



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Αγρονόμων Τοπογράφων Μηχανικών

Προγραμματιστικές Τεχνικές

Βασίλειος Βεσκούκης

Δρ. Ηλεκτρολόγος Μηχανικός &
Μηχανικός Υπολογιστών ΕΜΠ
v.vescoukis@cs.ntua.gr

Ρωμύλος Κορακίτης

Αστροφυσικός
Αναπλ. Καθηγητής ΕΜΠ
romylos@survey.ntua.gr

Ανακεφαλαίωση

Εμβέλεια μεταβλητών



ΓΡΑΠΤΗ ΕΞΕΤΑΣΗ ΦΕΒΡΟΥΑΡΙΟΥ 2006

Όνοματεπώνυμο:

Εξάμηνο:

Αρ. Μητρώου:

Αίθουσα:

Θέση:

1.1	
1.2	
1.3	
1.4	
1.5	
2	
3.1	
3.2	
4	
5	

Θέμα 1 (μονάδες 25)

Δίνεται το παρακάτω τμήμα προγράμματος C++

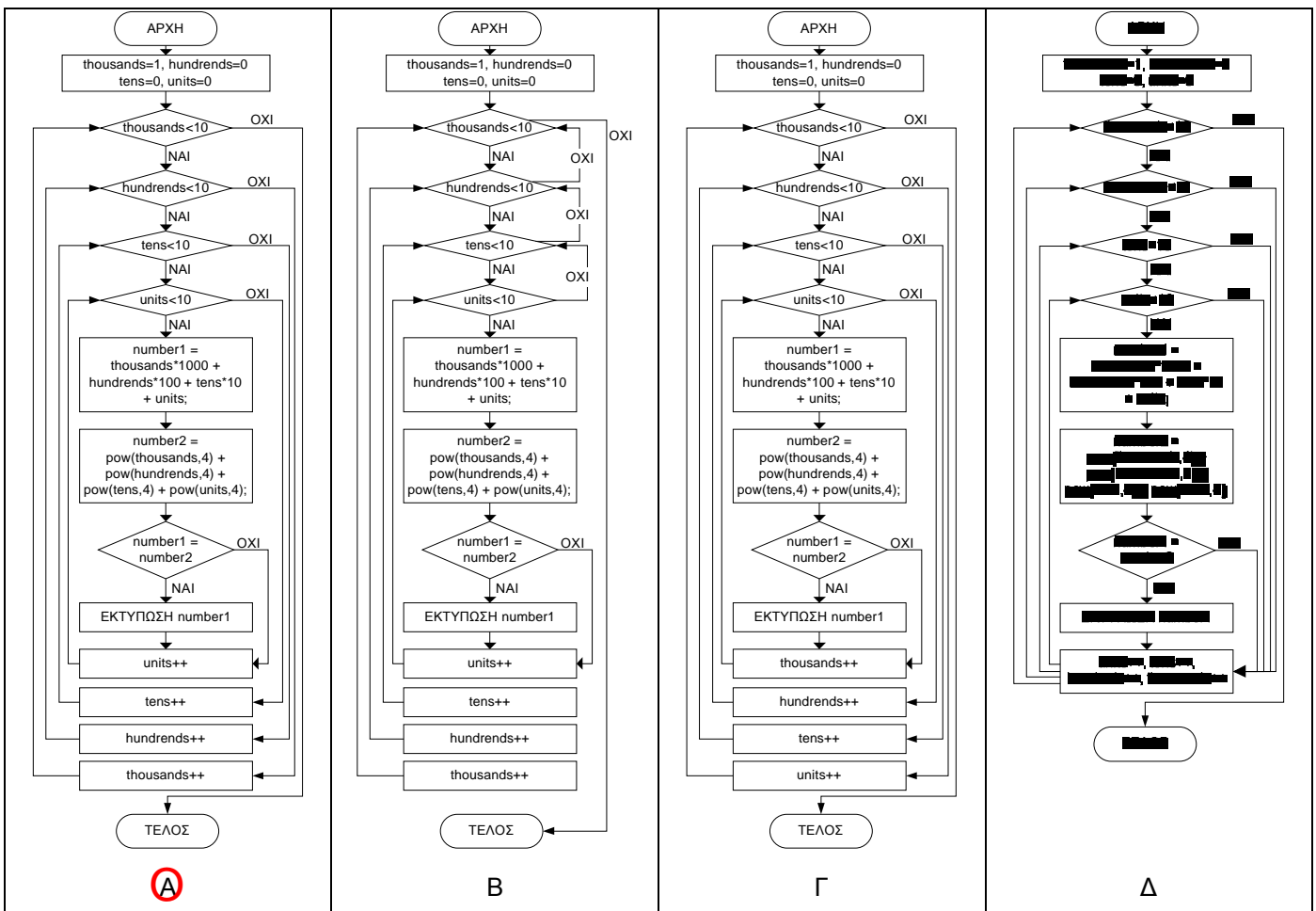
```
int thousands, hundreds, tens, units, rep=0;
float number1, number2;

for (thousands=1;thousands<=9;thousands++)
  for (hundreds=0;hundreds<=9; hundreds++)
    for (tens=0;tens<=9;tens++)
      for (units=0;units<=9;units++)
        { rep++;
          number1=thousands*1000+hundreds*100+tens*10+units;
          number2=pow(thousands,4)+pow(hundreds,4)+pow(tens,4)+pow(units,4);
          if (number1==number2)
            cout<<number1<<endl;
        }
}
```

