



Κεφάλαιο 3

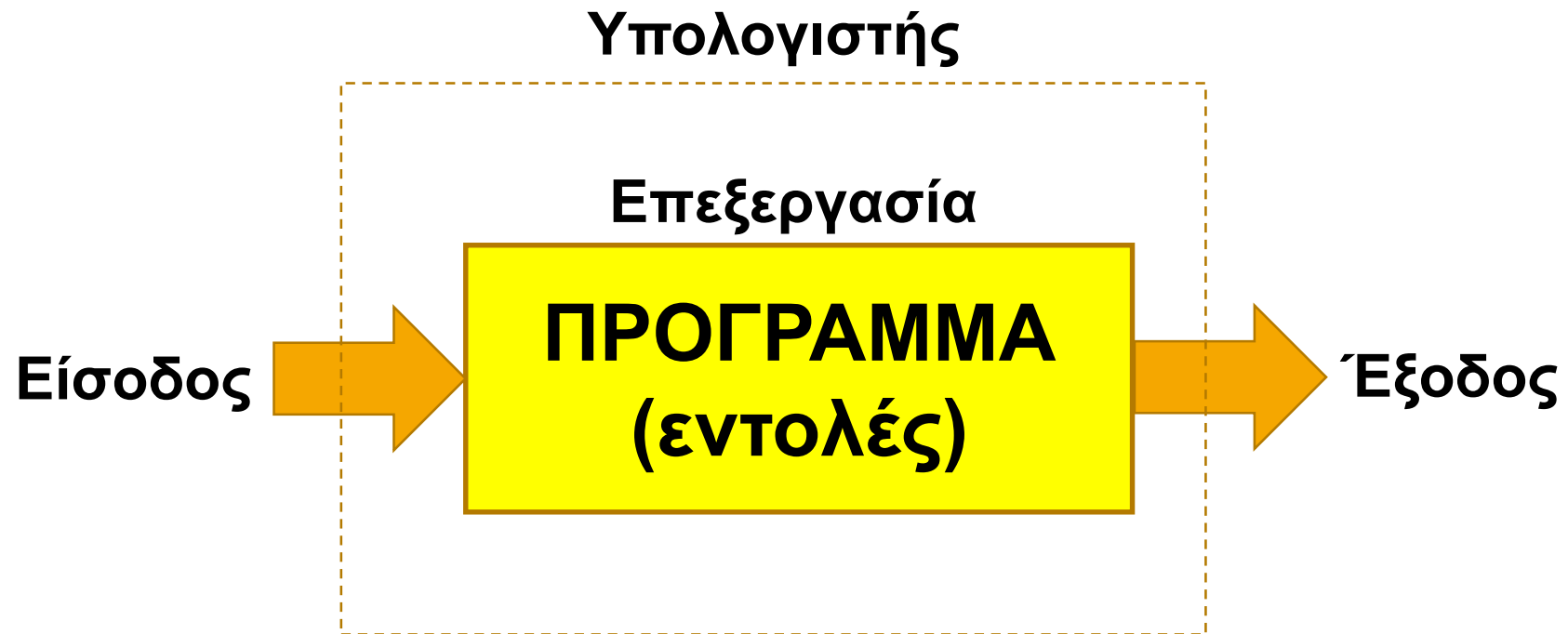
Προγραμματίζοντας
για πρώτη φορά

Πληροφορική & Υπολογιστές

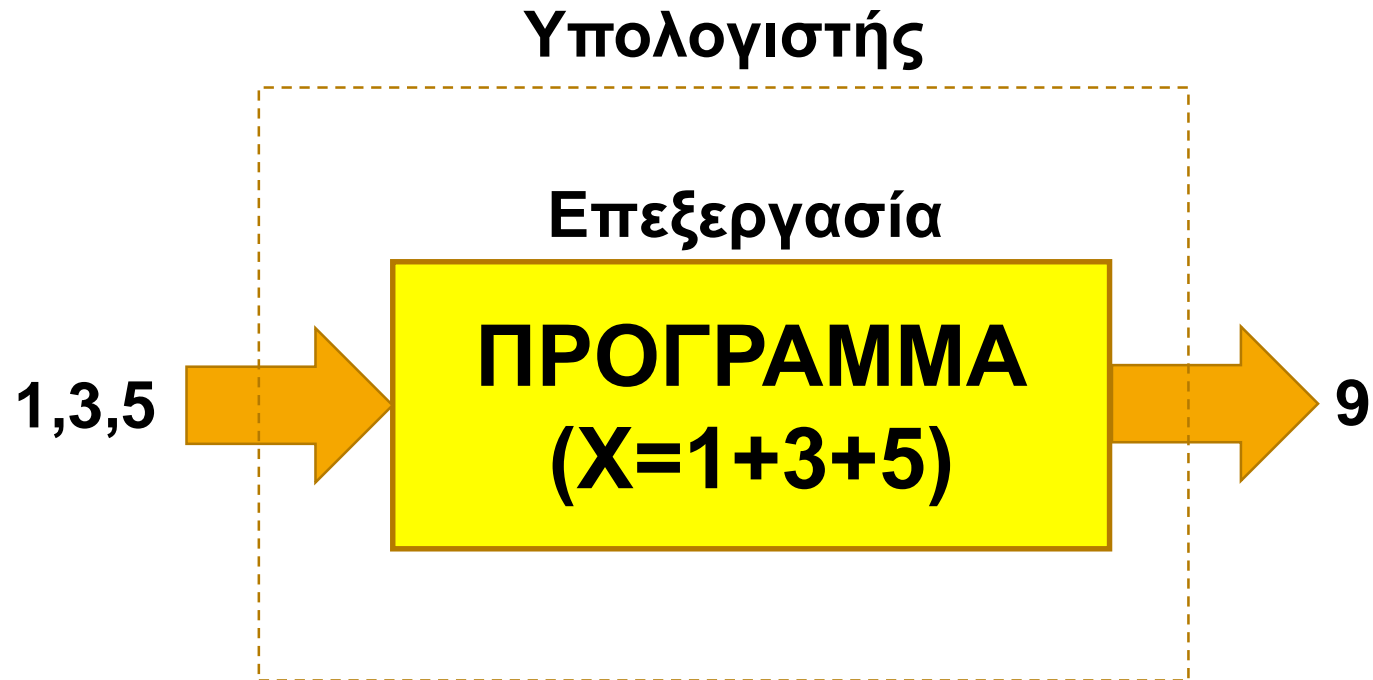
- Η πληροφορική ή αλλιώς η επιστήμη των υπολογιστών, δεν ταυτίζεται με αυτό που ονομάζουμε υπολογιστή
- Ο υπολογιστής είναι απλά το μέσο για να υλοποιήσουμε μια λύση
- Η πληροφορική σχετίζεται με τη σχεδίαση της λύσης, καθώς και με τον τρόπο που θα πρέπει αυτή να υλοποιηθεί
- Όταν μιλάμε λοιπόν για πληροφορική, δεν εννοούμε τους υπολογιστές, αλλά την επιστήμη



Εκτέλεση προγράμματος



Εκτέλεση προγράμματος (παράδειγμα υπολογισμού αθροίσματος)



Πρόγραμμα

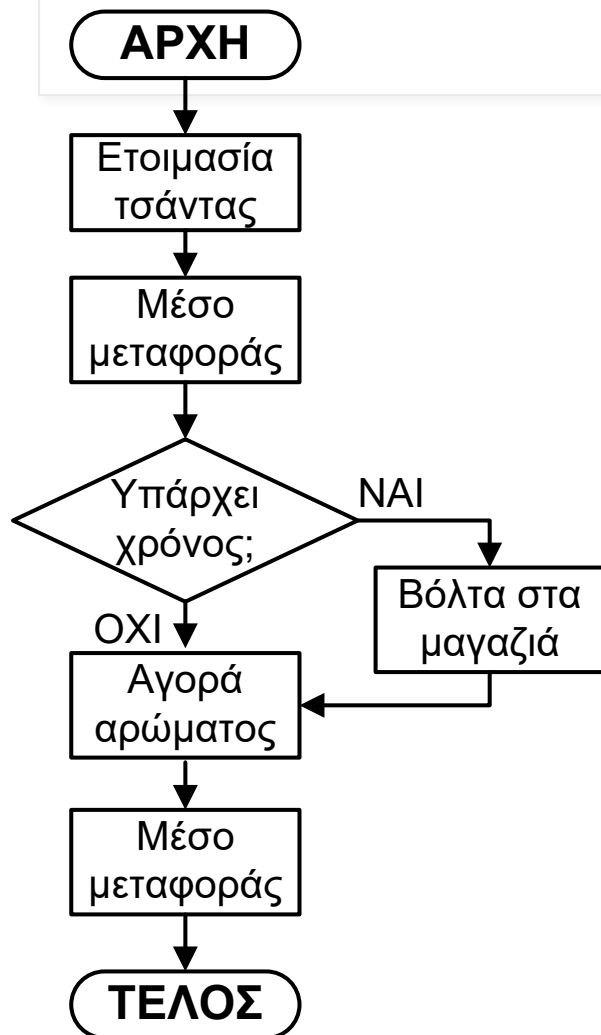
- Ένα σύνολο σαφώς καθορισμένων εντολών
- Η κωδικοποίηση του αλγόριθμου
- Αλγόριθμος = η πεπερασμένη σειρά βημάτων για να λυθεί το πρόβλημα
- Αναπτύσσεται με μια γλώσσα προγραμματισμού
- Κάθε γλώσσα προγραμματισμού έχει τις δικές της εντολές

