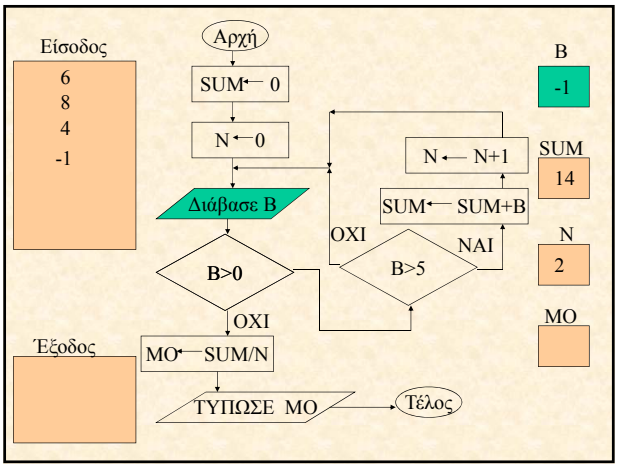

Είσοδος
6
8
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N ← N+1
SUM← SUM+Β
ΟΧΙ
ΜΟ←SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
8
SUM
14
N
2
ΜΟ

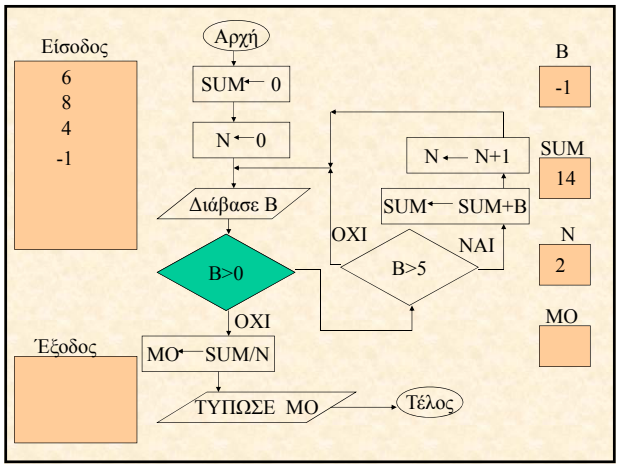

Είσοδος
6
8
4
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N ← N+1
SUM← SUM+Β
ΟΧΙ
ΜΟ←SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
4
SUM
14
N
2
ΜΟ

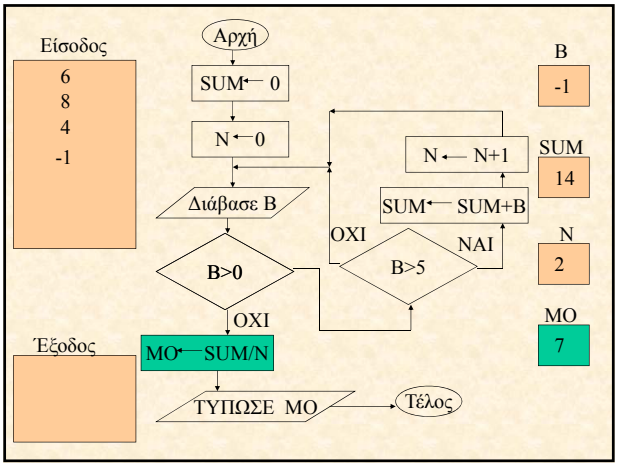

Είσοδος
6
8
4
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N ← N+1
SUM← SUM+Β
ΟΧΙ
ΜΟ←SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
4
SUM
14
N
2
ΜΟ

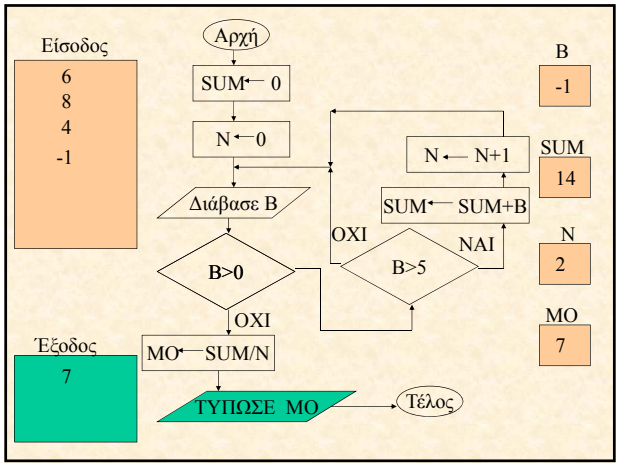

Είσοδος
6
8
4
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N← N+1
SUM← SUM+Β
ΟΧΙ
ΜΟ← SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
4
SUM
14
N
2
ΜΟ