Υπενθυμίζεται ότι η συνάρτηση $\text{pow}(x, y)$ υπολογίζει την τιμή x^y .

1 (μονάδες 7)

Υποδείξτε σε ποιον αλγόριθμο αντιστοιχεί το πρόγραμμα αυτό.



2 (μονάδες 3)

Πόσες επαναλήψεις θα συμβούν κατά την εκτέλεσή του; Εναλλακτικά μπορείτε να υποδείξετε μια προσθήκη στο πρόγραμμα η οποία να χρησιμοποιείται στον υπολογισμό των επαναλήψεων.

- 9000 επαναλήψεις
- προσθήκη:

3 (μονάδες 4)

Τι τιμές παίρνει η μεταβλητή `number1` κατά την εκτέλεση του προγράμματος; Υποδείξτε τα όρια των τιμών αυτών.

Ακέραιες τιμές από 1000 ως 9999

4 (μονάδες 5)

Προτείνετε έναν τρόπο ώστε η μεταβλητή `number1` να παίρνει τις ίδιες τιμές, γράφοντας ένα πολύ απλούστερο πρόγραμμα (αγνοείστε τη μεταβλητή `number2` και τη σύγκριση που ακολουθεί).

```
for (number1=1000; number1<10000; number1++);
```

5 (μονάδες 6)

Τι υπολογίζει και εκτυπώνει το πρόγραμμα;

Τους τετραψήφιους ακέραιους αριθμούς που είναι ίσοι με το άθροισμα των τετάρτων δυνάμεων των ψηφίων τους.

(αυτοί είναι οι 1634, 8208 και 9474)

Δηλαδή: $1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 256 = 1634$

Θέμα 2 (μονάδες 15)

Να συμπληρώσετε τον πίνακα με την αποτίμηση των εκφράσεων C++, όταν έχουν προηγηθεί οι δηλώσεις

```
int i=1, j=2, k=3;  
float a=10.0, b=20.0, c=30.0;  
bool p=TRUE, q=FALSE;
```

ΑΑ	ΕΚΦΡΑΣΗ	ΤΙΜΗ	ΤΥΠΟΣ
	<code>10*(a+b)==300</code>	True	bool
	<code>2*(i+k)-j+j%i</code>	6	integer
1	<code>c<b-80 && b>10*k</code>	FALSE	bool
2	<code>c/a+i*j</code>	5	float
3	<code>(a+b+c)/j-i*c</code>	0	float
4	<code>a/i+(k*c)</code>	100	float
5	<code>a == i*c - 20</code>	TRUE	bool
6	<code>a/i+a/k</code>	40/3	float
7	<code>a * (i%j)+ j%k</code>	12	float
8	<code>p ((a<k) (c>b))</code>	TRUE	bool
9	<code>p q</code>	TRUE	bool
10	<code>(a-j)/c-a</code>	-292/300	float

Θέμα 3 (μονάδες 30)

Σε ένα πρόγραμμα C++ έχουν γίνει οι ακόλουθες δηλώσεις μεταβλητών:

```
int a[5]={3, 2, 4, 5, 0};
int i, j, k, s, x, y, xp, yp, m;
```

1 (μονάδες 20)

Αντιστοιχίστε τα τμήματα προγράμματος στα αριστερά με τις περιγραφές στα δεξιά. Θα περισσέψουν 5 περιγραφές.