Προγραμματιστής:

- (α) επιλέγει τη γλώσσα προγραμματισμού
- (β) αναπτύσσει τον κώδικα (εντολές)
- (γ) υλοποιεί τη δομή του προγράμματος
- (δ) ακολουθεί πιστά τον αλγόριθμο



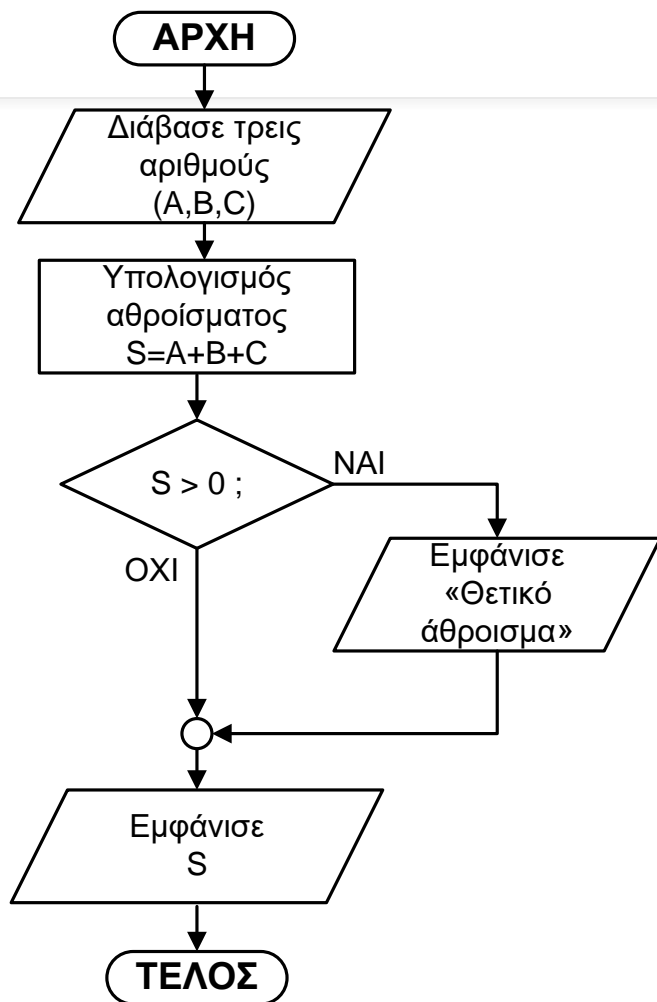
Αλγόριθμος = η πεπερασμένη σειρά βημάτων για να λυθεί το πρόβλημα



- **Πρόβλημα:** Να πάμε για ψώνια
- **Αλγόριθμος:** Τα βήματα που θα ακολουθήσουμε
- **Βήματα:** δεν σχετίζονται με τον τρόπο που θα υλοποιηθούν
- **Επιλογές υλοποίησης (παράδειγμα)**
 - Μέσο μεταφοράς: λεωφορείο ή μετρό ή ταξί
 - Κατάστημα αγοράς
- Πιθανοί περιορισμοί
 - Ελαχιστοποίηση χρόνου μετάβασης = διαφοροποίηση τρόπων υλοποίησης



Πρόσθεση τριών αριθμών & και εμφάνιση αν το άθροισμα είναι θετικό



- **Διάγραμμα ροής:** περιγράφει τον αλγόριθμο, δηλαδή τη «λογική» που θα ακολουθήσουμε
- Ο προγραμματιστής είναι ελεύθερος να επιλέξει όποια γλώσσα προγραμματισμού επιθυμεί, ακολουθώντας όμως πιστά τα βήματα του αλγόριθμου



Υλοποίηση σε γλώσσα C

Για την υλοποίηση σε γλώσσα C, θα χρειαστούμε έναν compiler και ένα περιβάλλον ανάπτυξης κώδικα. Ο compiler, μεταγλωττίζει τον πηγαίο κώδικα (αρχικός κώδικας) στη γλώσσα που καταλαβαίνει ο υπολογιστής, προκειμένου να είναι δυνατή η εκτέλεσή του (στην πραγματικότητα παράγει ένα εκτελέσιμο αρχείο).

Μπορείτε να κατεβάσετε δωρεάν λογισμικό από τις ακόλουθες διευθύνσεις:

<https://www.bloodshed.net/devcpp.html> (για το Dev-c++)

ή

<http://www.codeblocks.org/> (για το Code Blocks)

ή το **visual studio community** της Microsoft

Εναλλακτικά αναζητήστε στο google τα **dev c++** ή **code blocks**



Υλοποίηση σε γλώσσα Python

Μπορείτε να κατεβάσετε δωρεάν την Python από την ακόλουθη διεύθυνση:

<https://www.python.org/downloads/>

Υλοποίηση σε γλώσσα Basic

Μπορείτε να κατεβάσετε δωρεάν την Quick Basic για DOS από την ακόλουθη διεύθυνση:

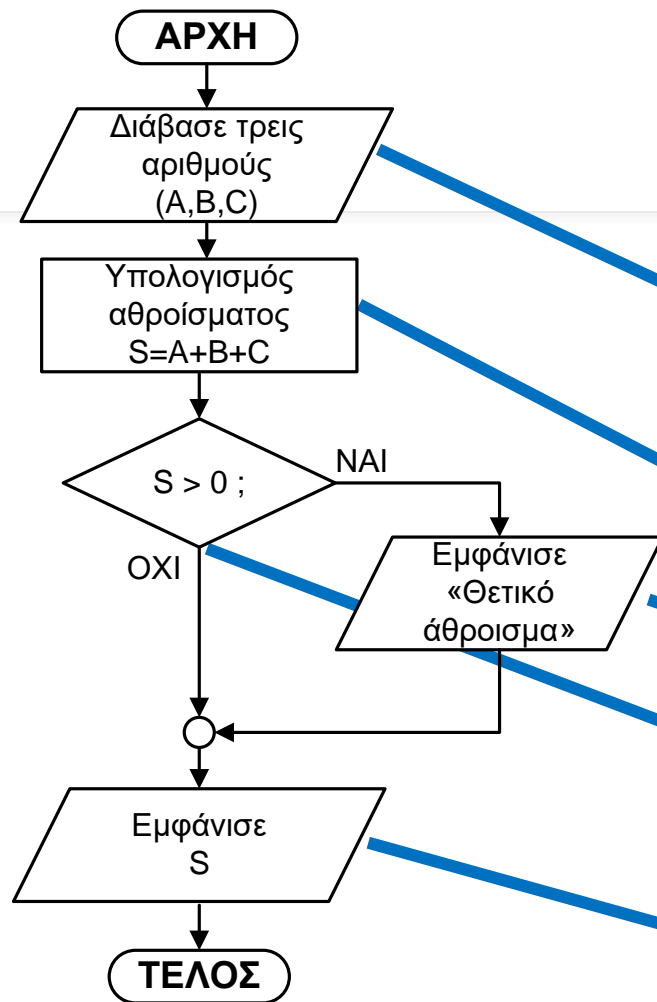
<https://www.qbasic.net/>

Θα την τρέξετε σε περιβάλλον DOS BOX το οποίο θα κατεβάσετε από την ακόλουθη διεύθυνση:

<https://www.dosbox.com/>



C



```
#include <stdio.h>
```

```
int main()  
{
```

```
int a, b, c, s;  
//read numbers
```

```
printf("a="); scanf("%d", &a);  
printf("b="); scanf("%d", &b);  
printf("c="); scanf("%d", &c);
```

```
//sum  
s = a + b + c;
```

