

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΠΟΛΙΤΙΣΜΟΥ

ΠΛΗΡΟΦΟΡΙΚΗ

ΣΗΜΕΙΩΣΕΙΣ

Σχεδίαση Λογικών Διαγραμμάτων

Προκαταρκτική Εκτέλεση Λογικού Διαγράμματος και Προγράμματος

Σεπτέμβριος 2017

ΠΕΡΙΕΧΟΜΕΝΑ

1. Πρόλογος	3
2. Λογικά Διαγράμματα.....	4
2.1 Δομημένος Προγραμματισμός	4
2.2 Σύμβολα Λογικών Διαγραμμάτων	5
2.3 Εισηγήσεις για ομοιόμορφο σχεδιασμό λογικών διαγραμμάτων	6
2.4 Δομές Προγραμματισμού	8
2.4.1 Ακολουθιακή Δομή	8
2.4.2 Δομή Διακλάδωσης	9
2.4.3 Επαναληπτική Δομή.....	10
3. Προκαταρκτική Εκτέλεση.....	14
3.1 Δομές Προγραμματισμού	15
3.1.1 Ακολουθιακή Δομή	15
3.1.2 Δομή Διακλάδωσης.....	16
3.1.3 Επαναληπτική Δομή.....	19
3.4 Υποπρογράμματα	22
3.4.1 Συναρτήσεις που επιστρέφουν μόνο μια τιμή (εντολή return).....	22
3.4.2 Συναρτήσεις με χρήση παραμέτρων με αναφορά μόνο	24
3.4.3 Συναρτήσεις με χρήση παραμέτρων με αναφορά και την εντολή return	25
3.4.4 Συναρτήσεις που ΔΕΝ έχουν ούτε τυπικές παράμετρος αναφοράς ούτε την εντολή return αλλά το αποτέλεσμα τυπώνεται εντός της συνάρτησης	27
3.5 Μονοδιάστατοι Πίνακες	28
3.6 Πίνακες δύο Διαστάσεων	30
4. Βιβλιογραφία	35

Συνάδελφοι,

Οι σημειώσεις αυτές έχουν γραφτεί για να καλύψουν τις ανάγκες του μαθήματος της Πληροφορικής σε Γυμνάσιο και Λύκειο. Έχουν ως στόχο να μας βοηθήσουν να χρησιμοποιήσουμε όλοι ομοιόμορφες μεθόδους στο σχεδιασμό λογικών διαγραμμάτων αλλά και στην προκαταρκτική εκτέλεση των προγραμμάτων ή και των λογικών διαγραμμάτων. Οι σημειώσεις αποτελούνται από δύο ενότητες με παραδείγματα.

Η πρώτη ενότητα αφορά το σχεδιασμό των λογικών διαγραμμάτων, διαδικασία αναγκαία στην ανάπτυξη των προγραμμάτων. Τα σύμβολα που χρησιμοποιήθηκαν ακολουθούν το πρότυπο ANSI - Flowchart Symbols.

Η δεύτερη ενότητα αφορά την προκαταρκτική εκτέλεση προγραμμάτων ή / και λογικών διαγραμμάτων.

Κατά την προκαταρκτική εκτέλεση θα πρέπει να δίνεται ιδιαίτερη έμφαση στην πιστή εκτέλεση όλων των εντολών του προγράμματος ή και του λογικού διαγράμματος. Θα πρέπει να καταγράφονται αναλυτικά όλες οι σταθερές, οι μεταβλητές, οι συνθήκες και οι αντίστοιχες τιμές τους, ώστε να βοηθηθούν οι μαθητές στην κατανόηση της ροής και των αλλαγών των δεδομένων στο πρόγραμμα ή και στο λογικό διάγραμμα.

Στην προσπάθεια μας να συμβαδίσουμε με τα υφιστάμενα εγχειρίδια του Γυμνασίου και του Λυκείου χρησιμοποιήσαμε μεθοδολογία από διάφορες έγκυρες πηγές (βλέπε βιβλιογραφία), χωρίς να χάνεται η συνοχή και η ομοιομορφία. Βέβαια δεν αποκλείονται άλλες προσεγγίσεις παρουσίασης των λογικών διαγραμμάτων και προκαταρκτικών εκτελέσεων νοουμένου ότι είναι επιστημονικά αποδεκτές.

Συγγραφική ομάδα:

Μιλτιάδου Μάριος, Β.Δ.

Σχίζα Ιουλία, Β.Δ.

Σωτικόπουλος Κωνσταντίνος, Β.Δ.

Τορτούρης Μιχάλης, Β.Δ.

Εποπτεία:

Χατζησάββας Γιώργος, ΕΜΕ

Αναθεώρηση για την C++:

Άκης Συκοπετρίτης, Καθηγητής Πληροφορικής

Κωνσταντίνος Σωφρονίου, Σύμβουλος Πληροφορικής

Λογικά Διαγράμματα

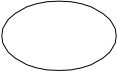


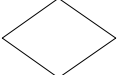


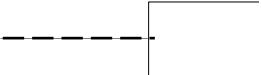
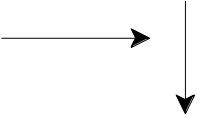
Δομημένος Προγραμματισμός

Ο δομημένος προγραμματισμός είναι η τεχνική που διαιρεί το πρόγραμμα σε λογικά τμήματα χρησιμοποιώντας βασικές δομές. Η τεχνική αυτή μας βοηθά στην κωδικοποίηση, έλεγχο, και συντήρηση ενός προγράμματος. Στην παραγωγή λογισμικού, ο δομημένος προγραμματισμός χρησιμοποιείται ως βασική αρχή προγραμματισμού. Βασικός στόχος του δομημένου προγραμματισμού είναι η παραγωγή προγραμμάτων που έχουν μια καθορισμένη μορφή και γι' αυτό γίνονται πιο εύκολα κατανοητά, όχι μόνο από το δημιουργό τους αλλά και από άλλους προγραμματιστές που στο μέλλον θα ασχοληθούν με τα προγράμματα αυτά.

Ο δομημένος προγραμματισμός χρησιμοποιεί μόνο τρεις βασικές δομές ελέγχου που είναι:

1. Η Ακολουθιακή Δομή (Sequence)
2. Η Δομή Διακλάδωσης (Selection) και η επέκτασή της, η Περιπτωσιακή Δομή (Switch)
3. Η Δομή Επανάληψης (Iteration)

Σύμβολα Λογικών Διαγραμμάτων

Σύμβολο	Επεξήγηση
	Αρχή - Τέλος Κυρίως Προγράμματος Είσοδος - Έξοδος Υποπρογραμμάτων
	Επεξεργασία
	Είσοδος Δεδομένων – Έξοδος Αποτελεσμάτων
	Απόφαση
	Υποπρόγραμμα (=Συνάρτηση)
	Σύνδεσμος
	Σχόλιο
	Ροή

ANSI Flowchart symbols (Capron, H. L., Perron, J. D., p. 520)