1	<pre>s=0; for (i=0; i<5; i++) { s=s+a[i]; } cout<<endl<<"s="<<s;</pre>	E	
2	<pre>x=a[0]; xp=0; for (i=1; i<5;i++) if (a[i]>x) { x=a[i]; xp=i; } cout<<endl<<"xp="<<xp;</pre>	Δ	
3	<pre>m=0; for (i=0; i<5;i++) if (a[i]%2==1) m++; cout<<endl<<"m="<<m;</pre>	B	
4	<pre>x=a[0]; for (i=1; i<5;i++) { if (a[i]>x) x=a[i]; } cout<<endl<<"x="<<x;</pre>		Υπολ. & εκτυπ. το μέγιστο στοιχείο του πίνακα a
5	<pre>y=a[0]; yp=0; for (i=1; i<5;i++) if (a[i]<y) { y=a[i]; yp=i; } cout<<endl<<"yp="<<yp;</pre>	Γ	
6	<pre>for (i=0; i<5;i++) cout<<a[i]<<endl; cout<<endl;</pre>	Λ	
7	<pre>y=a[0]; for (i=1; i<5;i++) { if (a[i]<y) y=a[i]; } cout<<endl<<"y="<<y;</pre>		Υπολ. & εκτυπ. το ελάχιστο στοιχείο του πίνακα a
8	<pre>s=1; for (i=0; i<5; i++) s=s*a[i]; cout<<endl<<"s="<<s;</pre>	Z	
9	<pre>s=0; for (i=0;i<=4;i++) s=s*a[i]; cout<<endl<<"s="<<s;</pre>	Θ	
10	<pre>for (i=1;i<=3;i=i+2) { if (a[i-1]/a[i]<>2) k=k!; }</pre>	N	

- A. Υπολογίζει και εκτυπώνει το πλήθος των άρτιων στοιχείων του πίνακα a
- B. Υπολογίζει και εκτυπώνει το πλήθος των περιπτών στοιχείων του πίνακα a
- Γ. Υπολογίζει και εκτυπώνει τη θέση του ελάχιστου στοιχείου του πίνακα a
- Δ. Υπολογίζει και εκτυπώνει τη θέση του μέγιστου στοιχείου του πίνακα a
- E. Υπολογίζει το και εκτυπώνει το άθροισμα των τιμών των στοιχείων του πίνακα a
- Z. Υπολογίζει και εκτυπώνει το γινόμενο των τιμών των στοιχείων του πίνακα a
- H. Υπολογίζει με λάθος τρόπο και εκτυπώνει το άθροισμα των τιμών των στοιχείων του πίνακα a
- Θ. Υπολογίζει με λάθος τρόπο και εκτυπώνει το γινόμενο των τιμών των στοιχείων του πίνακα a
- I. Υπολογίζει και εκτυπώνει το πλήθος των περιπτών στοιχείων του πίνακα a
- K. Υπολογίζει και εκτυπώνει το πλήθος των άρτιων στοιχείων του πίνακα a
- Λ. Εκτυπώνει όλα τα στοιχεία του πίνακα a, ένα σε κάθε γραμμή
- M. Εκτυπώνει όλα τα στοιχεία του πίνακα a, με κενό μεταξύ τους
- N. Δεν είναι πρόγραμμα C++ (παράγει σφάλμα κατά την μεταγλώττιση)
- Ξ. Παράγει σφάλμα κατά την εκτέλεση

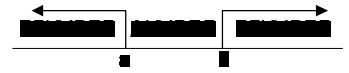
2 (μονάδες 10)

Γράψτε τα τμήματα προγράμματος 4 και 8 χρησιμοποιώντας τη δομή ελέγχου while αντί της δομής for

4	<pre>x=a[0]; i=1; while (i<5) { if (a[i]>x) x=a[i]; i++; } cout<< endl<<"x="<<x;</pre>	8	<pre>s=1; i=0; while (i<5) { s=s*a[i]; i++; } cout<< endl<<"s="<<s;</pre>
---	--	---	--

Θέμα 4 (μονάδες 15)