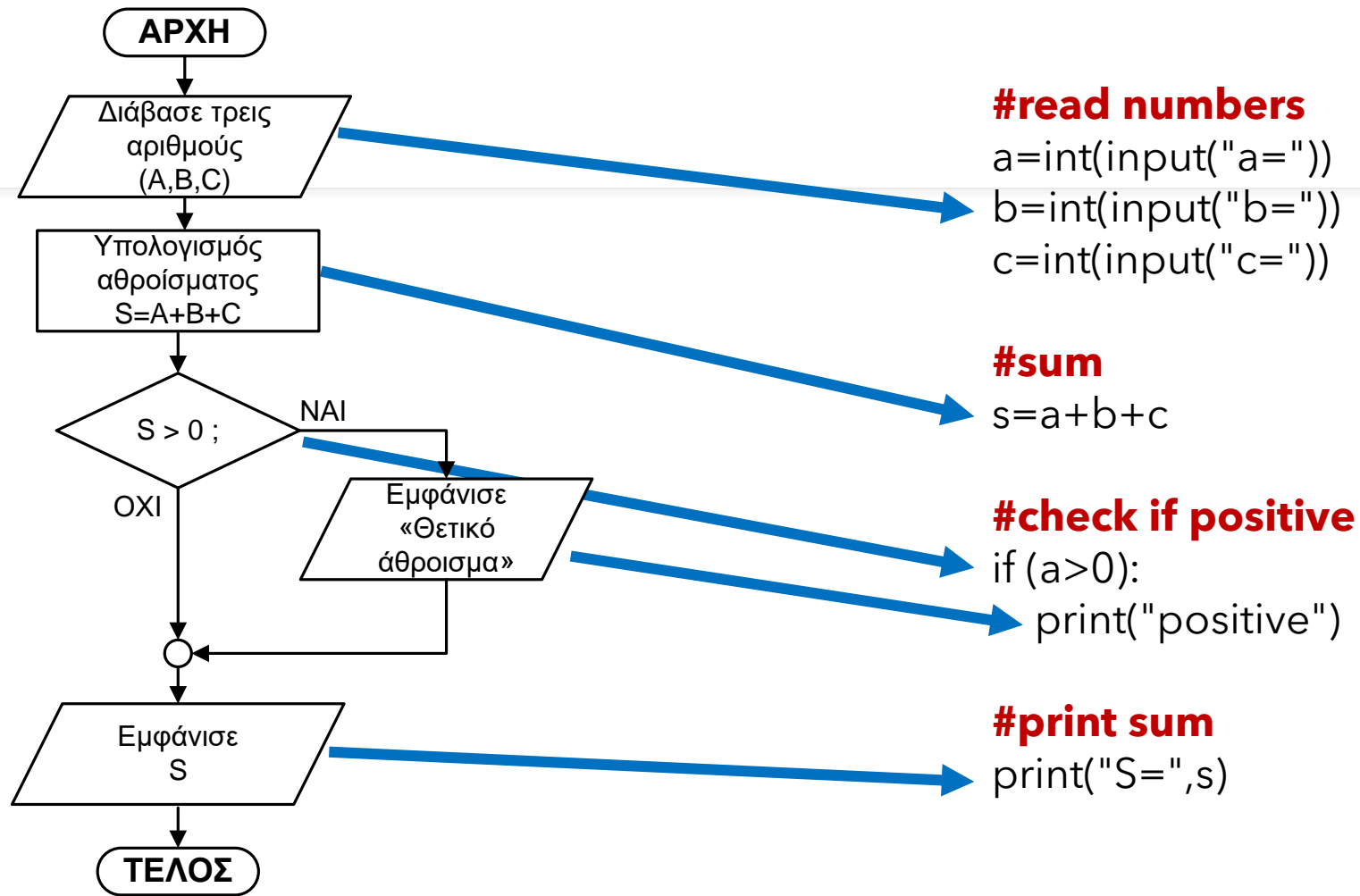
```
//check if positive  
if (a > 0) printf("positive");
```

```
//print sum  
printf("\nS=%d", s);
```

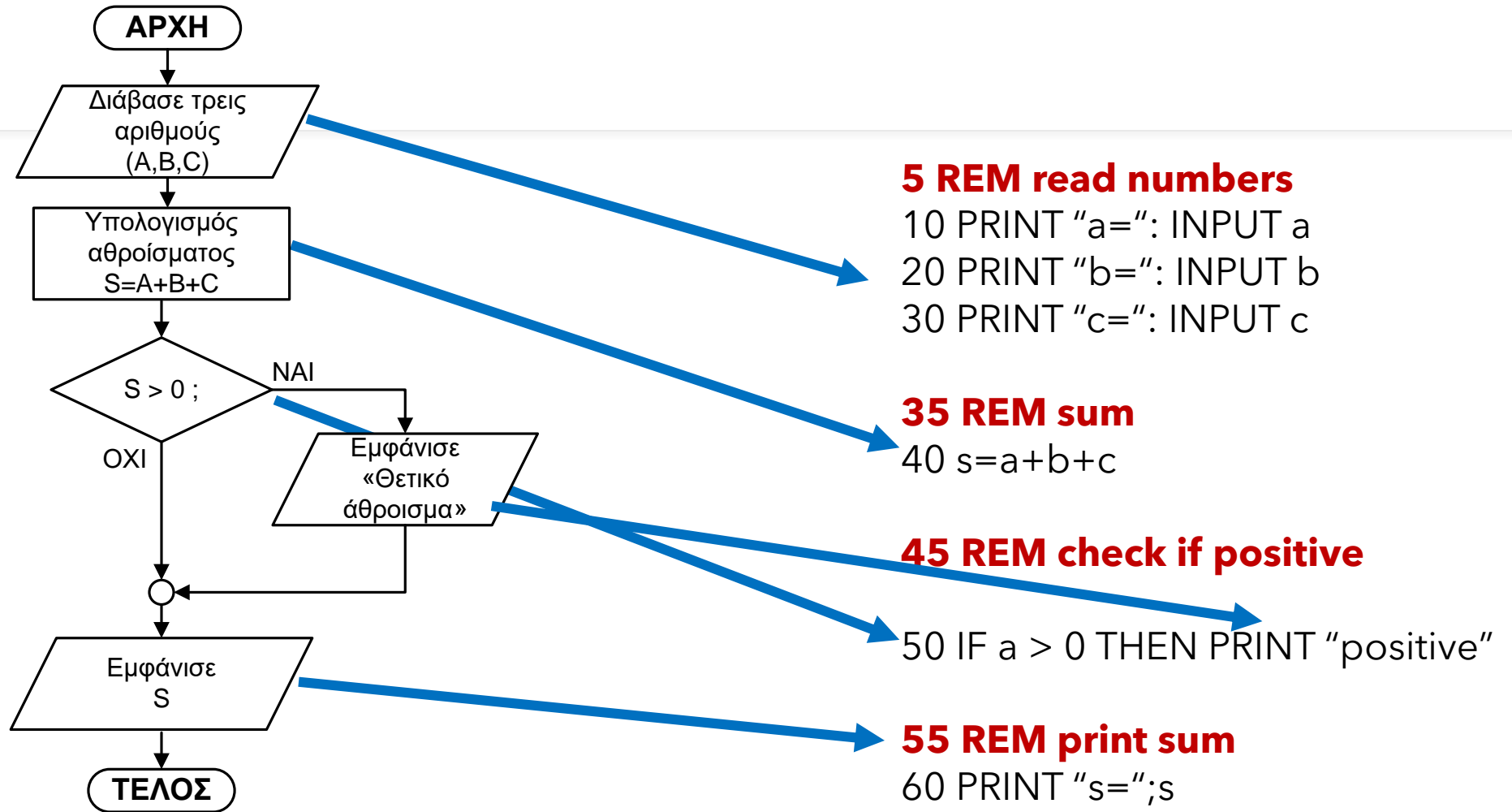
```
}
```



Python



Basic



C

```
#include <stdio.h>

int main()
{

    int a, b, c, s;
    //read numbers
    printf("a="); scanf("%d", &a);
    printf("b="); scanf("%d", &b);
    printf("c="); scanf("%d", &c);

    //sum
    s = a + b + c;

    //check if positive
    if (a > 0)
    printf("positive");

    //print sum
    printf("\nS=%d", s);
}
```

Python

```
#read numbers
a=int(input("a="))
b=int(input("b="))
c=int(input("c="))

#sum
s=a+b+c

#check if positive
if (a>0):
    print("positive")

#print sum
print("S=",s)
```

Basic

5 REM read numbers

```
10 PRINT "a=": INPUT a
20 PRINT "b=": INPUT b
30 PRINT "c=": INPUT c
```

35 REM sum

```
40 s=a+b+c
```

45 REM check if positive

```
50 IF a > 0 THEN PRINT "positive"
```

55 REM print sum

```
60 PRINT "s=";s
```



Δραστηριότητα

Υλοποιήστε τα προηγούμενα προγράμματα στον υπολογιστή. Κάντε τις απαραίτητες ενέργειες για να εγκαταστήσετε το απαραίτητο λογισμικό, αν δεν υπάρχει ήδη στον υπολογιστή.

Μελετώντας τα προηγούμενα προγράμματα, θα πάρετε μια γεύση για το τι είναι ο προγραμματισμός αλλά και για τα αντίστοιχα περιβάλλοντα προγραμματισμού που απαιτούνται.



Προγραμματιστής

(στενή έννοια)

Κωδικοποιεί τον έτοιμο αλγόριθμο σε γλώσσα προγραμματισμού

Προγραμματιστής

(ευρεία έννοια)

Σχεδιάζει τον αλγόριθμο και στη συνέχεια τον κωδικοποιεί σε γλώσσα προγραμματισμού



Αλήθειες

- Ο αλγόριθμος περιγράφει τα βήματα για τη λύση ενός προβλήματος
- Ο αλγόριθμος δεν σχετίζεται απαραίτητα με υπολογιστή
- Ο αλγόριθμος δεν σχετίζεται με τη γλώσσα προγραμματισμού
- Αν γνωρίζω τις εντολές δεν σημαίνει ότι γνωρίζω προγραμματισμό
- Αν δεν μπορώ να αναπτύξω ένα πρόγραμμα, τότε δεν γνωρίζω τον αλγόριθμο



Δομημένος Προγραμματισμός

Οι Bohm και Jacopini, δίνουν μια βάση σε αυτό που γνωρίζουμε ως δομημένο προγραμματισμό, διατυπώνοντας τρία χαρακτηριστικά που είναι αρκετά για να αναπτυχθεί οποιοδήποτε πρόγραμμα. Δηλαδή, τρεις κανόνες για τη σχεδίαση ενός προγράμματος:

Ακολουθία εντολών. Όταν δηλαδή μία εντολή ή μια ενότητα προγράμματος διαδέχεται η μια την άλλη σε επίπεδο εκτέλεσης. Εκτελούνται στη σειρά με την οποία και έχουν αναπτυχθεί.