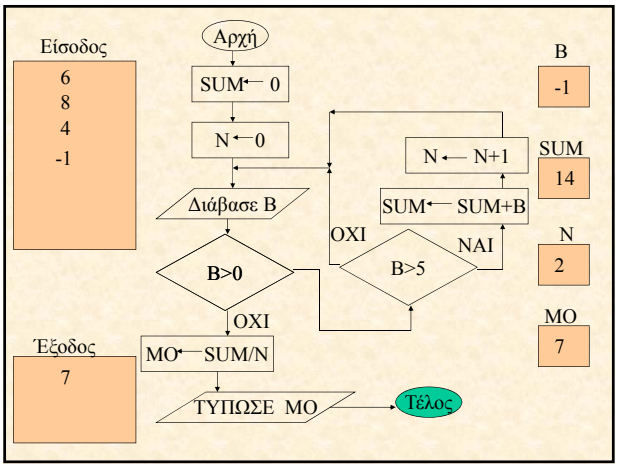

Είσοδος
6
8
4
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N← N+1
SUM← SUM+Β
ΟΧΙ
ΜΟ← SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
4
SUM
14
N
2
ΜΟ


Είσοδος
6
8
4
-1
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N← N+1
SUM← SUM+Β
ΟΧΙ
ΜΟ← SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
-1
SUM
14
N
2
ΜΟ


Είσοδος
6
8
4
-1
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
N← N+1
SUM← SUM+Β
ΟΧΙ
ΜΟ← SUM/N
ΤΥΠΩΣΕ ΜΟ
Τέλος
Έξοδος
Β
-1
SUM
14
N
2
ΜΟ


Είσοδος
6
8
4
-1
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
SUM← SUM+Β
N ← N+1
Β
-1
SUM
14
Ν
2
ΜΟ
7
ΜΟ← SUM/N
Έξοδος
ΤΥΠΩΣΕ ΜΟ
Τέλος

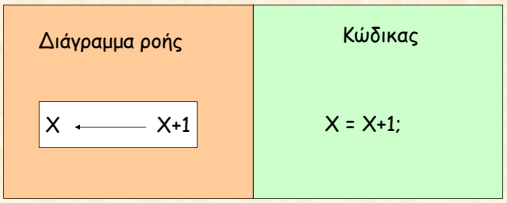

Είσοδος
6
8
4
-1
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
SUM← SUM+Β
N ← N+1
Β
-1
SUM
14
Ν
2
ΜΟ
7
ΜΟ← SUM/N
Έξοδος
7
ΤΥΠΩΣΕ ΜΟ
Τέλος


Είσοδος
6
8
4
-1
Αρχή
SUM← 0
N←0
Διάβασε Β
Β>0
ΟΧΙ
Β>5
ΝΑΙ
SUM← SUM+Β
N ← N+1
Β
-1
SUM
14
Ν
2
ΜΟ
7
ΜΟ← SUM/N
Έξοδος
7
ΤΥΠΩΣΕ ΜΟ
Τέλος

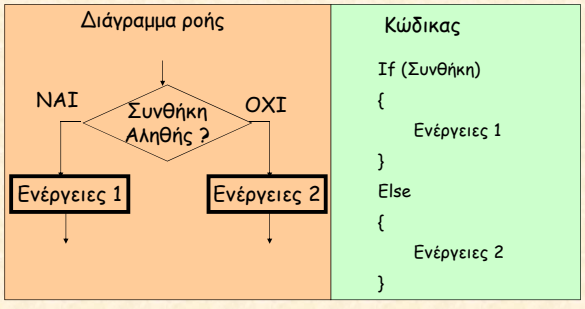

Επανέλαβε όσο η συνθήκη είναι αληθής
Ενέργειες
Συνθήκη Αληθής ?
ΝΑΙ
ΟΧΙ
Σε αυτή τη επανάληψης οι ενέργειες θα εκτελεστούν τουλάχιστον μία φορά, αφού εκτελούνται πριν εξεταστεί για πρώτη φορά η συνθήκη.