Γράψτε ένα πρόγραμμα C++ που διαβάζει από το πληκτρολόγιο την τιμή μιας μεταβλητής x εξασφαλίζοντας ότι αυτή ΔΕΝ ανήκει στο διάστημα $[a, b]$ όπου $a \leq b$



```
float a, b, x;
// Είσοδος τιμών για τα a και b
do
    cout<<"\n Enter x outside range ("<<a<<","<<b<<" : ";
    cin>>x;
while ((x>=a) && (x<=b));
```

Θέμα (μονάδες 25)

Γράψτε ένα πρόγραμμα C++ που εναλλάσσει τα στοιχεία των γραμμών ενός τετραγωνικού πίνακα ακεραίων $N \times N$ με αυτά των στηλών του. Μπορείτε να χρησιμοποιήσετε τη συνάρτηση `swap(a,b)` η οποία αντιμεταθέτει τις τιμές των `a` και `b`. Επιδιώξτε το πρόγραμμά σας να κάνει τις ελάχιστες δυνατές πράξεις.

1	2	3		1	4	7
4	5	6	↔	2	5	8
7	8	9		3	6	9

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

void swap(float &a, float &b)           // συνάρτηση αντιμετάθεσης τιμών

int main()
{
    int i, j, n;
    n=3;           // διάσταση του πίνακα
    float p[n][n];

    // Είσοδος τιμών του πίνακα

    for (i=0;i<n;i++)
        for (j=i+1;j<n;j++)
            swap(p[i][j],p[j][i]);

    return 0;
}
```

Εμβέλεια μεταβλητών

Ο όρος **εμβέλεια** μιας μεταβλητής (scope of name) αναφέρεται στο τμήμα του προγράμματος στο οποίο έχει εφαρμογή η δήλωση μιας μεταβλητής, δηλαδή αναγνωρίζεται η μεταβλητή.

Στην C++, η βασική μονάδα (τμήμα) προγράμματος είναι το **block**, μια ομάδα εντολών που περικλείεται σε άγκιστρα {}. Η εμβέλεια των ονομάτων των μεταβλητών προσδιορίζεται από μερικούς βασικούς κανόνες:

1. Ονόματα μεταβλητών ορισμένα σ' ένα block έχουν τοπική (local) ισχύ, μέσα στο block αυτό.
 2. Αν ένα block περιέχεται σ' ένα άλλο (είναι δηλαδή «θυγατρικό») τότε 'κληρονομεί' τις μεταβλητές του «μητρικού» block.
 3. Μεταβλητές που ορίζονται έξω από οποιοδήποτε block θεωρούνται 'καθολικές' (global) και έχουν ισχύ από το σημείο ορισμού τους μέχρι το τέλος του πηγαίου κώδικα.
 4. Μια τοπική μεταβλητή υπερισχύει μιας καθολικότερης μεταβλητής με το ίδιο όνομα.
- Ίδιοι κανόνες εμβέλειας ισχύουν και για τα άλλα προσδιοριστικά (identifiers), όπως π.χ. τα ονόματα των συναρτήσεων.

Κάθε συνάρτηση ενός προγράμματος αποτελεί ένα block (που πιθανόν περικλείει άλλα μικρότερα block στο εσωτερικό του), επομένως ισχύουν οι παραπάνω κανόνες. Όμως, η C++ δεν επιτρέπει τον ορισμό μιας συνάρτησης στο εσωτερικό μιας άλλης, δεν υπάρχουν δηλαδή 'θυγατρικές' συναρτήσεις.

Εμβέλεια μεταβλητών - παράδειγμα

```

#include <iostream>
using namespace std;

const double rate = 10.50;
int z;

void Ftwo(int a, int b, char x);
void Fthree(int one, double y, int z);

int main () {
    int num, first;
    double x, y, z;
    char name, last;
    ...
return 0; }

int w;

