



Όνοματεπώνυμο Μαθητή:..... Ημ/νία:.....

Ορισμός και κλήση συναρτήσεων

Η συνάρτηση (function) είναι ένας τύπος υποπρογράμματος που υπολογίζει και επιστρέφει μόνο μία τιμή στο κυρίως πρόγραμμα.

```
#include <iostream>
using namespace std;
int add_2_nums(int a, int b){           // Ορισμός συνάρτησης
    return a + b;                       // Επιστροφή αθροίσματος
}                                       // Τέλος σώματος συνάρτησης
int main() {                            // Κύρια συνάρτηση (main)
    cout << add_2_nums(65, 97);        // Κλήση συνάρτησης add_2_nums
    return 0;                           // στο κυρίως πρόγραμμα
}                                       // Τέλος προγράμματος
```

Ορισμός και κλήση συναρτήσεων

```
int add_2_nums(int a, int b){
```

Από την επικεφαλίδα συμπεραίνουμε τα εξής:

- Η συνάρτηση θα επιστρέφει έναν ακέραιο αριθμό γιατί είναι τύπου integer
- Το όνομα της συνάρτησης είναι add_2_nums
- Η συνάρτηση δέχεται δύο παραμέτρους (a, b) ως ακέραιες τιμές
-

Η κλήση της συνάρτησης γίνεται στο κυρίως πρόγραμμα ως εξής:

```
cout << add_2_nums(65, 97);           // Αποτέλεσμα: 162
```

Τοπικές (local) και καθολικές (global) μεταβλητές

Μέσα σε μία συνάρτηση μπορούμε να δηλώσουμε τοπικές μεταβλητές.

- Οι τοπικές μεταβλητές μπορούν να χρησιμοποιηθούν μέσα σε μία δηλωμένη συνάρτηση, όμως δεν είναι προσβάσιμες από την κύρια συνάρτηση (main).
- Αντίθετα, μία καθολική μεταβλητή η οποία δηλώνεται εκτός της κύριας συνάρτησης, μπορεί να είναι προσβάσιμη από το κυρίως πρόγραμμα αλλά και από όσες συναρτήσεις ορίζονται σε αυτό.

**Παράδειγμα 1**

Να δημιουργήσετε πρόγραμμα το οποίο να καλεί μία συνάρτηση η οποία να δέχεται παραμετρικά δύο ακέραιους αριθμούς και να επιστρέφει τον μεγαλύτερο από τους δύο. Να θεωρήσετε ότι οι δύο ακέραιοι που θα δοθούν δεν θα είναι ίσοι.

Λύση

```

1. #include <iostream>
2. using namespace std;
3.
4. int max_num(int a, int b) { // Ορισμός συνάρτησης
5. if (a>b) // Αν a>b η συνάρτηση θα επιστρέψει
6. return a; // το a και θα τερματίσει
7. return b; // Αλλιώς θα επιστρέψει το b. Το
8. } // else μπορούμε να το παραβλέψουμε
9.
10. int main() { // Κύρια συνάρτηση (main)
11. int num1, num2;
12. cin >> num1 >> num2; // Καταχώριση τιμών
13. cout << max_num(num1,num2); // Εμφάνιση μέγιστου
14. return 0;
15. }
```

Παράδειγμα 2 - Τοπικές (local) και καθολικές (global) μεταβλητές

```

1. #include <iostream>
2. using namespace std;
3. int glob = 0; // Καθολική (public)
4.
5. int F(int x){
6. int num = 0; // Τοπική (local)
7. glob += 10;
8. num = glob + x;
9. glob = val; // ΣΦΑΛΜΑ
10. return num;
11. }
12.
13. int main(){
14. int val = 0; // Τοπική (local)
15. cin >> val;
16. cout << num << endl; // ΣΦΑΛΜΑ
17. cout << x << endl; // ΣΦΑΛΜΑ
18. cout << glob << endl; // OK
19. cout << F(val) << endl;
20. return 0;
21. }
```

**Παράδειγμα3:**

Το πρόγραμμα αυτό βρίσκει το συνολικό άθροισμα των ψηφίων των μονάδων για άγνωστο πλήθος ακεραίων, καθώς και το πλήθος των ακεραίων. Κάνει χρήση συνάρτησης η οποία υπολογίζει το ψηφίο των μονάδων ενός ακεραίου. Η είσοδος τερματίζει όταν δοθεί ο αριθμός 0.