Από το διάγραμμα ροής στον κώδικα

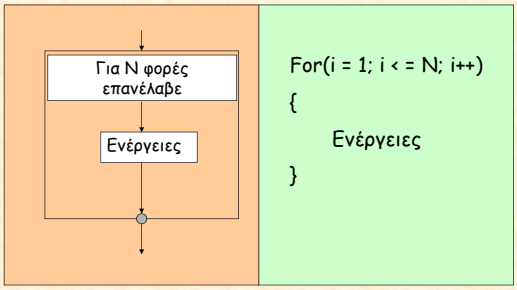
Εντολές απόδοσης τιμής



Διακλάδωση

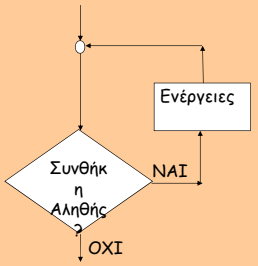


Για Ν φορές επανέλαβε



Επανέλαβε εφόσον η συνθήκη είναι αληθής

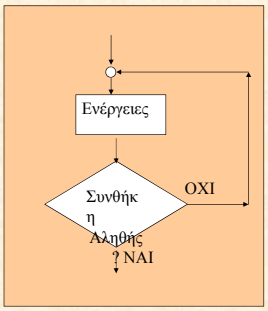
Διάγραμμα ροής



Ψευδοκώδικας

```
while (Συνθήκη)
{
    Ενέργειες
}
```

Επανέλαβε όσο η συνθήκη είναι αληθής




```
do
{
    Ενέργειες
}
while (Συνθήκη);
```

Κώδικας

Το πρώτο πρόγραμμα

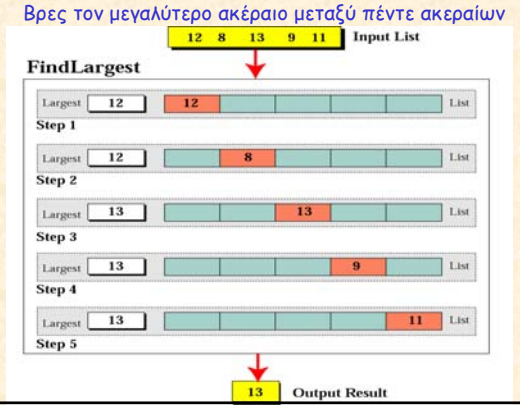
```
#include<stdio.h>
void main()
{
        printf("hello world");
}
```

Πρόσθεση 5 αριθμών

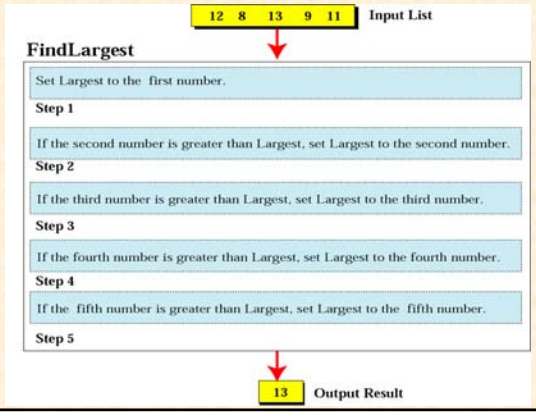
```
#include<stdio.h>

void main()
{float sum, a1, a2, a3, a4, a5, av;
a1=1.65;
a2=1.80;
a3=1.76;
a4=1.73;
a5=1.63;
sum=a1+a2+a3+a4+a5;
av=sum/5.0;
printf("%.2f",av);
return;
}
```

Παράδειγμα 4

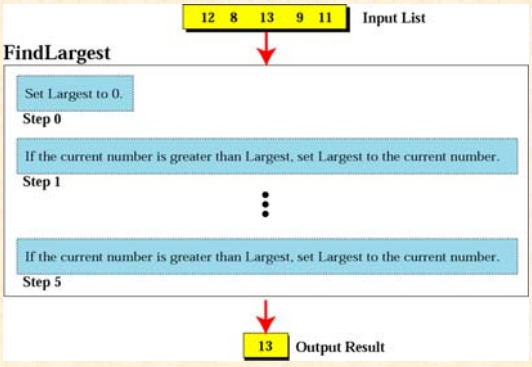
Βρες τον μεγαλύτερο ακέραιο μεταξύ πέντε ακεραίων