Έλεγχος. Αυτό σημαίνει την εκτέλεση μιας ή περισσότερων εντολών βάσει μιας συνθήκης που μπορεί να είναι αληθής ή ψευδής.

Επανάληψη. Επαναληπτική εκτέλεση εντολών, όσο ικανοποιείται μια συνθήκη.

Από το 1980 και μετά, άρχισε η εποχή του δομημένου προγραμματισμού, με την ευρεία αποδοχή του. Χαρακτηριστική είναι η επιστημονική εργασία με τίτλο *On Folk Theorems* του Harel το 1980, που δείχνει αυτή την αποδοχή.



Ακολουθία εντολών

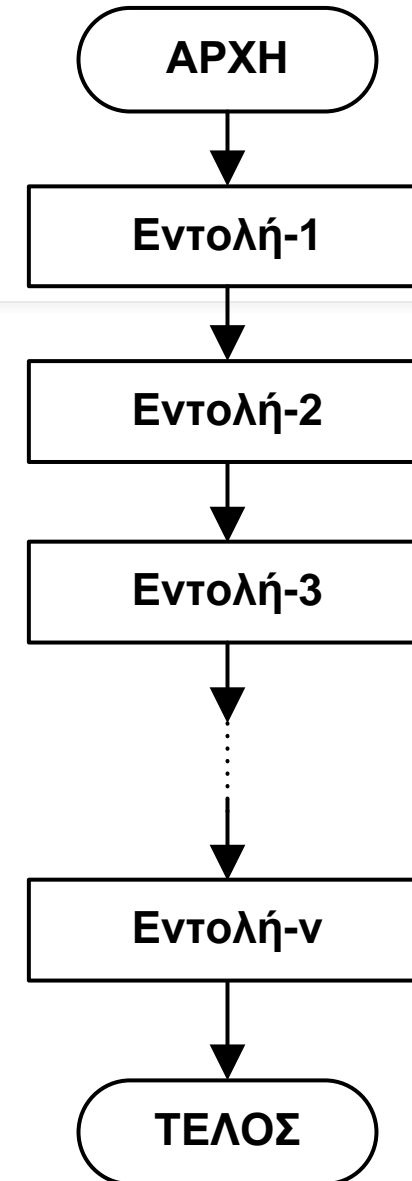
Εντολή-1; (Πρώτη εντολή)

Εντολή-2; (Δεύτερη εντολή)

Εντολή-3; (Τρίτη εντολή)

•
•
•

Εντολή-ν; (ν-οστή εντολή)



Έλεγχος

Αν (συνθήκη)

{

(ΕΝΟΤΗΤΑ T)

Εντολή-T1;

Εντολή-T2;

·

Εντολή-Tn;

}

διαφορετικά

{

(ΕΝΟΤΗΤΑ F)

Εντολή-F1;

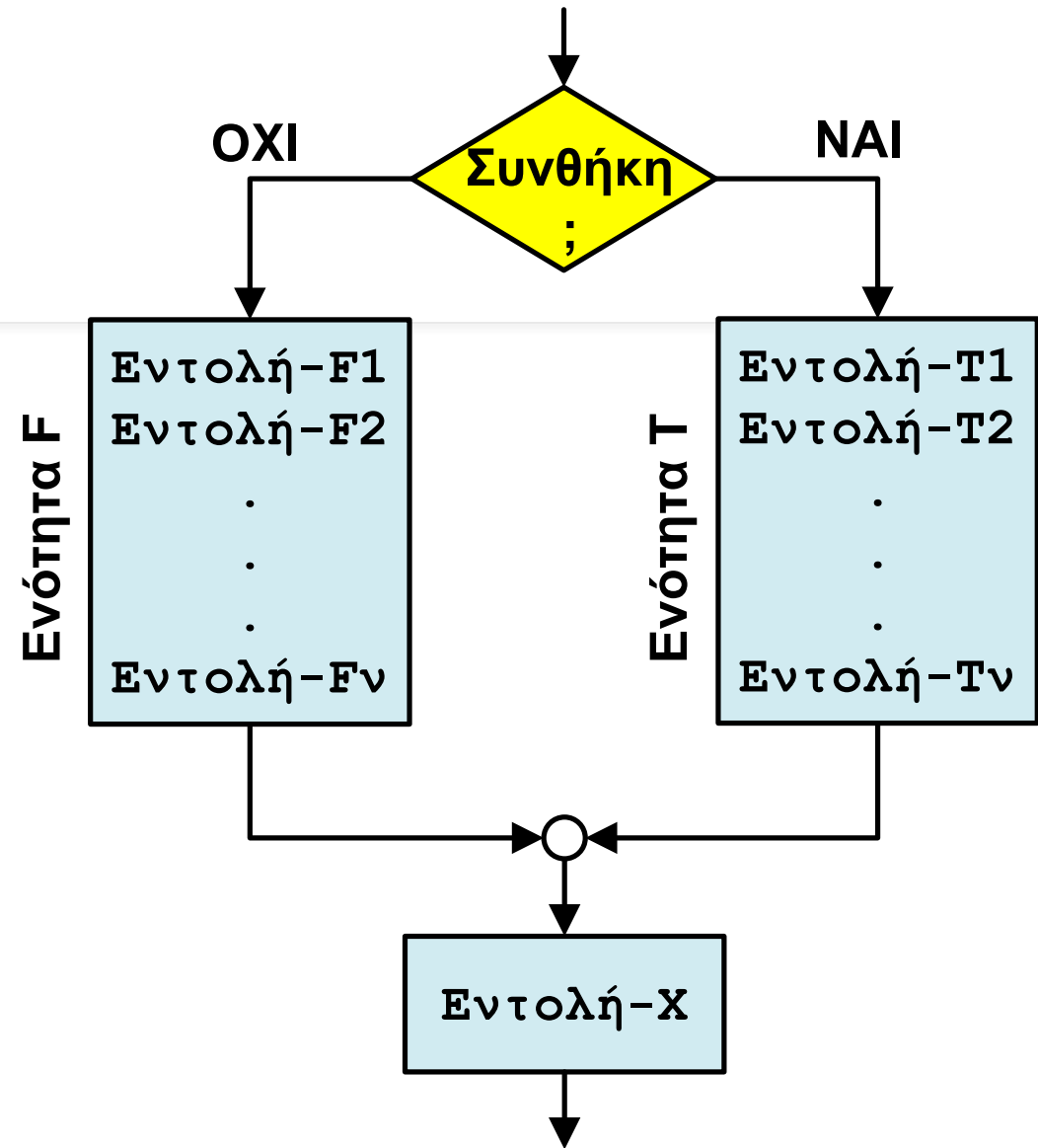
Εντολή-F2;

·

Εντολή-Fn;

}

Εντολή-X;