Λύση

```
1. #include <iostream>
2. using namespace std;
3.
4.
5. int cnt = 0;           // Καθολική (public)
6.
7.
8. int monades(int x) {
9.     int num = x%10;    // Τοπική (local)
10.    cnt++;
11.    return num;
12. }
13.
14. int main(){
15.    int a, sum = 0;
16.    cin >> a;
17.    while(a!=0){
18.        sum += monades(a);
19.        cin >> a;
20.    }
21.    cout << sum << endl << cnt;
22.    return 0;
23. }
```

**Άσκηση 1**

Να βρείτε τι θα επιστρέψει η πιο κάτω συνάρτηση αν δοθεί παραμετρικά η τιμή 25.

```
int check(int a){
if (a>25)
return a / 10;
return a % 10;
}
```

Απάντηση:

Άσκηση 2

Να συμπληρώσετε την πιο κάτω συνάρτηση με το όνομα multiply ή οποία δέχεται παραμετρικά δύο ακέραιους και επιστρέφει το γινόμενο τους.

```
_____ ( _____ x, _____ y) {
_____
}
```

Άσκηση 3

Να συμπληρώσετε τα κενά στο πιο κάτω πρόγραμμα. Η συνάρτηση AverageLastDigit υπολογίζει τον μέσο όρο των ψηφίων των μονάδων 3 ακεραίων αριθμών.

```
#include<iostream>

using namespace std;

float AverageLastDigit(_____) {
return _____;
}

int main(){
int a, b, c;
cin >> a >> b >> c;
_____
return 0;
}
```



Άσκηση 4

Να δημιουργήσετε πρόγραμμα το οποίο να καλεί μία συνάρτηση με το όνομα **next**, η οποία να δέχεται παραμετρικά έναν ακέραιο αριθμό και να επιστρέφει τον επόμενο του.

Παράδειγμα εισόδου

7

Παράδειγμα εξόδου

8

Απάντηση:

Άσκηση 5

Να δημιουργήσετε πρόγραμμα το οποίο να καλεί μία συνάρτηση με το όνομα **power**, η οποία να δέχεται παραμετρικά δύο ακέραιους αριθμούς (x , y) και να επιστρέφει τη δύναμη xy .

Απάντηση:



Άσκηση 6

Να δημιουργήσετε πρόγραμμα το οποίο να καλεί μία συνάρτηση με το όνομα **prod**, η οποία να δέχεται παραμετρικά δύο πραγματικούς αριθμούς και να επιστρέφει το γινόμενό τους, το οποίο να εμφανίζεται στην οθόνη από το κυρίως πρόγραμμα, με δύο δεκαδικά ψηφία.

Sample Input

2.3 6.8

Sample Output

15.64

Απάντηση:

Άσκηση 7

Να δημιουργήσετε πρόγραμμα το οποίο να καλεί μία συνάρτηση με το όνομα **mod**, η οποία να δέχεται παραμετρικά δύο ακεραίους αριθμούς και να επιστρέφει το υπόλοιπο της διαίρεσης του μεγαλύτερου με τον μικρότερο αριθμό. Οι αριθμοί μπορούν να δοθούν με οποιαδήποτε σειρά.

Απάντηση:

**Άσκηση 7**

Να δημιουργήσετε πρόγραμμα το οποίο θα δέχεται ακέραιους αριθμούς μέχρι να δοθεί ο αριθμός -1. Το πρόγραμμα θα καλεί μία συνάρτηση η οποία θα δέχεται παραμετρικά έναν ακέραιο, θα ελέγχει αν είναι μονοψήφιος και να επιστρέφει true αν είναι μονοψήφιος ή false αν δεν είναι. Το πρόγραμμα να εμφανίζει το πλήθος όλων των μονοψήφιων αριθμών και το πλήθος όλων των ακεραίων που έχουν δοθεί. Η εισαγωγή των ακεραίων θα γίνεται στο κυρίως πρόγραμμα.

Παράδειγμα εισόδου

7 43 8 91 3 -1

Παράδειγμα εξόδου

One-digit numbers: 3

Total numbers: 5

Απάντηση:

**Άσκηση 8**

Να δημιουργήσετε πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και έναν χαρακτήρα και να χρησιμοποιεί τη συνάρτηση `triangle`, για να τυπώσει ένα ορθογώνιο τρίγωνο που θα έχει `N` γραμμές και θα σχηματίζεται από τον χαρακτήρα `C`.

Πρότυπο συνάρτησης: `void triangle(int n, char c);`

Απάντηση:



Άσκηση 9

Τέλειος καλείται ένας ακέραιος αριθμός όταν το άθροισμα των θετικών διαιρετών του, εκτός του ιδίου, είναι ίσο με τον αριθμό.