void Ftwo(int a, int b, char x) {
    int count;
    ... }

void Fthree(int one, double y, int z)
{
    int a;
    ...
    //BlockFour
    { char a;
    ...
    }//end BlockFour
    ... }

```

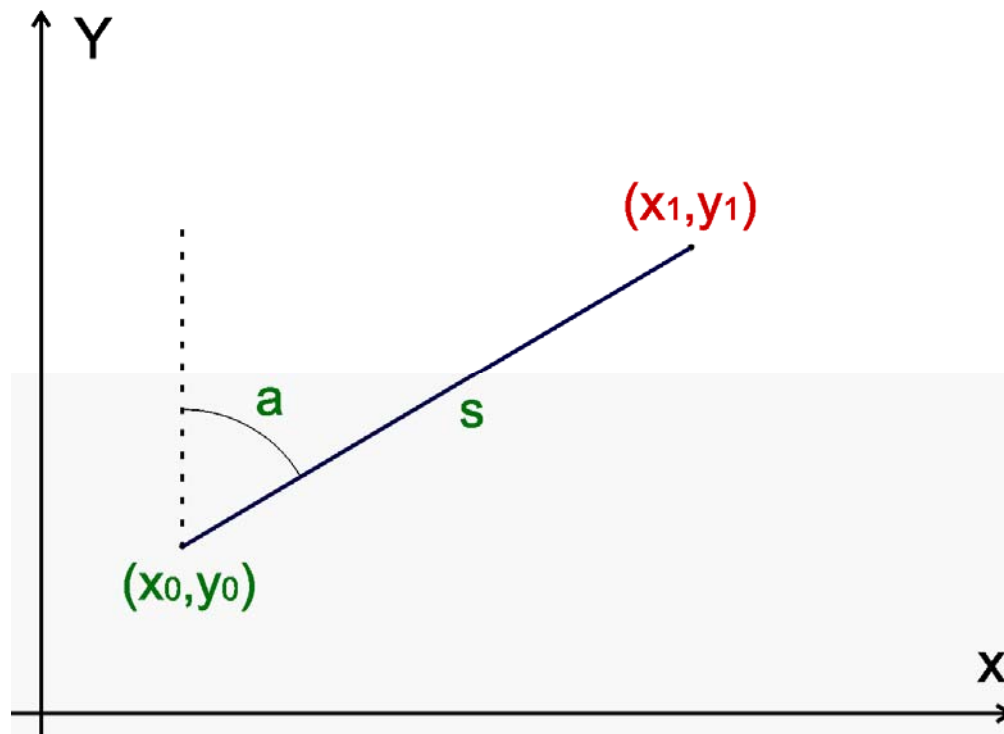
Εμβέλεια των μεταβλητών στα διάφορα τμήματα

Όνομα	Ftwo	Fthree	BlockFour	main
Καθολικές: rate	NAI	NAI	NAI	NAI
z	NAI	OXI	OXI	OXI
w	NAI	NAI	NAI	OXI
main: (function name)	NAI	NAI	NAI	NAI
Τοπικές της main	OXI	OXI	OXI	NAI
Ftwo: (function name)	NAI	NAI	NAI	NAI
a (παράμετρος)	NAI	OXI	OXI	OXI
b (παράμετρος)	NAI	OXI	OXI	OXI
x (παράμετρος)	NAI	OXI	OXI	OXI
Τοπικές της Ftwo	NAI	OXI	OXI	OXI
Fthree: (function name)	NAI	NAI	NAI	NAI
one (παράμετρος)	OXI	NAI	NAI	OXI
y (παράμετρος)	OXI	NAI	NAI	OXI
z (παράμετρος)	OXI	NAI	NAI	OXI
Τοπικές της Fthree	OXI	NAI	NAI	OXI
BlockFour: a (τοπική)	OXI	OXI	NAI	OXI

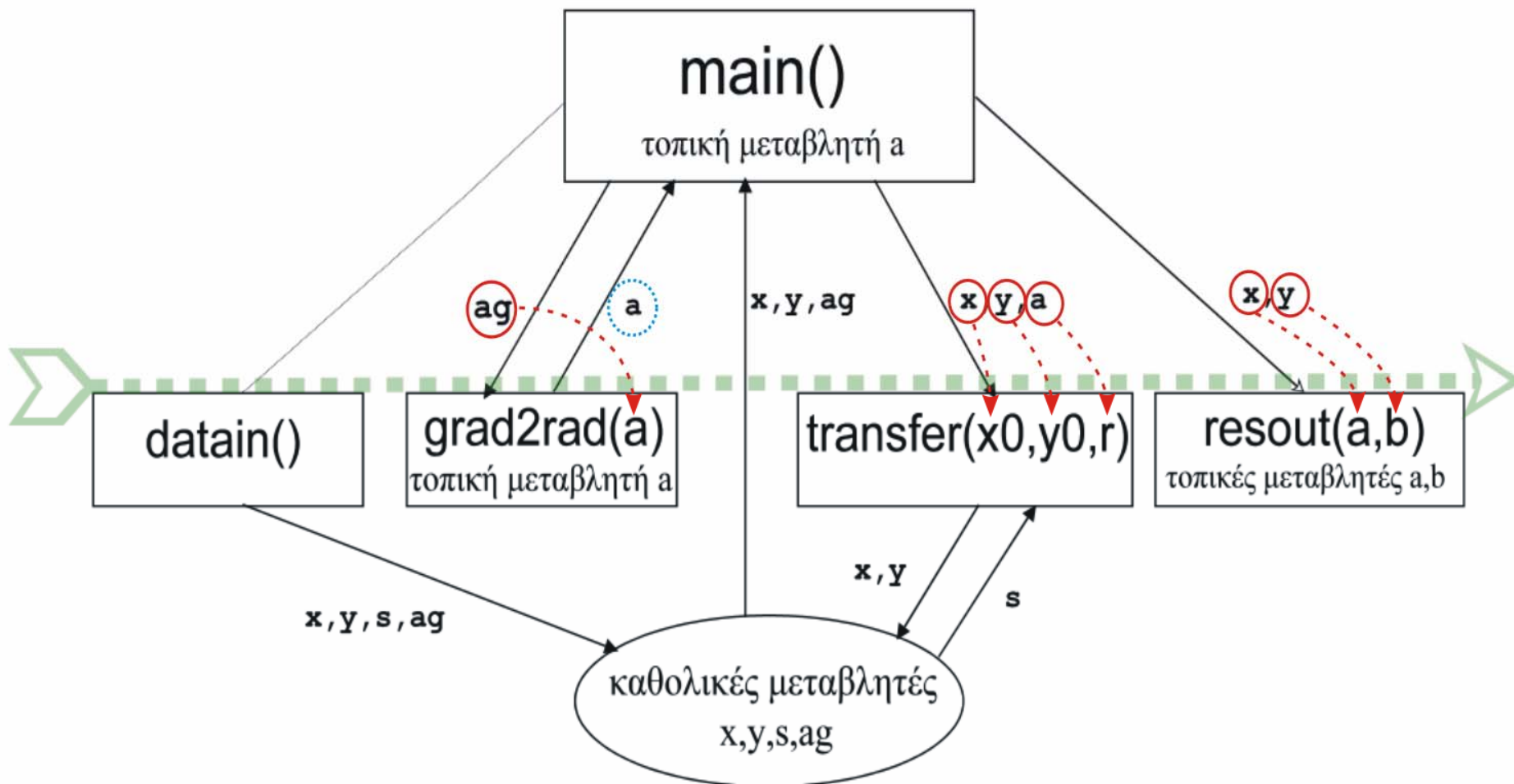
Παράδειγμα - Συναρτήσεις και εμβέλεια μεταβλητών

Πρόβλημα:

Είναι γνωστές οι ορθογώνιες συντεταγμένες (x_0, y_0) ενός σημείου T_0 και μετράται η απόσταση s και η γωνία διεύθυνσης a προς το σημείο T_1 . Να υπολογιστούν οι ορθογώνιες συντεταγμένες (x_1, y_1) του T_1 .