Εισηγήσεις για Ομοιόμορφο Σχεδιασμό Λογικών Διαγραμμάτων

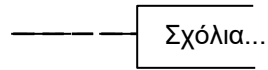
1. Το λογικό διάγραμμα πρέπει να αντιστοιχεί με το πρόγραμμα. Αντιστοιχία λογικού διαγράμματος και προγράμματος σημαίνει ότι:
 - i. οι τεχνικές και οι δομές προγραμματισμού πρέπει να είναι οι ίδιες στο λογικό διάγραμμα και στο πρόγραμμα. Για παράδειγμα, αν χρησιμοποιείται η δομή if-else στο λογικό διάγραμμα, θα πρέπει να αναμένεται η ίδια δομή και στο πρόγραμμα.
 - ii. Οι μεταβλητές πρέπει να είναι οι ίδιες στο λογικό διάγραμμα και στο πρόγραμμα.
2. Τα μηνύματα εισόδου και εξόδου στο λογικό διάγραμμα δεν είναι υποχρεωτικά. Το πρόγραμμα όμως, πρέπει να περιλαμβάνει τα κατάλληλα μηνύματα εισόδου και εξόδου (είτε το ζητά η άσκηση είτε όχι, στα πλαίσια του καλού προγραμματισμού).
3. Οι μεταβλητές που χρησιμοποιούνται σε λογικό διάγραμμα πρέπει να γράφονται με λατινικούς χαρακτήρες. Οι οδηγίες μπορούν να είναι στα Ελληνικά (διάβασε, τύπωσε, κτλ.).
4. Στις αποφάσεις στο λογικό διάγραμμα μπορούν να χρησιμοποιηθούν τα εξής: Αληθής – Ψευδής, ή A – Ψ, ή Ναι – Όχι, ή Ν – Ο, ή True – False, ή T – F, ή Yes – No, ή Y – N.
5. Ως σύμβολο εκχώρησης τιμής σε μεταβλητή συστήνεται όπως χρησιμοποιείται το βέλος (←) για ομοιομορφία, χωρίς όμως να απαγορεύονται άλλα σύμβολα όπως το =
6. Σε περίπτωση μαθηματικών πράξεων στο λογικό διάγραμμα, μπορούν να χρησιμοποιηθούν είτε μαθηματικά σύμβολα (όπως $\frac{a+b+c}{3}$, $\sqrt{a-b}$, x^2 , \leq , κτλ.), είτε σύμβολα που συνήθως χρησιμοποιούνται σε γλώσσες προγραμματισμού όπως: * για πολλαπλασιασμό, / για διαίρεση, MOD, %, DIV, <=, >=, κτλ.
7. Σε περίπτωση εκτύπωσης (πραγματικών ή και ακεραίων αριθμών, χαρακτήρων και ακολουθίας χαρακτήρων) σε λογικό διάγραμμα δεν χρειάζονται οι δείκτες πλάτους εκτύπωσης. Στο αντίστοιχο πρόγραμμα όμως, συστήνεται όπως χρησιμοποιούνται στα πλαίσια του καλού προγραμματισμού. Για παράδειγμα:

Pascal: `writeln(A:6:2, B:8, 'Onoma':8);`

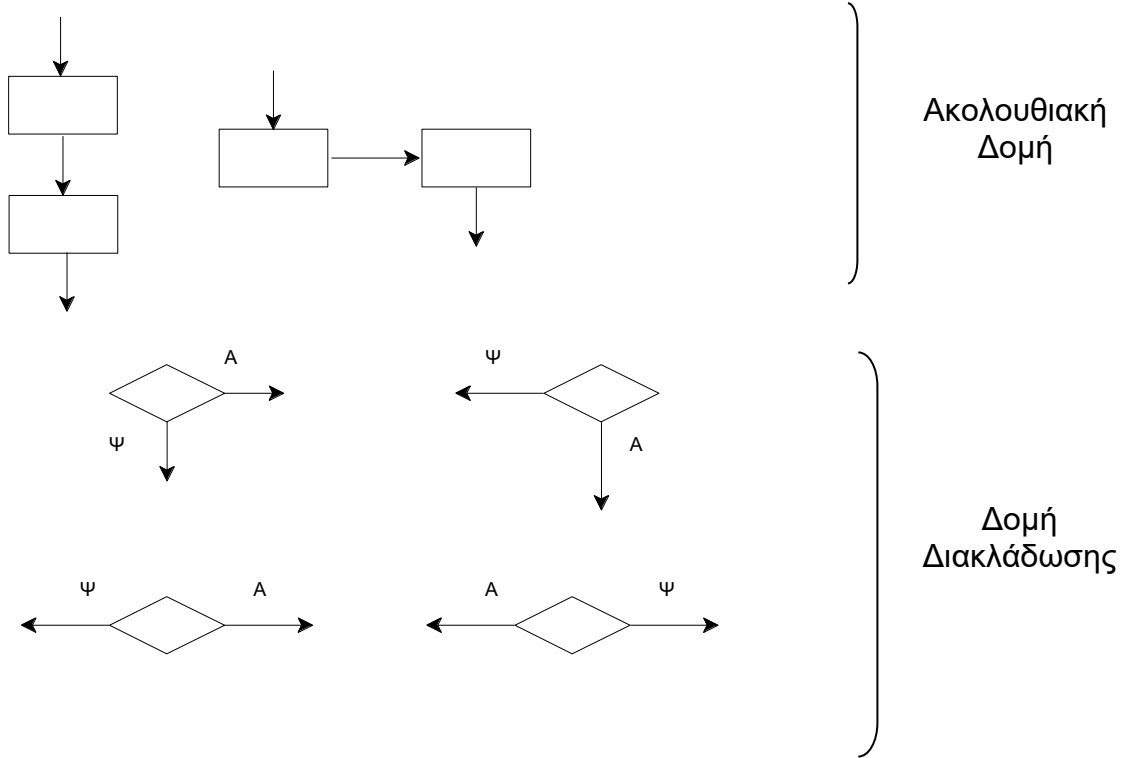
C++:

`cout<<fixed<<setw(8)<<setprecision(2)<<A<<setw(8)<<B<<setw(8)<<"Onoma";`

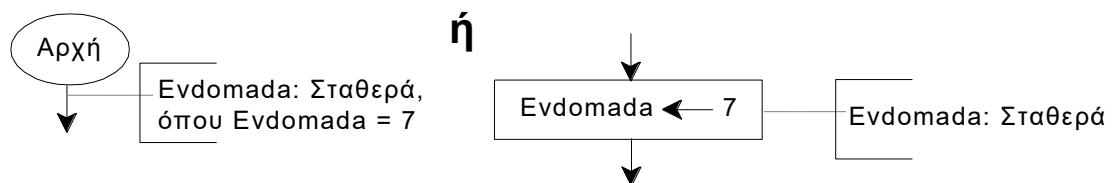
8. Στο λογικό διάγραμμα μπορούν να γραφτούν σχόλια χρησιμοποιώντας το πιο κάτω σύμβολο:



9. Η ροή στο λογικό διάγραμμα υποδεικνύεται από τα βέλη. Τα πιο κάτω σχήματα είναι αποδεκτά:

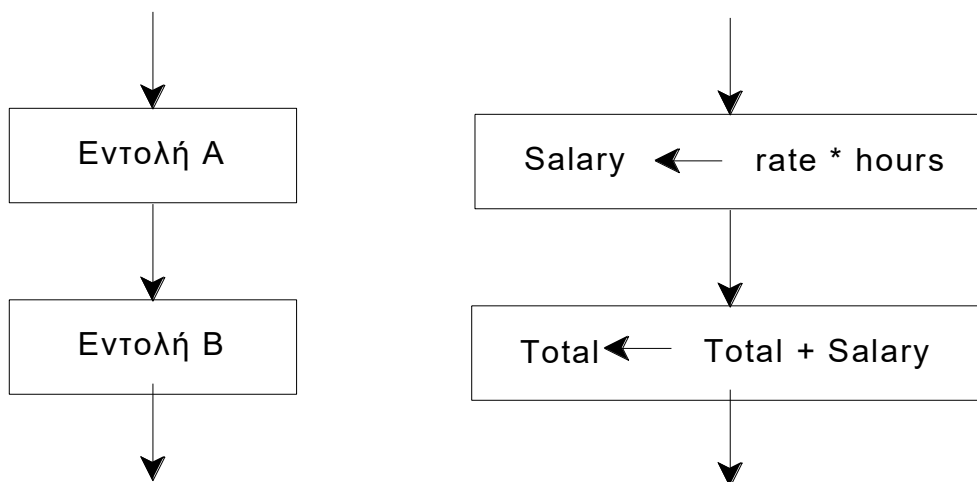


10. Οι σταθερές πρέπει να διαφοροποιούνται από τις μεταβλητές με τη χρήση σχολίου. Για παράδειγμα:



Δομές Προγραμματισμού

(α) Ακολουθιακή δομή

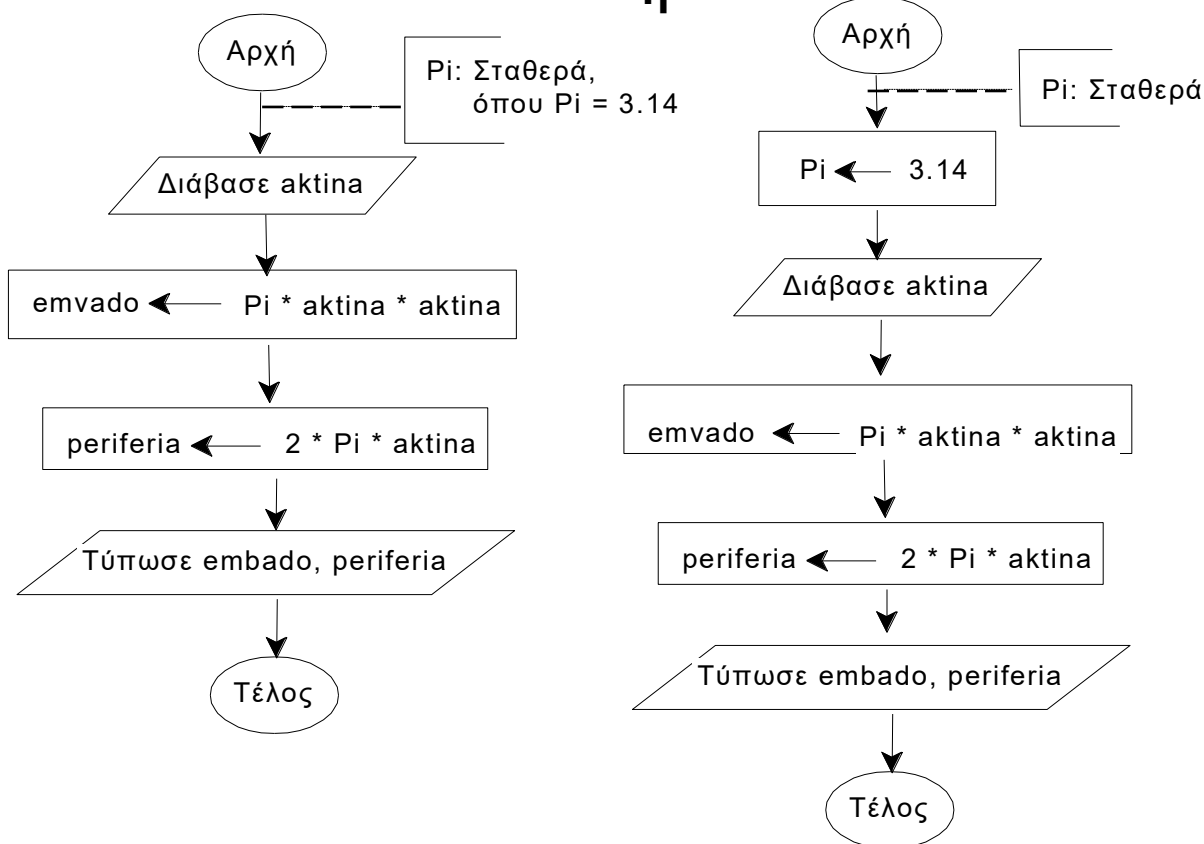


Παράδειγμα Ακολουθιακής Δομής

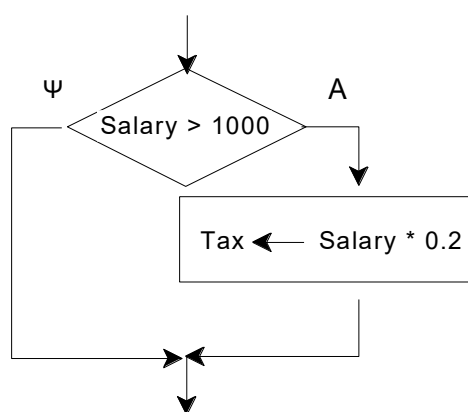
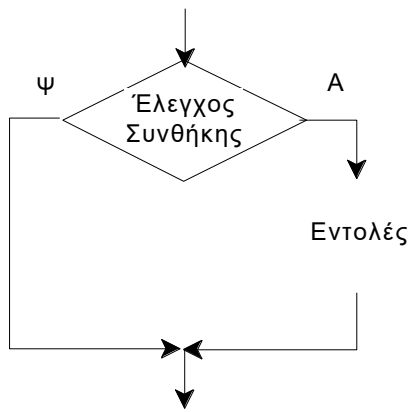
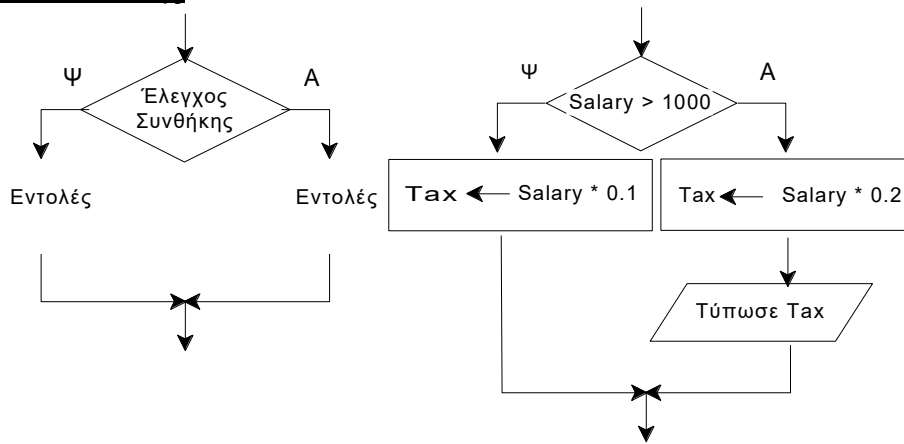
Να σχεδιάσετε λογικό διάγραμμα το οποίο να ζητά την ακτίνα του κύκλου και να υπολογίζει και παρουσιάζει το εμβαδόν και την περιφέρεια του.

(Σημείωση: Εμβαδόν Κύκλου = πr^2 , Περιφέρεια Κύκλου = $2\pi r$, $\pi = 3.14$)

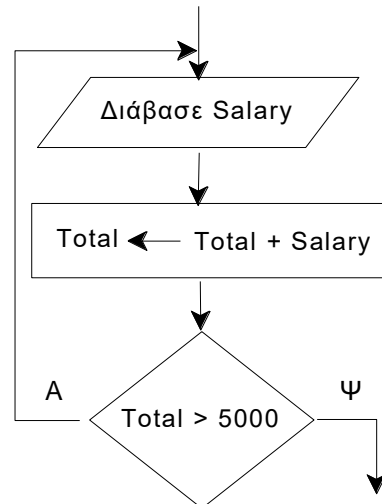
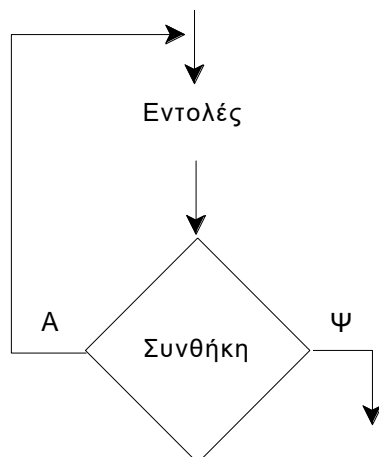
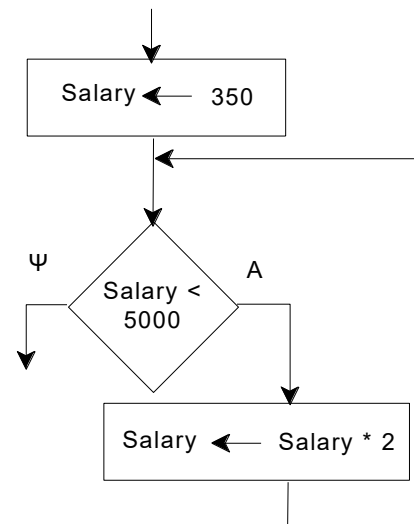
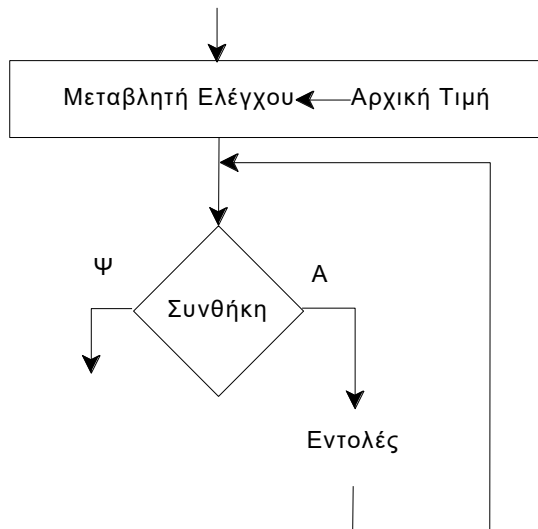
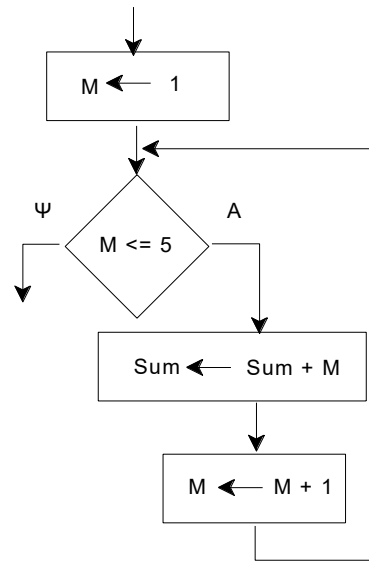
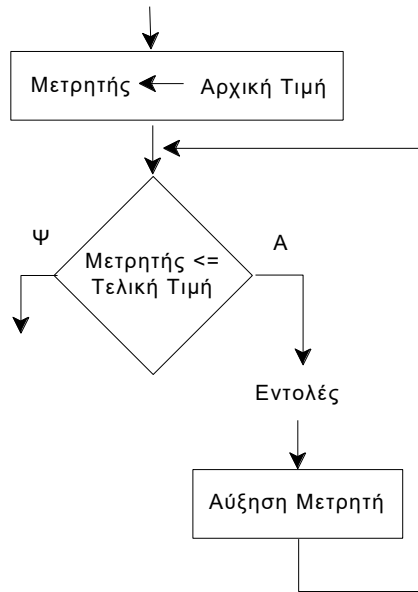
ή