Για παράδειγμα, ο αριθμός 28 είναι τέλειος αριθμός.

Οι διαιρέτες του 28 είναι οι 1, 2, 4, 7, 14 και

το άθροισμα αυτών είναι ίσο με 28 (1+2+4+7+14=28).

Να δημιουργήσετε πρόγραμμα το οποίο να καλεί τη συνάρτηση **perfect**, η οποία να δέχεται παραμετρικά έναν ακέραιο θετικό αριθμό και να επιστρέφει το μήνυμα «**perfect**» σε περίπτωση που ο αριθμός είναι τέλειος, ενώ, σε αντίθετη περίπτωση, να επιστρέφει το μήνυμα «**not perfect**».

Απάντηση:

**Άσκηση 10**

Να γραφεί πρόγραμμα το οποίο θα διαβάζει τέσσερις θετικούς πραγματικούς αριθμούς (να γίνετε έλεγχος έτσι ώστε οι αριθμοί να είναι θετικοί) και θα υπολογίζει το άθροισμα τους. Αν το άθροισμα είναι μεγαλύτερο από 100, τότε θα υπολογίζει με τη βοήθεια της συνάρτησης **MESOS** το μέσο όρο τους και θα τον εμφανίζει. Σε διαφορετική περίπτωση, θα βρίσκει με τη βοήθεια της συνάρτησης **MEGISTOS** το μέγιστο αριθμό τον οποίο και θα εμφανίζει.

Απάντηση:

**Άσκηση 11**

Μια πλάστιγγα (ζυγαριά μεγάλου βάρους) ζυγίζει οχήματα σε ένα τελωνειακό σταθμό. Αν το βάρος του οχήματος είναι μέχρι 2000 κιλά, το όχημα περνάει τον έλεγχο χωρίς επιβάρυνση. Αν το βάρος είναι μεταξύ 2000 και 3000 κιλών, τότε προστίθεται επιβάρυνση 10 σέντ ανά κιλό. Αν το βάρος υπερβαίνει τα 3000 κιλά, τότε η επιβάρυνση είναι 15 σέντ ανά κιλό. Να σχεδιάσετε λογικό διάγραμμα και να γραφεί πρόγραμμα το οποίο να υπολογίζεται η συνολική επιβάρυνση για N οχήματα (N ακέραιος θετικός) ως εξής :

A) στο κυρίως πρόγραμμα να ζητείται το βάρος για κάθε όχημα (με έλεγχο για βάρος μόνο μεγαλύτερο του μηδενός.

B) με τη συνάρτηση TEST η οποία θα δέχεται το βάρος για κάθε όχημα και θα επιστρέφει την επιβάρυνση για αυτό.

Γ) στο κυρίως πρόγραμμα θα εκτυπώνει την συνολική επιβάρυνση.

Απάντηση:



Σ υ ν α ρ τ ή σ ε ι ς τ η ς C++

Συνάρτηση	Χρήση	Παράμετροι	Παράδειγμα
<code>sqrt(x)</code>	Επιστρέφει την τετραγωνική ρίζα του αριθμού x . Η επιστρεφόμενη τιμή είναι δεκαδικός αριθμός.	Ένας θετικός αριθμός (ακέραιος ή δεκαδικός).	<code>sqrt(9)=3.0</code>
<code>abs(x)</code>	Επιστρέφει την απόλυτη τιμή του αριθμού x . Η επιστρεφόμενη τιμή εξαρτάται από τον τύπο του αριθμού x .	Ένας αριθμός (ακέραιος ή δεκαδικός).	<code>abs(-2)=2</code> <code>abs(-3.1)=3.1</code>
<code>pow(x, y)</code>	Επιστρέφει το αποτέλεσμα της δύναμης x^y . Η επιστρεφόμενη τιμή είναι δεκαδικός αριθμός.	Δύο αριθμοί (δεκαδικοί).	<code>pow(2.0, 3.0)=8.0</code> <code>pow(2.0, -1.0)=0.5</code>
<code>trunc(x)</code>	Επιστρέφει το ακέραιο μέρος του αριθμού x σε δεκαδική μορφή, αγνοώντας το δεκαδικό μέρος του.	Ένας δεκαδικός αριθμός.	<code>trunc(1.7)=1.0</code> <code>trunc(-2.1)=-2.0</code>
<code>round(x)</code>	Επιστρέφει το ακέραιο μέρος του αριθμού x σε δεκαδική μορφή, στρογγυλοποιημένο στην πλησιέστερη τιμή.	Ένας δεκαδικός αριθμός.	<code>round(1.5)=2.0</code> <code>round(1.4)=1.0</code>