Επανάληψη

ΓΙΑ μεταβλητή=αρχική_τιμή ΕΩΣ τελική_τιμή

Εντολή-1

Εντολή-2

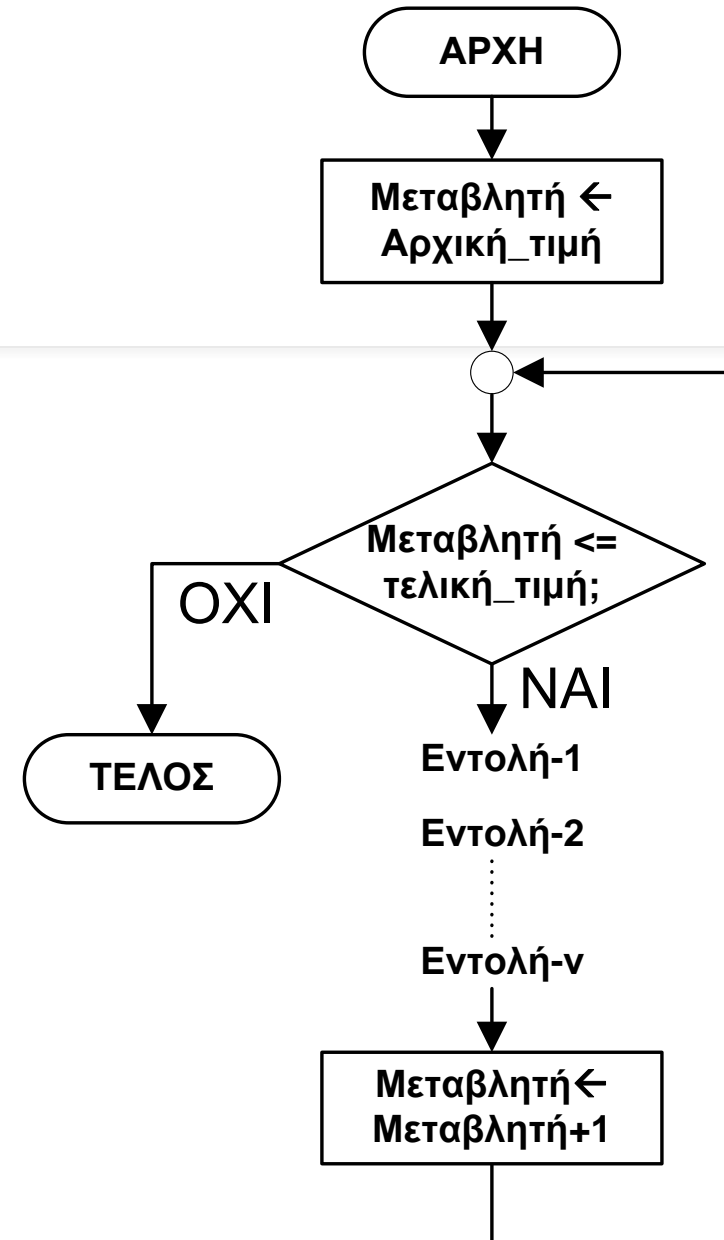
·

·

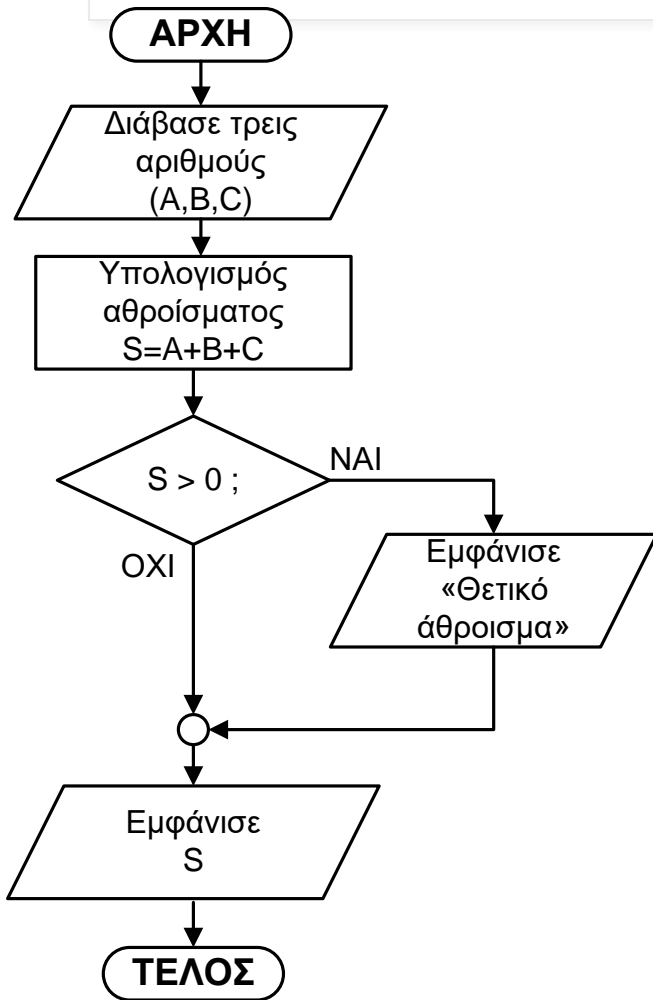
·

Εντολή-ν

Τέλος-ΓΙΑ



Δραστηριότητα



#read numbers

```
a=int(input("a="))  
b=int(input("b="))  
c=int(input("c="))
```

#sum

```
s=a+b+c
```

#check if positive

```
if (a>0):  
    print("positive")
```

#print sum

```
print("S=",s)
```

Μελετήστε το διπλανό παράδειγμα και αναγνωρίστε τους κανόνες του δομημένου προγραμματισμού



**Συνέχεια με ένα ακόμα παράδειγμα για
την κατανόηση του αλγόριθμου**



Αλγόριθμος για ελληνικό καφέ:

- Βήμα 1:** Συγκεντρώνουμε τα υλικά (ζάχαρη, καφές, κουταλάκι, φλυτζάνι, μπρίκι, νερό)
- Βήμα 2:** Βάζουμε νερό, καφέ και ζάχαρη στο μπρίκι
- Βήμα 3:** Ανάβουμε το μάτι
- Βήμα 4:** Ανακατεύουμε τα υλικά
- Βήμα 5:** Ελέγχω αν ο καφές είναι έτοιμος. Αν δεν είναι, γυρίζω στο βήμα 5, διαφορετικά προχωράω στο επόμενο βήμα
- Βήμα 6:** Κλείνω το μάτι της κουζίνας, βάζω στο φλυτζάνι τον καφέ και σερβίρω



Περιγραφή με ψευδοκώδικα

Αλγόριθμος ΚΑΦΕ

Μεταβλητές

ζάχαρη, καφές, κουτάλι, φλυτζάνι, μπρίκι, νερό

ΑΡΧΗ

Βάλε νερό, καφέ και ζάχαρη στο μπρίκι

Άναψε το μάτι της κουζίνας

Ανακάτεψε τα υλικά

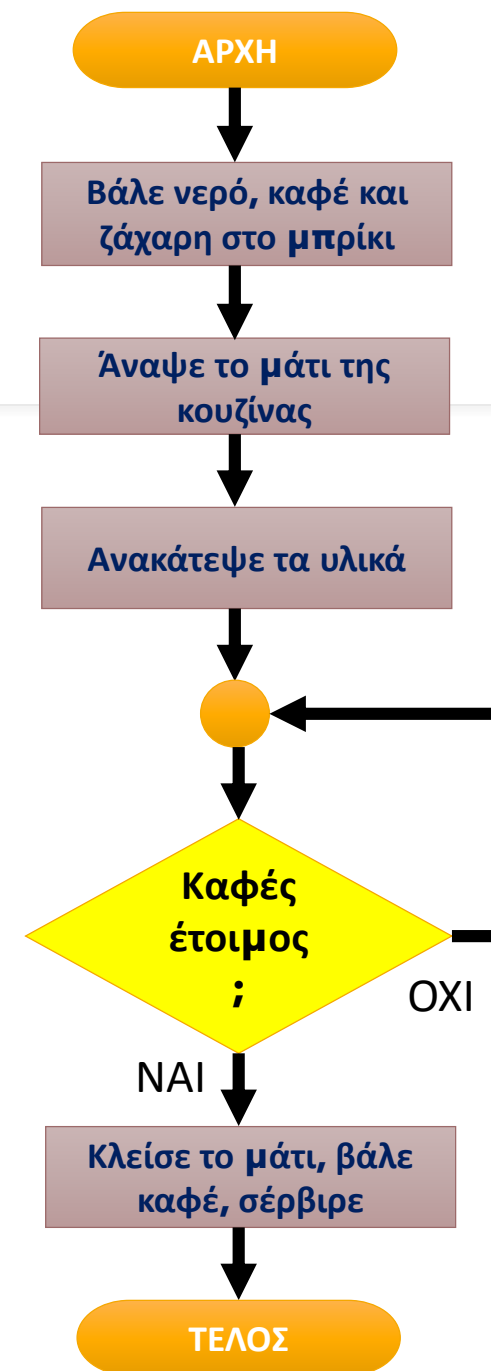
A: Αν ο καφές είναι έτοιμος, τότε πήγαινε στο B
διαφορετικά πήγαινε στο A

B: Κλείσε το μάτι της κουζίνας, βάλε στο φλυτζάνι τον καφέ και σέρβιρε

ΤΕΛΟΣ

Ο ψευδοκώδικας είναι μια πιο αυστηρή περιγραφή του αλγόριθμου, που πλησιάζει περισσότερο τη μορφή του πραγματικού κώδικα που θα αναπτυχθεί

- Περιορίζει τα σφάλματα στην κωδικοποίηση
- Πιο πρακτικός για σύνθετους αλγόριθμους



Περιγραφή με διάγραμμα ροής



**Περιγράψτε τον αλγόριθμο για
την ανάληψη χρημάτων από το
ΑΤΜ**

**Περιγράψτε τον αλγόριθμο για τη
λύση ενός προβλήματος της
επιλογής σας**



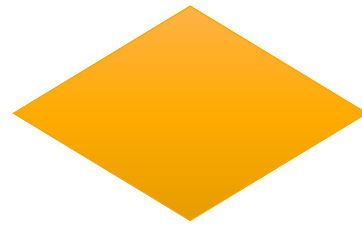
Βασικά Σύμβολα για τη σχεδίαση διαγράμματος ροής



Αρχή ή τέλος διαγράμματος (λέξεις: ΑΡΧΗ ή ΤΕΛΟΣ)



Επεξεργασία (π.χ. εκχώρηση τιμής, αριθμητική πράξη)



Έλεγχος συνθήκης (π.χ. αλλαγή ροής εκτέλεσης βάσει συνθήκης που μπορεί να είναι αληθής ή ψευδής)



Μετάβαση σε άλλο σημείο της ίδιας σελίδας (π.χ. εδώ καταλήγουν πολλαπλά βέλη)



Ροή εκτέλεσης (κατεύθυνση)

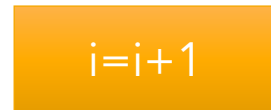
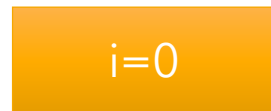


Είσοδος/Εξοδος (π.χ. διάβασμα από πληκτρολόγιο ή εμφάνιση στην οθόνη)