(β) Δομή Διακλάδωσης

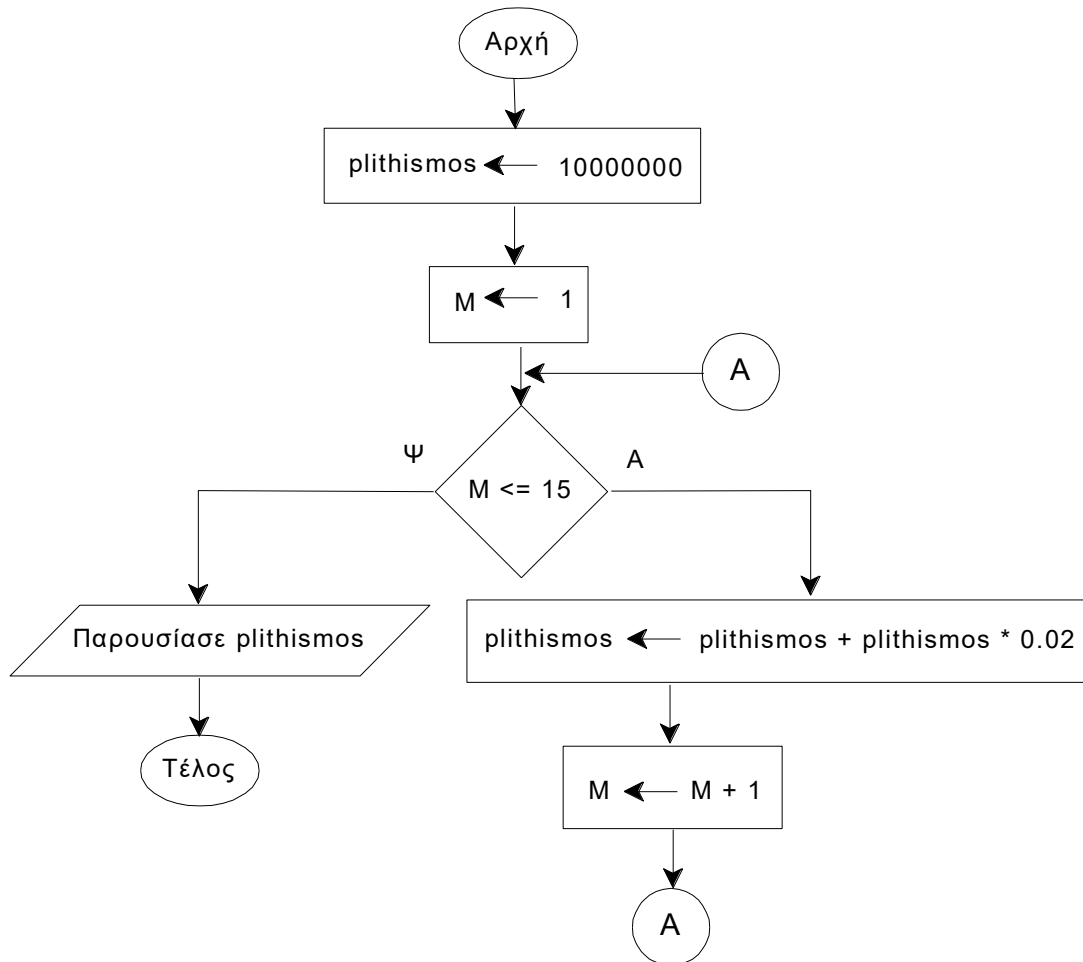


(γ) Επαναληπτική Δομή

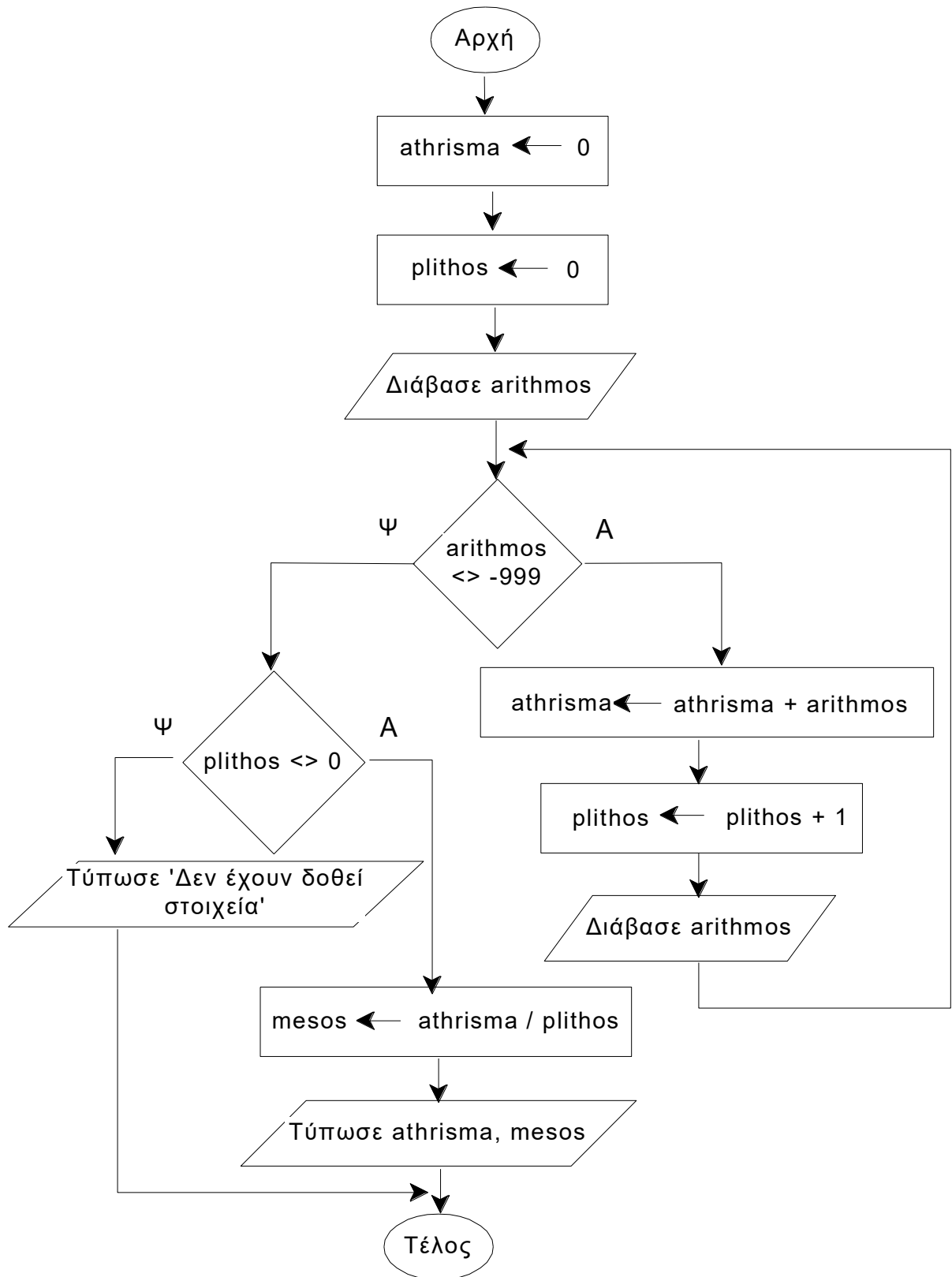


Παραδείγματα Επαναληπτικής Δομής

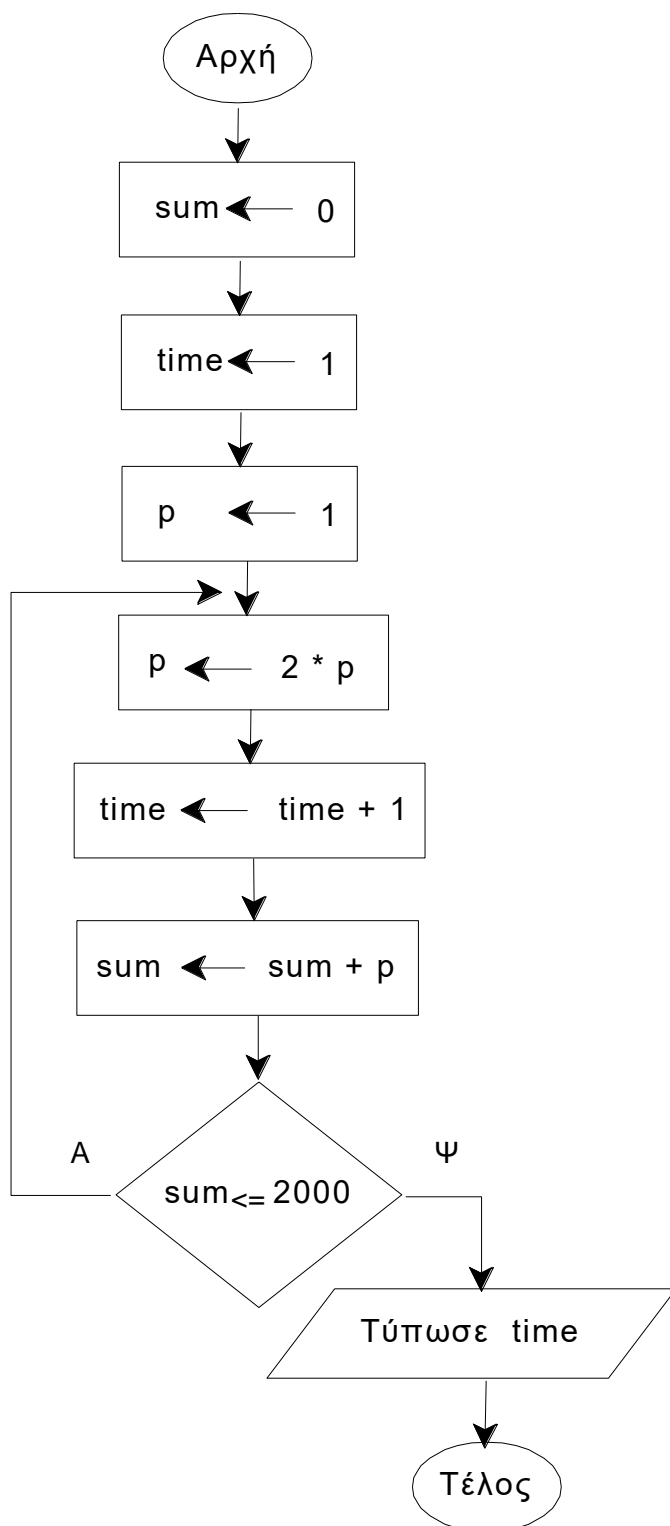
1. Ο πληθυσμός μιας χώρας είναι 10 εκατομμύρια και παρουσιάζει ετήσια αύξηση 2%. Να σχεδιάσετε λογικό διάγραμμα το οποίο να υπολογίζει και να παρουσιάζει τον πληθυσμό της χώρας μετά από 15 χρόνια.



2. Να σχεδιάσετε λογικό διάγραμμα το οποίο να δέχεται ένα πλήθος θετικών ακεραίων αριθμών και να υπολογίζει και παρουσιάζει το άθροισμα τους και τον μέσο όρο τους. Η διαδικασία να τερματίζεται με την εισαγωγή αρνητικού αριθμού.



3. Ο πατέρας του Γιάννη δίνει στο παιδί του διπλάσιο ποσό από τον προηγούμενο μήνα. Να σχεδιάσετε λογικό διάγραμμα το οποίο να υπολογίζει και παρουσιάζει το μικρότερο χρονικό διάστημα όπου ο Γιάννης θα πάρει ποσό που να υπερβαίνει τις 2000 ευρώ. Να υποθέσετε ότι ο Γιάννης πήρε την πρώτη φορά 1 ευρώ.



Προκαταρκτική Εκτέλεση

Κατά την προκαταρκτική εκτέλεση ο μαθητής εκτελεί βήμα προς βήμα όλες τις εντολές του προγράμματος ή του λογικού διαγράμματος, όπως θα τις εκτελούσε ο ηλεκτρονικός υπολογιστής. Σε κάθε βήμα σημειώνει τις τιμές των μεταβλητών ή των αποφάσεων που αλλάζουν. Για το σκοπό αυτό χρησιμοποιείται συνήθως ένας πίνακας στον οποίο καταχωρούνται:

- i. Οι τιμές των σταθερών (εάν υπάρχουν).
- ii. Οι αρχικές τιμές των μεταβλητών καθώς και οι τιμές τους μετά από κάθε αλλαγή.
- iii. Οι αποφάσεις και οι τιμές τους όταν αλλάζουν.
- iv. Οι δείκτες και το περιεχόμενο των πινάκων.