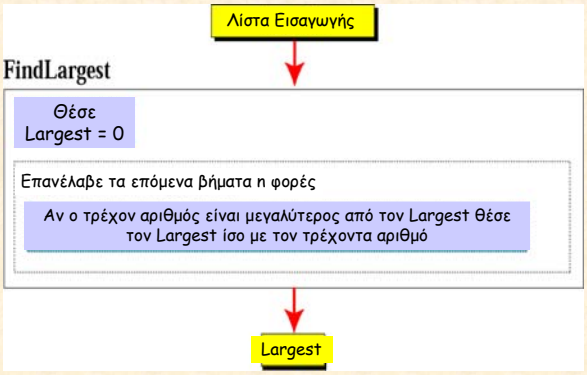
Ορίζοντας ενέργειες στον αλγόριθμο FindLargest



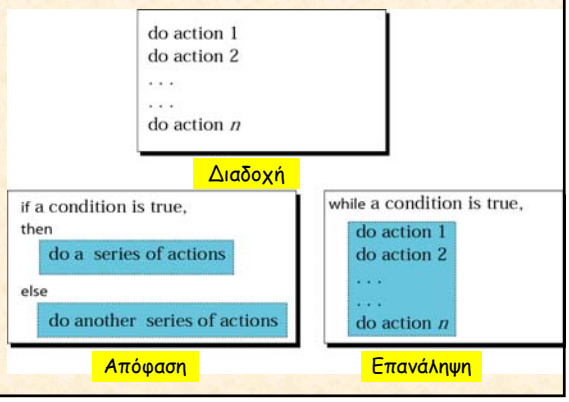
Επανεύρεση – ΑλγόριθμοςFindLargest



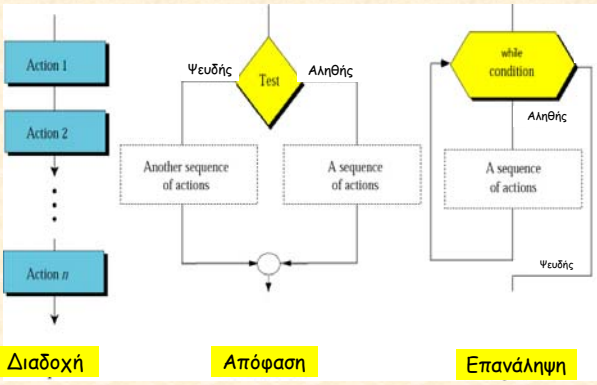
Γενίκευση του αλγορίθμου FindLargest



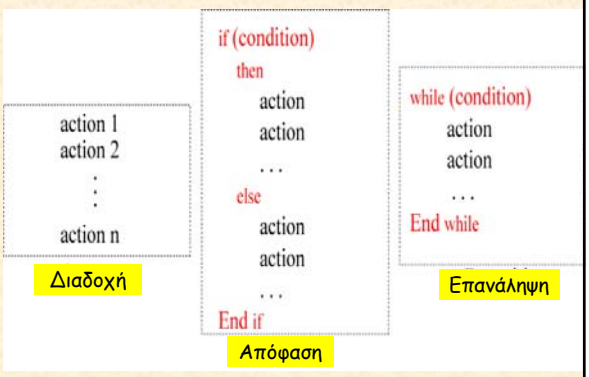
Τρείς απλές δομές



Διαγράμματα ροής των απλών δομών



Ψευδοκώδικες των απλών δομών



Παράδειγμα Α

Να γραφεί αλγόριθμος σε ψευδοκώδικα που βρίσκει τον μέσο όρο δύο αριθμών

Λύση

Algorithm: **Average of two**

```
AverageOfTwo
Input: Δύο αριθμοί
1. Πρόσθεσε τους δύο αριθμούς
2. Διαίρεσε το αποτέλεσμα με το 2
3. Επέστρεψε το αποτέλεσμα στο βήμα 2
End
```

Παράδειγμα Β

Να γραφεί αλγόριθμος αλλαγής αριθμητικού βαθμού σε βαθμό pass/no pass.