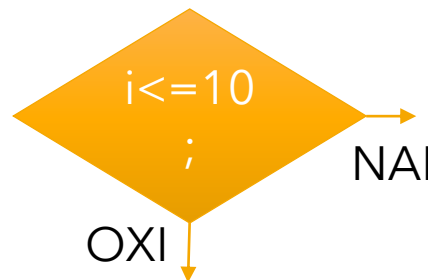
Παράδειγμα



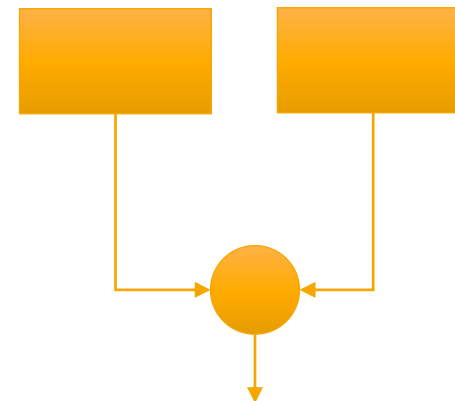
Παράδειγμα



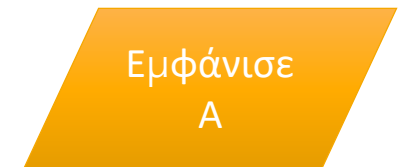
Παράδειγμα



Παράδειγμα



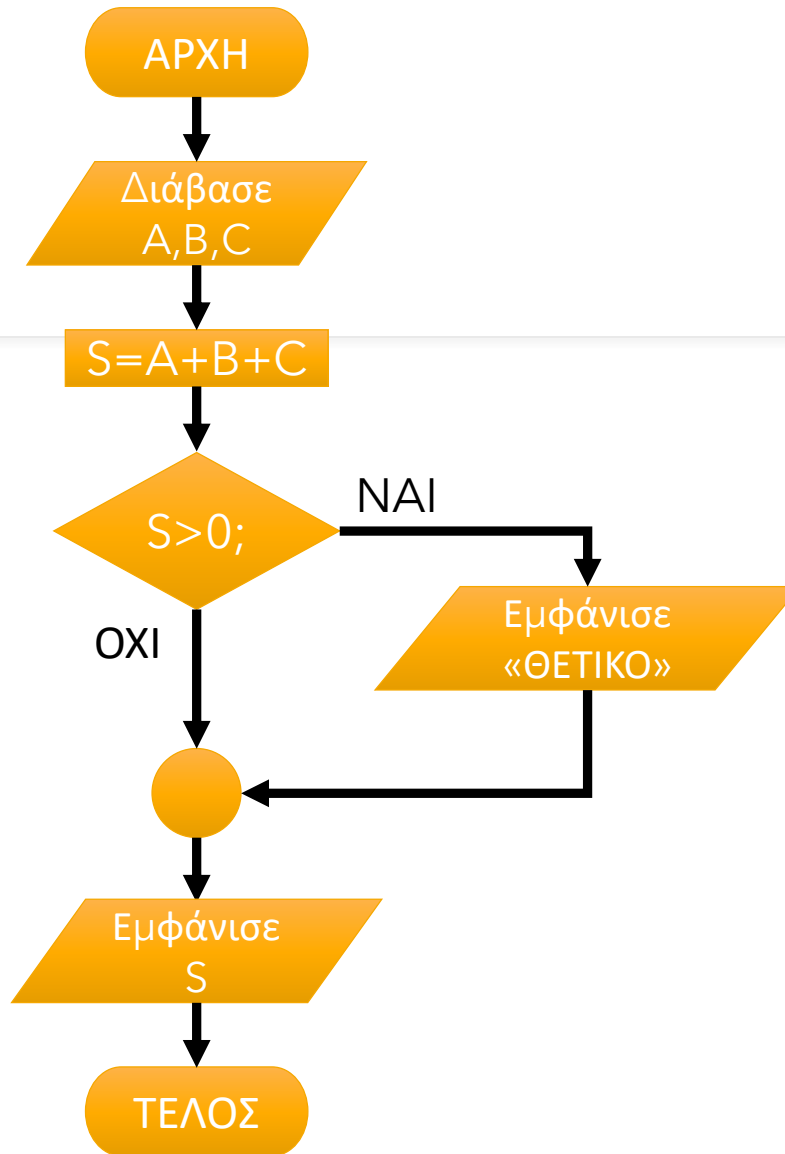
Παράδειγμα

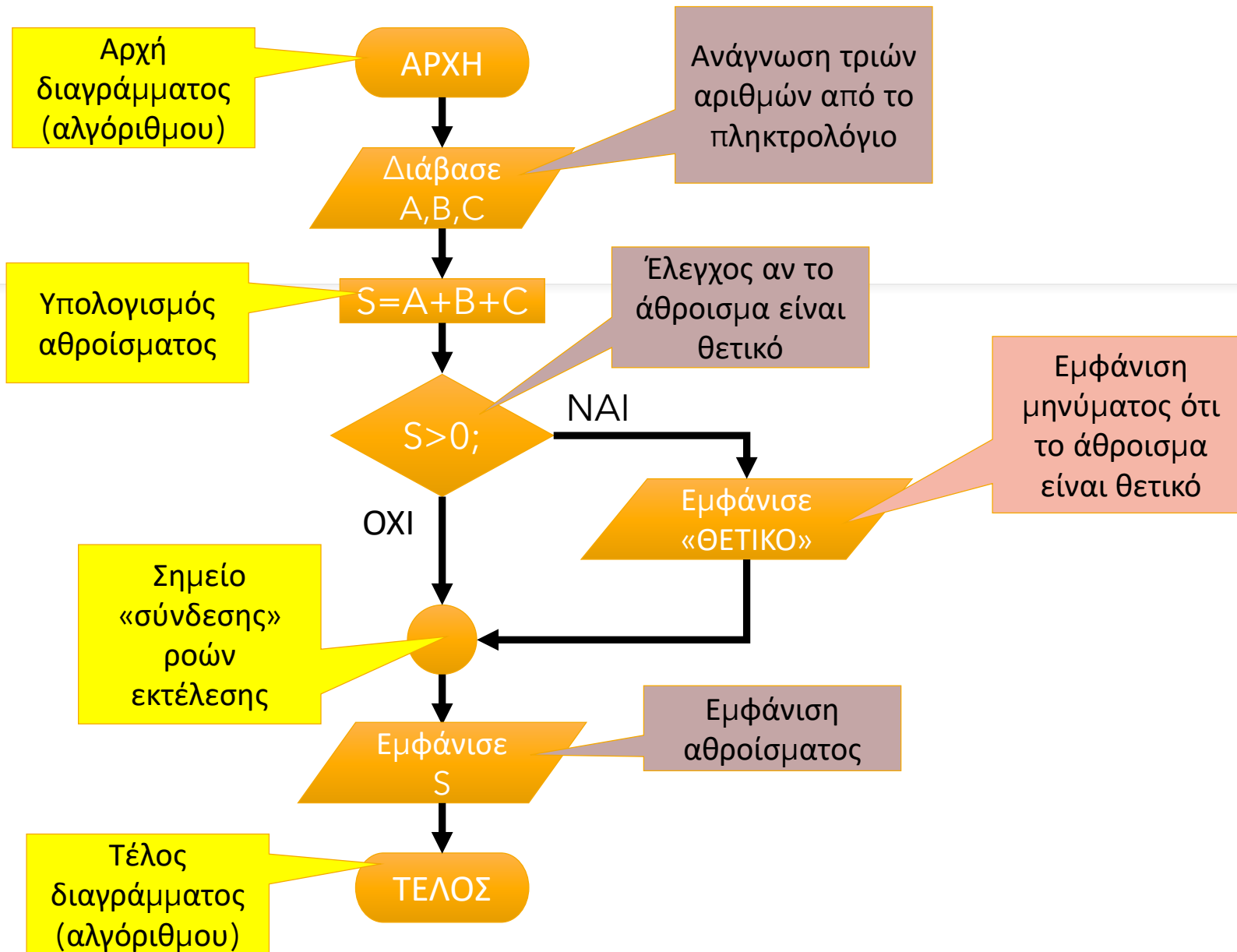


Άσκηση

Να σχεδιαστεί διάγραμμα ροής για ένα πρόγραμμα που θα υπολογίζει το άθροισμα τριών αριθμών. Αν το άθροισμα είναι θετικό, τότε θα εμφανίζεται ένα σχετικό μήνυμα, διαφορετικά απλά θα τερματίζεται το πρόγραμμα. Σε κάθε περίπτωση, θα εμφανίζεται το άθροισμα.







Άσκηση

Να αναπτυχθεί ψευδοκώδικας και διάγραμμα ροής για ένα πρόγραμμα που μετατρέπει τη θερμοκρασία από βαθμούς Φαρενάιτ σε βαθμούς Κελσίου

$$C = 5 * (F - 32) / 9$$



Λύση

Διάγραμμα ροής

Ψευδοκώδικας

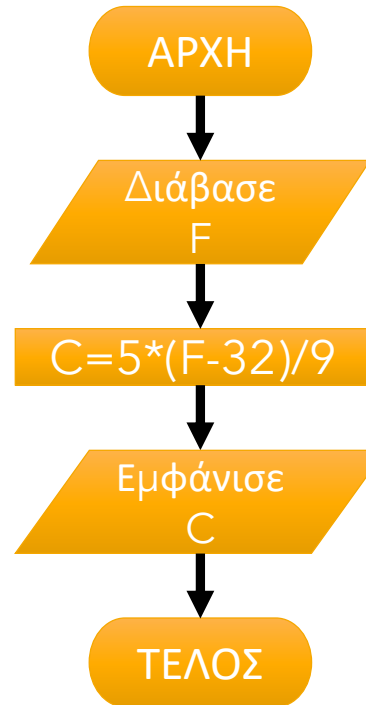
ΑΡΧΗ

Διάβασε F

$C = 5 * (F - 32) / 9$

Εμφάνισε C

ΤΕΛΟΣ



Λύση

Ένας πιο τυπικός Ψευδοκώδικας

Ψευδοκώδικας

ΑΡΧΗ

Διάβασε F

$C = 5 * (F - 32) / 9$

Εμφάνισε C

ΤΕΛΟΣ

ΑΛΓΟΡΙΘΜΟΣ ΜΕΤΑΤΡΟΠΗΣ $F \rightarrow C$

ΔΕΔΟΜΕΝΑ

F, C: REAL;