- v. Η παρουσίαση των μηνυμάτων και των αποτελεσμάτων του προγράμματος ή του λογικού διαγράμματος. Η παρουσίαση θεωρείται ανεξάρτητη από το κύριο πρόγραμμα ή τα υποπρογράμματα (συνάρτηση ή διαδικασία, αν υπάρχουν) και είναι ενιαία (δηλαδή, όπως θα εμφανιζόταν στην οθόνη του Η.Υ. μετά την εκτέλεση του προγράμματος).

Για σκοπούς εξοικονόμησης χώρου τα βήματα της προκαταρκτικής εκτέλεσης δεν είναι αναγκαίο να παρουσιάζονται οπωσδήποτε σε ξεχωριστές γραμμές (βλέπε παραδείγματα).

Δομές Προγραμματισμού

(α) Ακολουθιακή Δομή

Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου: 5.5, 8.0

```
// Τρίγωνο
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

int main() {
    float katheti1, katheti2, ypoteinousa, perimetros, emvadon;
    cout << "Δώσε το μήκος της κάθετης 1: ";
    cin >> katheti1;
    cout << "Δώσε το μήκος της κάθετης 2: ";
    cin >> katheti2;
    ypoteinousa = sqrt( pow( katheti1, 2.0 ) + pow( katheti2, 2.0 ) );
    perimetros = katheti1 + katheti2 + ypoteinousa ;
    emvadon = katheti1 * katheti2 / 2;
    cout << "Κάθετη 1: " << fixed << setprecision(2) << katheti1 << endl;
    cout << " Κάθετη 2: " << fixed << setprecision(2) << katheti2 << endl;
    cout << "Υποτείνουσα: " << fixed << setprecision(2) << ypoteinousa << endl;
    cout << "Περίμετρος: " << fixed << setprecision(2) << perimetros << endl;
    cout << "Εμβαδόν: " << fixed << setprecision(2) << emvadon << endl;
    return 0;
}
```

Μεταβλητές					Παρουσίαση
katheti1	katheti2	ypoteinousa	perimetros	emvadon	
5.5					Δώσε το μήκος της κάθετης 1:
	8.0				Δώσε το μήκος της κάθετης 2:
		9.71	23.21	22.00	Κάθετη 1: 5.50
					Κάθετη 2: 8.00
					Υποτείνουσα: 9.71
					Περίμετρος: 23.21
					Εμβαδόν: 22.00