Solution

Algorithm: Βαθμός Pass/no pass

```
   Pass/NoPassGrade
   Input: One number
1. if (the number is greater than or equal to 70)
   then
      1.1  Set the grade to "pass"
   else
      1.2  Set the grade to "nopass"
   End if
2. Return the grade
   End
```

Παράδειγμα Γ

Να γραφεί αλγόριθμος αλλαγής αριθμητικού βαθμού σε αλφαριθμητικό βαθμό.

Solution

Algorithm: Αλφαριθμητικός Βαθμός

```
   LetterGrade
   Input: One number
1. if (the number is between 90 and 100, inclusive)
   then
      1.1  Set the grade to "A"
   End if
2. if (the number is between 80 and 89, inclusive)
   then
      2.1  Set the grade to "B"
   End if
```

Continues on the next slide

Παράδειγμα Γ

Algorithm: Αλφαριθμητικός βαθμός

```
3. if (the number is between 70  and 79, inclusive)
   then
      3.1  Set the grade to "C"
   End if
4. if (the number is between 60  and 69, inclusive)
   then
      4.1  Set the grade to "D"
   End if
```

```
5. If (the number is less than 60)
   then
      5.1  Set the grade to "F"
   End if
6. Return the grade
   End
```

Παράδειγμα Δ

Algorithm: Find largest: Να βρεθεί ο μεγαλύτερος αριθμός ενός συνόλου αριθμών – δεν γνωρίζουμε το πλήθος τους

```
   FindLargest
   Input: A list of positive integers
1. Set Largest to 0
2. while (more integers)
     2.1  if (the integer is greater than Largest)
            then
              2.1.1  Set largest to the value of the
   integer
          End if
   End while
3. Return Largest
   End
```

Παράδειγμα Ε

Algorithm: Βρες τον μεγαλύτερο από 1000 αριθμούς

```
   FindLargest
   Input: 1000 positive integers
1. Set Largest to 0
2. Set Counter to 0
3. while (Counter less than 1000)
    3.1  if (the integer is greater than Largest)
           then
             3.1.1  Set Largest to the value of the
   integer
           End if
     3.2  Increment Counter
   End while
4. Return Largest
   End
```

Η έννοια του υπο-αλγορίθμου

```
FindLargest
Input: A list of integers
1. Set Largest to 0
2. while (more integers)
     2.1 FindLarger
   End while
3. Return Largest
End
```

```
FindLarger
Input: Largest and integer
1. if (integer greater than Largest)
     then
       1.1 Set Largest to the value of integer
     End if
End
```

Παράδειγμα Στ

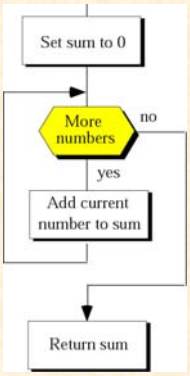
Αλγόριθμος: Find largest

```
   FindLargest
   Input: Λίστα θετικών ακεραίων
1. Θέσε Largest to 0
2. while (more integers)
    2.1  FindLarger
   End while
3. Return Largest
   End
```

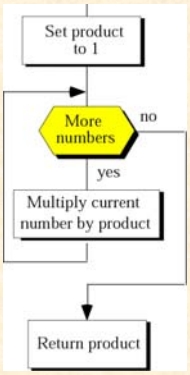
Υπο-αλγόριθμος: Find larger

```
   FindLarger
   Input: Ο μεγαλύτερος και ο τρέχον ακέραιος
1. if (the integer is greater than Largest)
   then
         1.1 Set Largest to the value of the integer
   End if
   End
```