ΑΡΧΗ

Διάβασε (F);

$C := 5 * (F - 32) / 9;$

Εμφάνισε (C)

ΤΕΛΟΣ



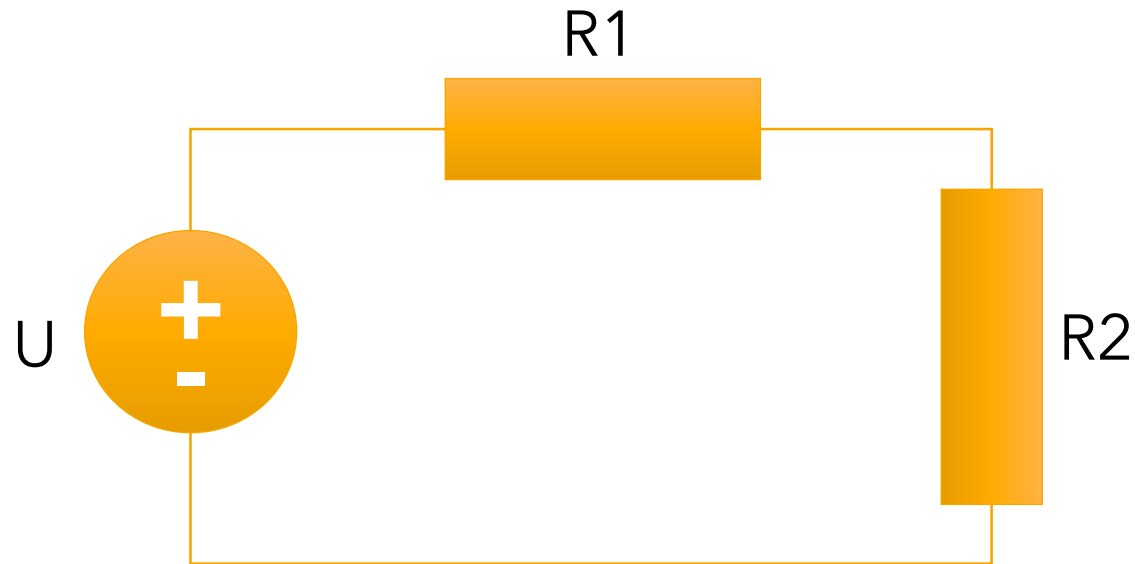
Ασκήσεις

1. Να αναπτυχθεί ψευδοκώδικας και διάγραμμα ροής για ένα πρόγραμμα που εμφανίζει τον μικρότερο από δύο αριθμούς
2. Να αναπτυχθεί ψευδοκώδικας και διάγραμμα ροής για ένα πρόγραμμα που υπολογίζει το άθροισμα τριών αριθμών και εμφανίζει σχετικό μήνυμα αν το αποτέλεσμα είναι αρνητικό, μηδέν ή θετικό
3. Να αναπτυχθεί ψευδοκώδικας και διάγραμμα ροής για ένα πρόγραμμα που υπολογίζει το εμβαδό ενός ορθογώνιου παραλληλόγραμμου. Αν το εμβαδό είναι μεγαλύτερο από 10 (υποθετικές μονάδες) θα εμφανίζεται ένα σχετικό μήνυμα.
4. Να αναπτυχθεί ψευδοκώδικας και διάγραμμα ροής για ένα πρόγραμμα που υπολογίζει τον όγκο δύο διαφορετικών κύβων και να εμφανίζει ποιος κύβος έχει τον μικρότερο όγκο



Ασκήσεις

5. Να αναπτυχθεί ψευδοκώδικας και διάγραμμα ροής για ένα πρόγραμμα που υπολογίζει την ισχύ στις αντιστάσεις του κυκλώματος και εμφανίζει μήνυμα με την αντίσταση που έχει τη μέγιστη ισχύ.



Ασκήσεις

6. Να αναπτυχθεί ψευδοκώδικας και διάγραμμα ροής για ένα πρόγραμμα που υπολογίζει και εμφανίζει το μέσο όρο 10 αριθμών που εισάγονται από το πληκτρολόγιο (χωρίς πίνακα)

7. Να αναπτυχθεί ψευδοκώδικας και διάγραμμα ροής για ένα πρόγραμμα που υπολογίζει και εμφανίζει το μέσο όρο 10 αριθμών που εισάγονται από το πληκτρολόγιο (με πίνακα)

8. Να αναπτυχθεί ψευδοκώδικας και διάγραμμα ροής για ένα πρόγραμμα που εμφανίζει τον ελάχιστο αριθμό σε ένα πίνακα ακεραίων 10 θέσεων που γεμίζει από το πληκτρολόγιο



Σχεδιάζοντας γραφικά

Ας υποθέσουμε ότι, πρόκειται να σχεδιάσουμε ένα κύκλο. Σε αυτό το παράδειγμα, θα διαχωρίσουμε την πληροφορική (επιστήμη) από τους υπολογιστές.



Σχεδιάζοντας γραφικά

Βήμα 1

Ο μαθηματικός σχεδιάζει και αναλύει τριγωνομετρικά τον κύκλο, σημειώνοντας παράλληλα τα σημεία ενδιαφέροντος

Οι συντεταγμένες του σημείου σ_{xy} , είναι

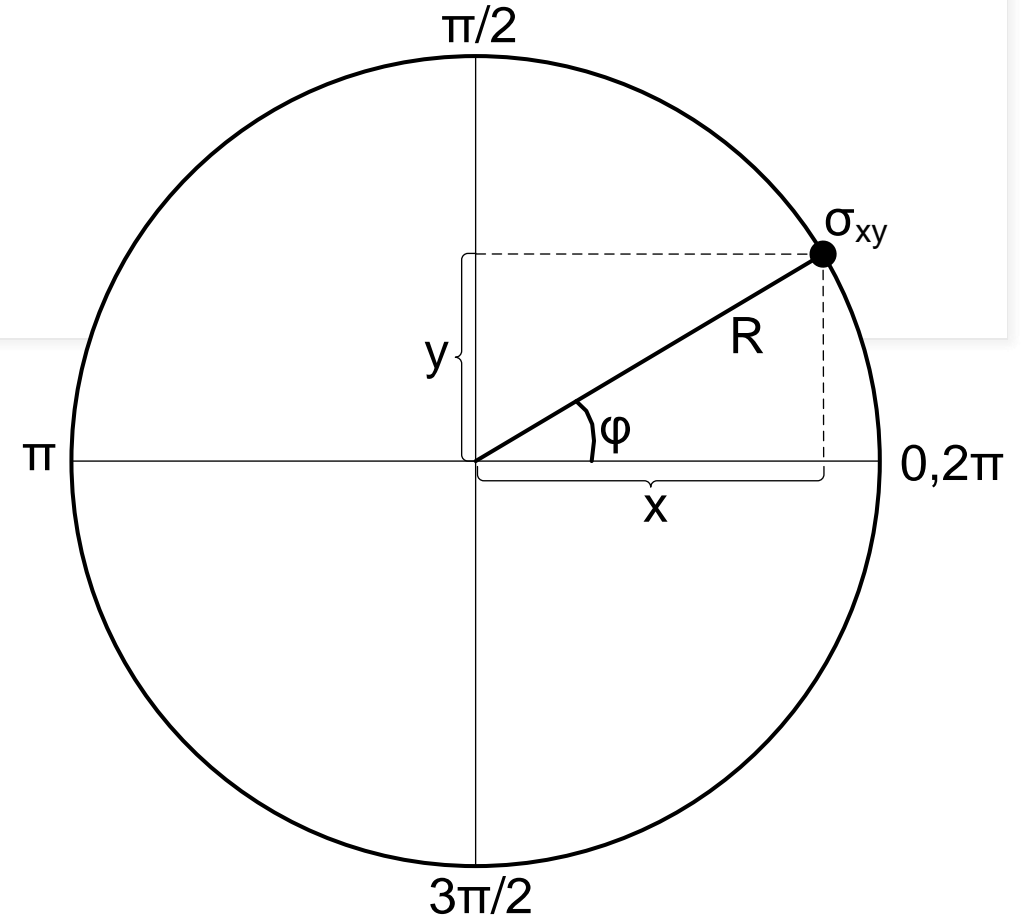
$$(x_0 + R * \cos \varphi), (y_0 + R * \sin \varphi)$$

Όπου

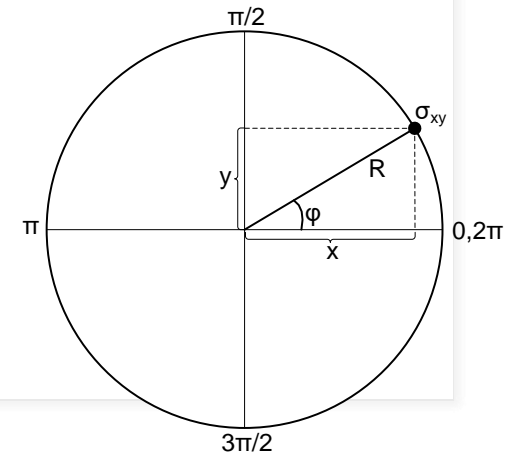
x_0, y_0 : το κέντρο του κύκλου

R : η ακτίνα

φ : η γωνία (0 έως 2π)



Σχεδιάζοντας γραφικά



Βήμα 3

Ο προγραμματιστής (π.χ. μηχανικός πληροφορικής) υλοποιεί τον κώδικα βάσει του αλγόριθμου, χρησιμοποιώντας έναν υπολογιστή. Είναι δική του επιλογή η γλώσσα προγραμματισμού.