(β) Δομή Διακλάδωσης

Παράδειγμα 1: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου: 2000, 100

```
// Rent_A_Car
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    int kivismos;
    float timi, poso, ekptosi;
    ekptosi = 0.0;
    cout << "Δώσε τον κυβισμό: ";
    cin >> kivismos;
    cout << "Δώσε την αρχική τιμή: ";
    cin >> timi;

    if ( kivismos >= 1500 )
        ekptosi = timi * 10/100;

    poso = timi - ekptosi;
    cout << "Ο κυβισμός είναι: " << kivismos << endl;
    cout << "Η αρχική τιμή είναι: " << fixed << setprecision(2) << timi << endl;
    cout << "Η έκπτωση είναι: " << fixed << setprecision(2) << ekptosi << endl;
    cout << "Η τιμή πληρωμής είναι: " << fixed << setprecision(2) << poso <<
endl;
    return 0;
}
```

Μεταβλητές				Απόφαση		Παρουσίαση
ekptosi	kivismos	timi	poso	kivismos>=1500	A/Ψ	
0.0						Δώσε τον κυβισμό: Δώσε την αρχική τιμή: Ο κυβισμός είναι: 2000 Η αρχική τιμή είναι: 100.00 Η έκπτωση είναι: 10.00 Η τιμή πληρωμής είναι: 90.00
	2000	100		2000>=1500	A	
10.0			90.0			

Παράδειγμα 2: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου για την περίπτωση 1: M, 45 και για την περίπτωση 2: F, 28 .

```
// Insurance
#include <iostream>
using namespace std;
int main() {
    int age, asfalistra;
    char sex;
    cout << "Δώσε το φύλο: ";
    cin >> sex;
    cout << "Δώσε την ηλικία: ";
    cin >> age;
    if ( sex == 'M' )
        if ( age < 25 )
            asfalistra = 300;
        else
            asfalistra = 250;
    else if ( age < 21 )
        asfalistra = 200;
    else
        asfalistra = 150;
    cout << "Ασφάλιστρα: " << asfalistra;
    return 0;
}
```

Περίπτωση 1

Μεταβλητές			Απόφαση				Παρουσίαση
age	asfalistra	sex	sex=='M'	A/Ψ	age<25	A/Ψ	Δώσε το φύλο: Δώσε την ηλικία: Ασφάλιστρα: 250
		M					
45							
	250		M==M	A	45<25	Ψ	

Περίπτωση 2

Μεταβλητές			Απόφαση				Παρουσίαση
age	asfalistra	sex	sex=='M'	A/Ψ	age<21	A/Ψ	Δώσε το φύλο: Δώσε την ηλικία: Ασφάλιστρα: 150
		F					
28							
	150		F==M	Ψ	28<21	Ψ	

Περιπτωσιακή Δομή

Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου :
Maria, 4 .

```
// Eniaio;
#include <iostream>
#include <string>
using namespace std;
int main() {
    string name;
    int code;
    cout << "Δώσε το όνομα: ";
    cin >> name;
    cout << "Δώσε τον κωδικό: ";
    cin >> code;
    switch ( code ) {
        case 1: cout << "ΒΙΟΛΟΓΙΑ"; break;
        case 2: cout << "ΜΑΘΗΜΑΤΙΚΑ"; break;
        case 3: cout << "ΕΛΛΗΝΙΚΑ"; break;
        case 4: cout << "ΠΛΗΡΟΦΟΡΙΚΗ"; break;
        case 5: cout << "ΙΣΤΟΡΙΑ"; break;
        case 6: cout << "ΤΕΧΝΗ"; break;
        default: cout << "Μη επιτρεπτός κωδικός αριθμός. "; break;
    }
    return 0;
}
```

Μεταβλητές		Περίπτωση code (Α/Ψ)							Παρουσίαση
name	code	1	2	3	4	5	6	Αλλιώς	
Maria									Δώσε το όνομα: Δώσε τον κωδικό: ΠΛΗΡΟΦΟΡΙΚΗ
	4				A				

(γ) Επαναληπτική Δομή

Παράδειγμα 1: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος.

```
// Average
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    int no, sum, metritis;
    float mesos;
    sum = 0;
    metritis = 0;
    for ( no=10; no<=15; no++ ) {
        sum = sum + no;
        metritis = metritis + 1;
    }
    mesos = (float) sum / metritis;
    cout << "Ο μέσος όρος είναι: " << fixed << setprecision(2) << mesos;
    return 0;
}
```

Μεταβλητές				Απόφαση		Παρουσίαση
no	sum	metritis	mesos	no<=15	A/Ψ	Ο μέσος όρος είναι: 12.50
10	0	0		10<=15	A	
	10	1				
11	21	2		11<=15	A	
12	33	3		12<=15	A	
13	46	4		13<=15	A	
14	60	5		14<=15	A	
15	75	6		15<=15	A	
16				16<=15	Ψ	
			12.50			

Παράδειγμα 2: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος.

```
// Sum_Of_Odd_Nos;
#include <iostream>
using namespace std;
int main() {
    int arithmos, sum;
    sum = 0;
    arithmos = 1;
    while ( arithmos <= 11 ) {
        sum = sum + arithmos;
        arithmos = arithmos + 2;
    }
    cout << "Το άθροισμα των μονών αριθμών (1-11) είναι: " << sum;
    return 0;
}
```

Μεταβλητές		Απόφαση		Παρουσίαση
arithmos	sum	arithmos<=11	A/Ψ	Το άθροισμα των μονών αριθμών (1-11) είναι: 36
1	0	1<=11	A	
	1			
3	4	3<=11	A	
5	9	5<=11	A	
7	16	7<=11	A	
9	25	9<=11	A	
11	36	11<=11	A	
13		13<=11	Ψ	

Παράδειγμα 3: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος.

```
// Sum_Of_Odd_Nos;
#include <iostream>
using namespace std;
int main() {
    int arithmos, sum;
    sum = 0;
    arithmos = 1;
    do {
        sum = sum + arithmos;
        arithmos = arithmos + 2;
    } while ( arithmos <= 11);
    cout << "Το άθροισμα των μονών αριθμών (1-11) είναι: " << sum;
    return 0;
}
```

Μεταβλητές		Απόφαση		Παρουσίαση
arithmos	sum	arithmos<=11	A/Ψ	Το άθροισμα των μονών αριθμών (1-11) είναι: 36
1	0			
3	1	3<=11	A	
5	4	5<=11	A	
7	9	7<=11	A	
9	16	9<=11	A	
11	25	11<=11	A	
13	36	13<=11	Ψ	

Υποπρογράμματα - Συναρτήσεις που υπολογίζουν και επιστρέφουν μόνο μια τιμή στο κύριως πρόγραμμα (με τη χρήση της εντολής return)

Παράδειγμα 1: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου: 10.5, 12.5 .

```
// Demo1;
#include <iostream>
#include <iomanip>
using namespace std;

float calculate ( float m, float p) { // συνάρτηση calculate
    float emvadon, perimetros;
    emvadon = p * m;
    perimetros = 2.0 * (m + p);
    cout << "Το εμβαδόν είναι: " << fixed << setprecision(2) << emvadon << endl;
    return perimetros;
}

int main() { // κύριο πρόγραμμα
    float mikos, platos, perim;
    cout << "Δώσε το μήκος: ";
    cin >> mikos;
    cout << "Δώσε το πλάτος: ";
    cin >> platos;
    perim = calculate ( mikos, platos );
    cout << "Η περίμετρος είναι: " << fixed << setprecision(2) << perim << endl;
    return 0;
}
```

Κύρια συνάρτηση (main)

Μεταβλητές			Παρουσίαση
mikos	platos	perim	
10.5	12.5	46	Δώσε το μήκος: Δώσε το πλάτος: Το εμβαδόν είναι: 131.25 Η περίμετρος είναι: 46.00

Συνάρτηση calculate

Τυπικές παράμετροι		Τοπικές μεταβλητές		Επιστρέφει
m	p	emvadon	perimetros	
10.5	12.5	131.25	46	46

Παράδειγμα 2: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου: 5, 10 .

```
// Μεγαλύτερος
#include <iostream>
using namespace std;

int bigger( int num1, int num2 ) {
    if ( num1 > num2 )
        return num1;
    else
        return num2;
}

int main() {
    int arithmos1, arithmos2;
    cout << "Δώσε δύο ακέραιους αριθμούς: ";
    cin >> arithmos1 >> arithmos2;
    cout << "Ο μεγαλύτερος είναι: " << bigger( arithmos1, arithmos2) << endl;
    return 0;
}
```

Κύρια συνάρτηση (main)

Μεταβλητές		Παρουσίαση
arithmos1	arithmos2	
5	10	Δώσε δύο ακέραιους αριθμούς: Ο μεγαλύτερος είναι: 10