Παράδειγμα - Συναρτήσεις και εμβέλεια μεταβλητών



Παράδειγμα - Συναρτήσεις και εμβέλεια μεταβλητών

```
//  
// Programming Techniques  
//  
// Using functions  
//  
// Example : Scope of variable names  
// Transfer of orthogonal coordinates  
//  
  
#include <iostream>  
#include <cmath>  
using namespace std;  
  
// Global variables  
  
float x,y,s,ag;  
  
void datain()  
{  
    cout<<"\nEnter x0, y0 : ";  
    cin>>x>>y;  
    cout<<"\nEnter distance s : ";  
    cin>>s;  
    cout<<"\nEnter direction angle (grad) : ";  
    cin>>ag;  
    return;  
}
```

```
void resout(float a, float b)  
{  
    cout<<"\nOrthogonal coordinates are :\n";  
    cout<<"X = "<<a<<"\tY = "<<b;  
    return;  
}  
  
float grad2rad(float a)  
{  
    const float pi=4.0*atan(1.0);  
    return a*pi/200;  
}  
  
void transfer(float x0, float y0, float r)  
{  
    x=x0+s*sin(r);  
    y=y0+s*cos(r);  
    return;  
}  
  
int main()  
{  
    float a;  
    datain();  
    a=grad2rad(ag);  
    transfer(x,y,a);  
    resout(x,y);  
    return 0;  
}
```