Όπως φαίνεται παραπάνω, ο υπολογιστής δεν έχει άμεση σχέση με την επιστήμη της πληροφορικής!



Υλοποίηση σε Processing

Κατεβάζουμε το Processing από τη διεύθυνση <https://processing.org/>

Το Processing «παράγει» άμεσα αποτελέσματα, δηλαδή αναπτύσσουμε τον κώδικα και στη συνέχεια πατάμε Run. Δεν χρειάζονται άλλες προϋποθέσεις για να τρέξει το πρόγραμμα που θα αναπτύξουμε.

Αλγόριθμος

$X_0 = X$ κέντρου, $Y_0 = Y$ κέντρου

$R =$ ακτίνα

Για i από 0 έως 2π , βήμα 0.01

Σχεδίαση σημείου στη θέση $X_0 + R * \cos(i)$, $Y_0 + R * \sin(i)$

Τέλος-Για

void setup()

```
{
  size(500, 500);
  surface.setTitle("Κύκλος");
  background(255,255,255);
  noLoop();
}
```

void draw()

```
{
  float i;
  int R=200;
  int X=250, Y=250;
  for (i=0;i<=2*PI;i+=0.01)
    set(round(X+R*cos(i)),round(Y+R*sin(i)), color(0));
}
```

Κώδικας Processing



Υλοποίηση σε Processing

Ο προγραμματιστής παρατήρησε ότι, οι συντεταγμένες του σημείου στον αλγόριθμο παράγουν αριθμούς με υποδιαστολή. Όμως, οι συντεταγμένες στην οθόνη είναι ακέραιοι αριθμοί. Έτσι, αποφάσισε να χρησιμοποιήσει τη συνάρτηση `round()` που κάνει στρογγυλοποίηση στον πλησιέστερο ακέραιο. Έτσι, τα ορίσματα της εντολής `set` είναι ακέραιοι αριθμοί και το πρόγραμμα μπορεί να δουλέψει κανονικά.

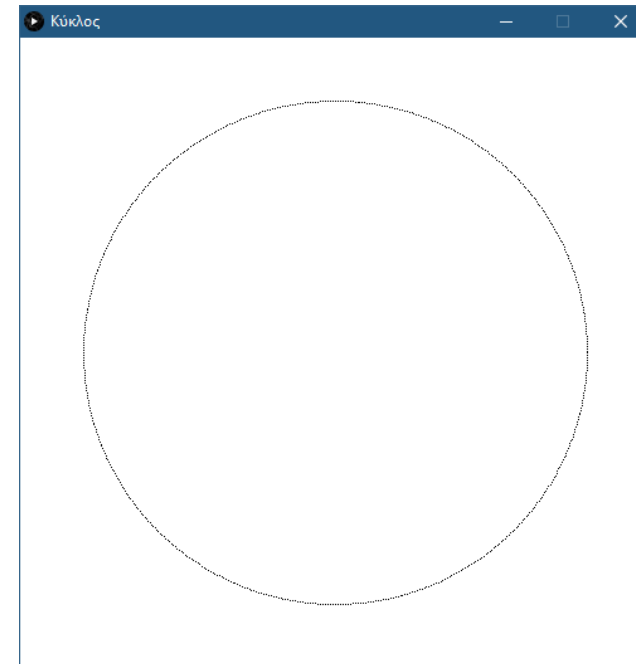
Αλγόριθμος

$$X0 + R * \cos(i), Y0 + R * \sin(i)$$

Κώδικας Processing

```
set(round(X+R*cos(i)),round(Y+R*sin(i)), color(0));
```

Αποτέλεσμα στην οθόνη



Υλοποίηση σε υπολογιστή της δεκαετίας του 1980

Το 1977 κυκλοφόρησε ο Apple II, ο πρώτος προσωπικός υπολογιστής μαζικής παραγωγής στον κόσμο. Τα επόμενα χρόνια, κυκλοφόρησαν διάφορα μοντέλα της σειράς II.



Apple IIe (εικόνα Π. Παπάζογλου)

Ανοίγουμε έναν browser και επισκεπτόμαστε τη διεύθυνση

<https://www.scullinsteel.com/apple2/>

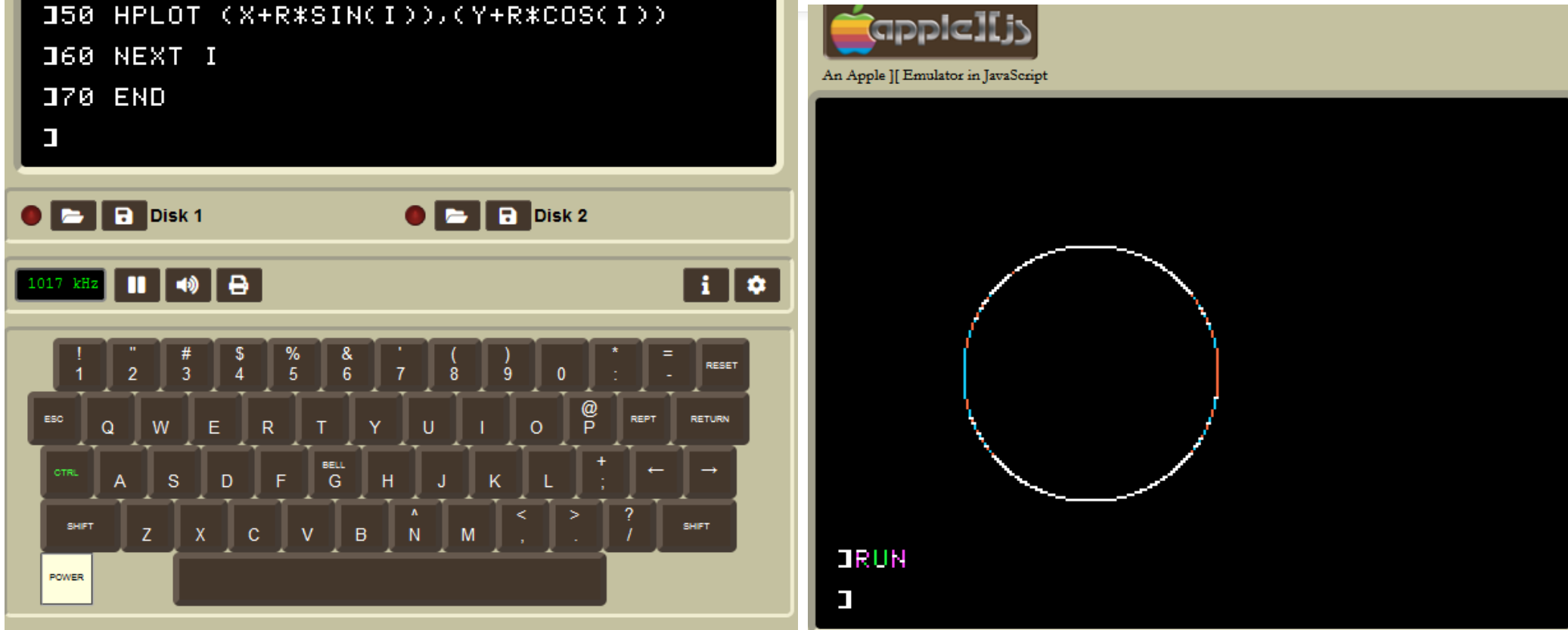
Εκεί θα πληκτρολογήσουμε το πρόγραμμα για τη σχεδίαση του κύκλου σε AppleSoft Basic (εικονικό περιβάλλον υπολογιστή Apple II, 1977) και θα διαπιστώσουμε ότι ο αλγόριθμος παραμένει ο ίδιος, ανεξάρτητα της τεχνολογίας υλοποίησης



Υλοποίηση σε υπολογιστή της δεκαετίας του 1980

```
110 HGR
120 X=100:Y=100
130 R=50
140 FOR I=0 TO 6.28 STEP 0.01
150 HPLLOT (X+R*SIN(I)),(Y+R*COS(I))
160 NEXT I
170 END
1
```

Πατάμε το πλήκτρο Reset (εικονικό πληκτρολόγιο) για να ξεκινήσουμε και πληκτρολογούμε το διπλανό πρόγραμμα. Τέλος, γράφουμε RUN και πατάμε <Enter> για να δούμε το αποτέλεσμα.



Π. ΠΑΠΑΖΟΓΛΟΥ

Οι διαφοροποιήσεις στον κώδικα οφείλονται στο συντακτικό της γλώσσας Applesoft Basic, καθώς και στο αντίστοιχο σύστημα συντεταγμένων του συγκεκριμένου υπολογιστή.



Στο υποστηρικτικό υλικό του βιβλίου θα βρείτε πλήρες set εισαγωγικών διαφανειών για προγραμματισμό σε γλώσσα C