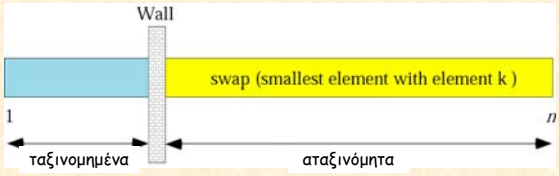
Άθροιση



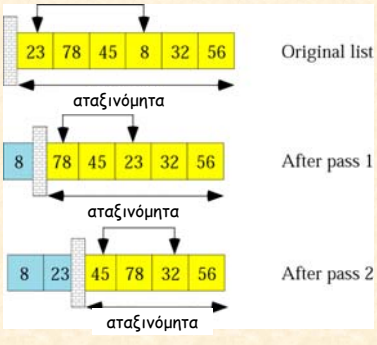
Γινόμενο



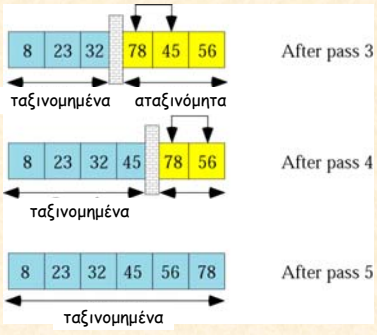
Ταξινόμηση επιλογής



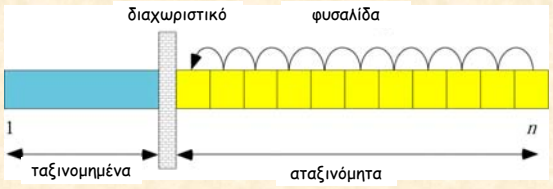
Παράδειγμα ταξινόμησης επιλογής



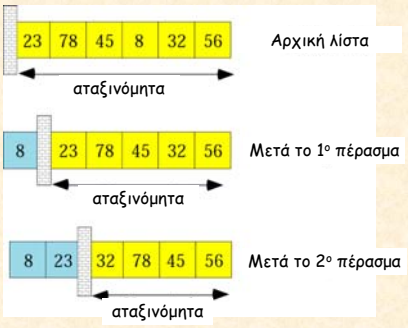
Παράδειγμα ταξινόμησης επιλογής



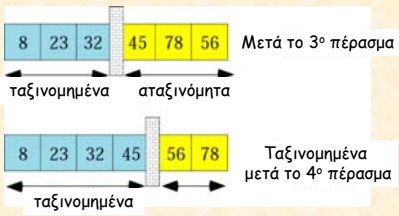
Ταξινόμηση φυσαλίδας (bubble sort)



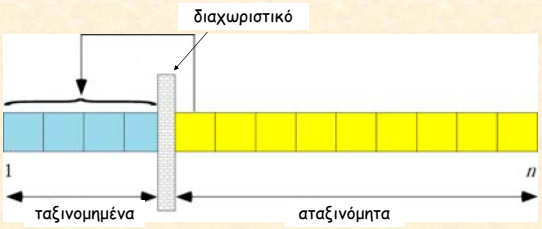
Παράδειγμα της ταξινόμησης φυσαλίδας



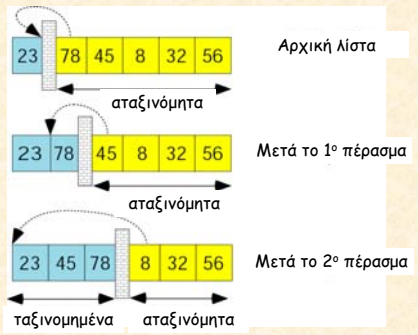
Παράδειγμα της ταξινόμηση φυσαλίδας



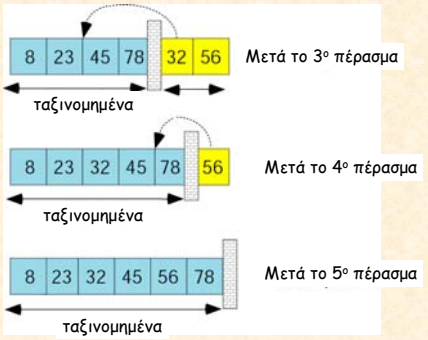
Ταξινόμηση εισαγωγής (Insertion sort)