Συνάρτηση bigger

Τυπικές παράμετροι		Απόφαση		Επιστρέφει
num1	num2	num1>num2	Α/Ψ	
5	10	5>10	Ψ	10

Υποπρογράμματα – Συναρτήσεις που έχουν τυπικές παράμετρος αναφοράς και δεν έχουν την εντολή return

Παράδειγμα 1: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου: 10, 12, 3.

```
// Vol_Per;
#include <iostream>
using namespace std;

void perivol( int length, int width, int height, int &perimeter, int &volume ) {
    perimeter = length + 2 * width * height;
    volume = length * width * height;
}

int main() {
    int len, wid, heit, perim, vol;
    cout << "Δώσε το μήκος: ";
    cin >> len;
    cout << "Δώσε το πλάτος: ";
    cin >> wid;
    cout << "Δώσε το ύψος: ";
    cin >> heit;
    perivol ( len, wid, heit, perim, vol);
    cout << "Η περίμετρος είναι: " << perim << endl ;
    cout << "Ο όγκος είναι: " << vol << endl;
    return 0;
}
```

Κύρια συνάρτηση (main)

Μεταβλητές					Παρουσίαση
len	wid	heit	perim	vol	Δώσε το μήκος:
10	12	3	82	360	Δώσε το πλάτος:
					Δώσε το ύψος:
					Η περίμετρος είναι: 82
					Ο όγκος είναι: 360

Συνάρτηση perivol

Τυπικές Παράμετροι Τιμών			Τυπικές Παράμετροι Αναφοράς	
length	width	height	perimeter	volume
10	12	3	82	360

Υποπρογράμματα – Συναρτήσεις που έχουν και τυπικές παράμετρους αναφοράς και την εντολή return.

Παράδειγμα 4.6 - σελ. βιβλίου 135
(Έχει τυπικές παράμετρους αναφοράς ΚΑΙ την εντολή return).

```
#include <iostream>
using namespace std;

int check(int a, int &b){
    b += 2;
    if (a>b)
        return a;
    return a+b;
}

int main(){
    int x = 3, y = 2;
    if (x>y)
        cout << check(x,y) << endl;
    else
        cout << check(y,x) << endl;
    cout << x << " " << y << endl;
    return 0;
}
```

Κύρια συνάρτηση (main)

Μεταβλητές		Αποφάσεις		Παρουσίαση
x	y	x>y	T/F	
3	2	3>2	T	
	4			7
				3□4

Συνάρτηση check

Τυπικές Παράμετροι Τιμών	Τυπικές Παράμετροι Αναφοράς	Αποφάσεις		Επιστρέφει
a	b	a>b	T/F	
3	2			
	4	3>4	F	7

Υποπρογράμματα – Συναρτήσεις που δεν έχουν ούτε τυπικές παράμετρος αναφοράς ούτε και την εντολή return αλλά το αποτέλεσμα τυπώνεται εντός της συνάρτησης.)

Παράδειγμα 1:

```
#include <iostream>
using namespace std;

void check(int a, int b){
    if (a>b)
        cout<<"a="<<a<<endl;
    else
        cout<<"b="<<b<<endl;
}

int main(){
    int x = 3, y = 2;
    if (x>y)
        check(x,y);
    else
        check(y,x);
    cout<<"x="<<x<<" "<<"y="<<y;
    return 0;
}
```

Κύρια συνάρτηση (main)

Μεταβλητές		Αποφάσεις		Παρουσίαση
x	y	x>y	T/F	
3	2	3>2	T	a=3
				x=3 y=2

Συνάρτηση check

Παράμετροι		Αποφάσεις	
a	b	a>b	T/F
3	2	3>2	T

Παράδειγμα 2: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου: 10, 12, 3.

```
// Vol_Per;
#include <iostream>
using namespace std;

void perivol( int length, int width, int height ) {
    int perimeter, volume;
    perimeter = length + 2 * width * height;
    volume = length * width * height;
    cout << "Η περίμετρος είναι: " << perimeter << endl;
    cout << "Ο όγκος είναι: " << volume << endl;
}

int main() {
    int len, wid, heit;
    cout << "Δώσε το μήκος: ";
    cin >> len;
    cout << "Δώσε το πλάτος: ";
    cin >> wid;
    cout << "Δώσε το ύψος: ";
    cin >> heit;
    perivol( len, wid, heit);
    return 0;
}
```

Κύρια συνάρτηση (main)

Μεταβλητές			Παρουσίαση
len	wid	heit	
10	12	3	Δώσε το μήκος:
			Δώσε το πλάτος:
			Δώσε το ύψος:
			Η περίμετρος είναι: 82
			Ο όγκος είναι: 360

Συνάρτηση perivol

Τυπικές Παράμετροι Τιμών			Τοπικές μεταβλητές	
length	width	height	perimeter	volume
10	12	3	82	360

Μονοδιάστατοι Πίνακες

Παράδειγμα 1: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου: 10, 20, 30, 40, 50 .

```
// Πίνακες
#include <iostream>
using namespace std;

const int LOW=0, HIGH=5;

int main() {
    int i;
    int lista[ HIGH ];
    for ( i=LOW; i<HIGH; i++ ) {
        cout << "Δώσε τον αριθμό: ";
        cin >> lista[ i ];
    }
    cout << "Αντίστροφα οι αριθμοί είναι: " << endl;
    for ( i = HIGH-1; i>=LOW; i-- )
        cout << lista[ i ] << endl;
    return 0;
}
```

Σταθερές		Μεταβλητές			Απόφαση			Παρουσίαση
LOW	HIGH	i	Πίνακας lista		i<5	i>=0	A/Ψ	
			Δείκτης	Περιεχόμενο				
0	5	0	0	10	0<5		A	Δώσε τον αριθμό:
		1	1	20	1<5		A	Δώσε τον αριθμό:
		2	2	30	2<5		A	Δώσε τον αριθμό:
		3	3	40	3<5		A	Δώσε τον αριθμό:
		4	4	50	4<5		A	Δώσε τον αριθμό:
		5			5<5		Ψ	Αντίστροφα οι αριθμοί
		4	4			4>=0	A	είναι:
		3	3			3>=0	A	50
		2	2			2>=0	A	40
		1	1			1>=0	A	30
		0	0			0>=0	A	20
		-1				-1>=0	Ψ	10

Παράδειγμα 2: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος.

```
// Πίνακες2
#include <iostream>
using namespace std;

int main() {
    int i, a[ 6 ];
    for ( i = 0; i< 5; i++)
        a[ i ] = (i+1) * 3;
    for ( i = 0; i< 4; i++)
        a[ i+1 ] = (i+1) * a[ i ];
    for ( i = 0; i< 5; i++ )
        cout << a[ i ] << endl;
    return 0;
}
```

Μεταβλητές			Απόφαση				Παρουσίαση
i	Πίνακας a		i < 5	i < 4	i < 5	A/Ψ	
	Δείκτης	Περιεχόμενο					
0	0	3	0 < 5			A	
1	1	6	1 < 5			A	3
2	2	9	2 < 5			A	3
3	3	12	3 < 5			A	6
4	4	15	4 < 5			A	18
5			5 < 5			Ψ	72
0	1	3		0 < 4		A	
1	2	6		1 < 4		A	
2	3	18		2 < 4		A	
3	4	72		3 < 4		A	
4				4 < 4		Ψ	
0	0				0 < 5	A	
1	1				1 < 5	A	
2	2				2 < 5	A	
3	3				3 < 5	A	
4	4				4 < 5	A	
5					5 < 5	Ψ	

Πίνακες δύο Διαστάσεων

Παράδειγμα 1: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου:3, 2, -1, 0, 6, 8, 11, 2, 10.

```
// Pinakes_Dis;
#include <iostream>
using namespace std;
const int MAX=3;

int main() {
    int i, j, sum, a[MAX][MAX];
    cout << "Δώσε τις τιμές του πίνακα: " << endl;
    for ( i = 0; i < MAX; i++ )
        for ( j = 0; j < MAX; j++ )
            cin >> a[ i ][ j ];
    sum = 0;
    for ( i = 0; i < MAX; i++ )
        for ( j = 0; j < MAX; j++ )
            if ( i == j )
                sum = sum + a[ i ][ j ];
    cout << "Το άθροισμα της κύριας διαγωνίου είναι: " << sum;
    return 0;
}
```

Σταθερά	Μεταβλητές					Απόφαση						Παρουσίαση	
	MAX	sum	i	j	Πίνακας a		i<MAX	A/Ψ	j<MAX	A/Ψ	i == j		A/Ψ
					Δείκτες	Περιεχόμενο							
3		0	0	0,0	3	0<3	A	0<3	A				<p>Δώσε τις τιμές του πίνακα:</p> <p>Το άθροισμα της κύριας διαγωνίου είναι: 19</p>
			1	0,1	2			1<3	A				
			2	0,2	-1			2<3	A				
			3					3<3	Ψ				
		1	0	1,0	0	1<3	A	0<3	A				
			1	1,1	6			1<3	A				
			2	1,2	8			2<3	A				
			3					3<3	Ψ				
		2	0	2,0	11	2<3	A	0<3	A				
			1	2,1	2			1<3	A				
			2	2,2	10			2<3	A				
			3					3<3	Ψ				
		3				3<3	Ψ						
	0												
	3	0	0	0,0		0<3	A	0<3	A	0==0	A		
			1	0,1				1<3	A	0==1	Ψ		
			2	0,2				2<3	A	0==2	Ψ		
			3					3<3	Ψ				
		1	0	1,0		1<3	A	0<3	A	1==0	Ψ		
	9		1	1,1				1<3	A	1==1	A		
			2	1,2				2<3	A	1==2	Ψ		
			3					3<3	Ψ				
		2	0	2,0		2<3	A	0<3	A	2==0	Ψ		
			1	2,1				1<3	A	2==1	Ψ		
	19		2	2,2				2<3	A	2==2	A		
			3					3<3	Ψ				
		3				3<3	Ψ						

Παράδειγμα 2: Να γίνει η προκαταρκτική εκτέλεση του πιο κάτω προγράμματος με τιμές εισόδου:3, 2, -1, 0, 6, 8, 11, 2, 10.

```
// Pinakes_Dis_1;
#include <iostream>
using namespace std;
const int MAX = 3;
int i, j;
void eisodos( int p[ ][ MAX ] ) {
    cout << "Δώσε τις τιμές του πίνακα: " << endl;
    for ( i = 0; i<MAX; i++ )
        for ( j = 0; j<MAX; j++ )
            cin >> p[ i ][ j ];
}
int calculate( int p[ ][ MAX ] ) {
    int total;
    total = 0;
    for ( i = 0; i<MAX; i++ )
        for ( j = 0; j<MAX; j++ )
            if ( i == j )
                total = total + p[ i ][ j ];
    return total;
}
int main() {
    int a[ MAX ][ MAX ] , sum;
    eisodos( a );
    sum = calculate( a );
    cout << "Το άθροισμα της κύριας διαγωνίου είναι: " << sum;
    return 0;
}
```


Σταθερές MAX	Μεταβλητές		Παρουσίαση	
	sum	Πίνακας a		
		Δείκτες		Περιεχόμενο
3		0,0	3	Δώσε τις τιμές του πίνακα: Το άθροισμα της κύριας διαγωνίου είναι: 19
		0,1	2	
		0,2	-1	
		1,0	0	
		1,1	6	
		1,2	8	
		2,0	11	
		2,1	2	
		2,2	10	
	19			

Συνάρτηση eisodos

Καθολικές Μεταβλητές		Τυπικές παράμετροι αναφοράς		Απόφαση			
i	j	Πίνακας p		i<MAX	A/Ψ	j<MAX	A/Ψ
		Δείκτες	Περιεχόμενο				
0	0	0,0	3	0<3	A	0<3	A
	1	0,1	2			1<3	A
	2	0,2	-1			2<3	A
	3					3<3	Ψ
1	0	1,0	0	1<3	A	0<3	A
	1	1,1	6			1<3	A
	2	1,2	8			2<3	A
	3					3<3	Ψ
2	0	2,0	11	2<3	A	0<3	A
	1	2,1	2			1<3	A
	2	2,2	10			2<3	A
	3					3<3	Ψ
3				3<3	Ψ		

Συνάρτηση calculate

Μεταβλητές			Τυπικές παράμετροι		Απόφαση					
total	Τοπικές	Καθολικές	Πίνακας p		i<MAX	A/Ψ	j<MAX	A/Ψ	i==j	A/Ψ
	i	j	Δείκτες	Περιεχόμενο						
	0									
3	0	0	0,0	3	0<3	A	0<3	A	0==0	A
		1	0,1	2			1<3	A	0==1	Ψ
		2	0,2	-1			2<3	A	0==2	Ψ
		3					3<3	Ψ		
	1	0	1,0	0	1<3	A	0<3	A	1==0	Ψ
9		1	1,1	6			1<3	A	1==1	A
		2	1,2	8			2<3	A	1==2	Ψ
		3					3<3	Ψ		
	2	0	2,0	11	2<3	A	0<3	A	2==0	Ψ
		1	2,1	2			1<3	A	2==1	Ψ
19		2	2,2	10			2<3	A	2==2	A
		3					3<3	Ψ		
	3				3<3	Ψ				

Σημείωση: Οι καθολικές μεταβλητές μπορούν να παρουσιάζονται και στον πίνακα προκαταρκτικής εκτέλεσης του κυρίως προγράμματος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. “Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον”. Γ΄ Λυκείου Τεχνολογική Κατεύθυνση – Βιβλίο καθηγητή
Αντωνάκος Νίκος, Βογιατζής Ιωάννης, Κατοπόδης Ιωάννης, Πατριαρχέας Κυριάκος. Υπουργείο Παιδείας και Θρησκευμάτων – Παιδαγωγικό Ινστιτούτο. Αθήνα 1999.
2. “Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον”. Γ΄ Λυκείου Τεχνολογική Κατεύθυνση – Τετράδιο μαθητή.
Αντωνάκος Νίκος, Βογιατζής Ιωάννης, Κατοπόδης Ιωάννης, Πατριαρχέας Κυριάκος. Υπουργείο Παιδείας και Θρησκευμάτων – Παιδαγωγικό Ινστιτούτο. Αθήνα 1999.
3. “Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον”. Γ΄ Λυκείου Τεχνολογική Κατεύθυνση – Βιβλίο μαθητή.
Αντωνάκος Νίκος, Βογιατζής Ιωάννης, Κατοπόδης Ιωάννης, Πατριαρχέας Κυριάκος. Υπουργείο Παιδείας και Θρησκευμάτων – Παιδαγωγικό Ινστιτούτο. Αθήνα 1999.
4. “Προγραμματισμός υπολογιστών με Visual Basic”. Βιβλίο μαθητή. Τομέας ηλεκτρονικών 2^{ος} κύκλος.
Βουτυράς Γεώργιος, Βιδιαδάκης Ανδρέας, Ματζάκος Πέτρος, Σκουρλάς Χρήστος. Οργανισμός Εκδόσεων Διδακτικών Βιβλίων, Αθήνα.
5. “Computer programming in the Pascal Language”. By Neal Golden, Antonio M. Lopez, Jr. Harcourt Brace Jovanovich, Publishers.
6. “Pascal – The software fundamentals of computer science”.
By Richard A, Meyers. Prentice-Hall International Editions.
7. “Software Engineering. A Practitioner’s Approach” .Third Edition. By Roger S. Pressman. McGraw Hill International Edition – 1992.
8. “Computing: An Introduction to Procedures and Procedure – Followers”. By Fred M. Tonge and Julian Feldman. McGraw-Hill Book Company.
9. “Computers and Information Systems”. Third edition. By H. L Capron and John D. Perron. The Benjamin / Cummings Publishing Company, Inc.
10. “FORTRAN a structured, disciplined style” . By Gordon B. Davis, Thomas R. Hoffmann. McGraw – Hill Company.
11. “Computer programming in the Basic Language”. Third Edition.
By Neal Golden. Harcourt Brace Jovanovich, Publishers.

12. "Computer programming in the Basic Language". Teachers manual and resource guide. Third Edition. By Neal Golden. Harcourt Brace Jovanovich, Publishers.
13. "Basic for students using the IBM PC". Second edition By Michael Trombetta. Addison – Wesley Publishing Company.
14. "Microsoft Basic using Modular Structure". Third Edition. By: Julia Case Bradley. WCB –1991.
15. "Structured VAX Basic and Basic-Plus". Second Edition. By: Teglovic Jr. and Kenneth D. Douglas. IRWIN-1987.
16. "Understanding Computer Science for Advanced Level". Fourth Edition. By: R. Bradley. Stanley Thornes Publishers Ltd 1999.