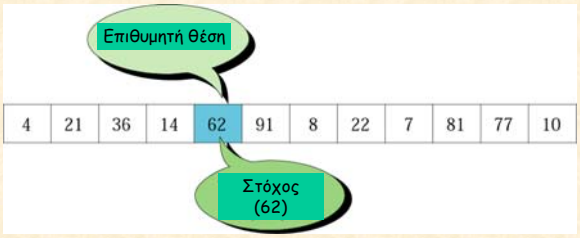
Παράδειγμα ταξινόμησης εισαγωγής



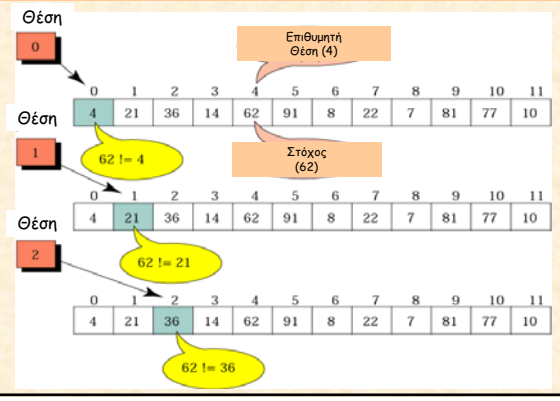
Παράδειγμα ταξινόμησης εισαγωγής



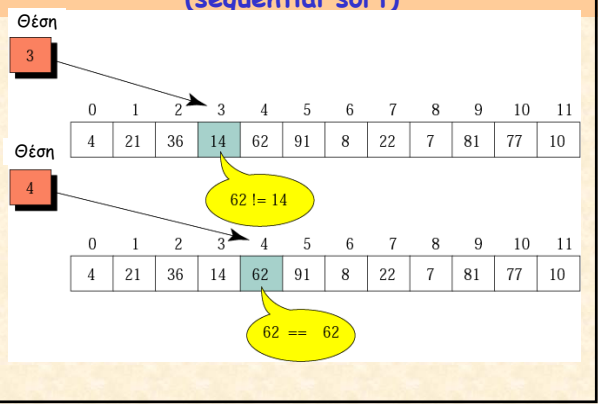
Η έννοια της αναζήτησης



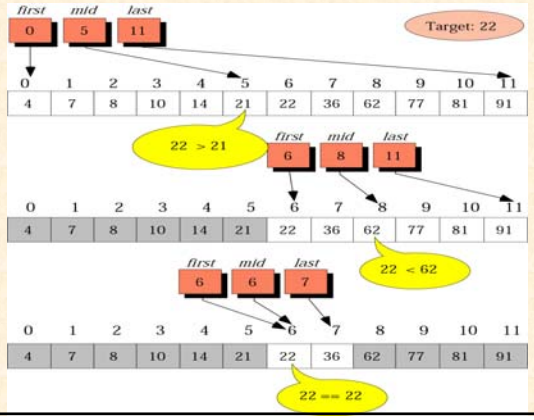
Παράδειγμα διαδοχικής ταξινόμησης (sequential sort)



Παράδειγμα διαδοχικής ταξινόμησης (sequential sort)



Παράδειγμα δυαδικής αναζήτησης



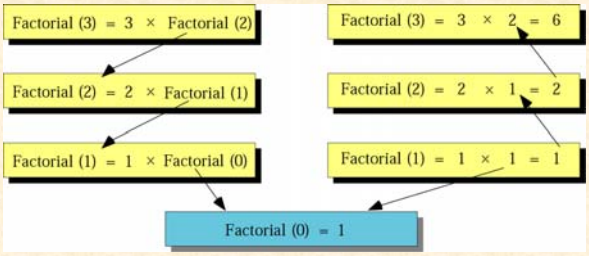
Επαναληπτικός Ορισμός του Παραγοντικού

$$\text{Factorial}(n) = \begin{cases} 1 & \text{if } n = 0 \\ n \times (n-1) \times (n-2) \times \ldots \times 3 \times 2 \times 1 & \text{if } n > 0 \end{cases}$$

Αναδρομικός Ορισμός του Παραγοντικού

$$\text{Factorial}(n) = \begin{cases} 1 & \text{if } n = 0 \\ n \times \text{Factorial}(n-1) & \text{if } n > 0 \end{cases}$$

Tracing recursive solution to factorial problem



Επαναληπτικό Παραγοντικό

Algorithm : Iterative factorial

```
   Factorial
   Input: Θετικός ακέραιος num
1. Θέσε FactN ίσο με 1
2. Θέσε i ίσο με 1
3. Όσο (i είναι μικρότερο ή ίσο με num)
       3.1  Θέσε FactN ίσο με FactN x I
       3.2  Αύξησε i κατά 1
   Τέλος όσο
1. Επέστρεψε FactN
   End
```

Αναδρομικό Παραγοντικό

Algorithm: Recursive factorial

```
   Factorial
   Input: Θετικός ακέραιος num
1. Αν (num είναι ίσος με 0)
   τότε
      1.1  επέστρεψε 1
   else
   1.2  επέστρεψε num x Factorial (num – 1)
   τέλος αν
   End
```