

Συναρτήσεις

Ιωάννης Γ. Τσούλος

2019

1 Συναρτήσεις

Το επόμενο θέμα που θα ασχοληθούμε είναι οι συναρτήσεις. Με τις συναρτήσεις μπορεί κανείς να γράφει κώδικα ο οποίος μπορεί να γραφεί μια φορά και να καλείται πολλές φορές. Επιπλέον οι συναρτήσεις μπορούν όπως και στα μαθηματικά να δεχθούν δεδομένα σαν είσοδο (ορίσματα) και να δώσουν έξοδο κάποια τιμή.

1.1 Συναρτήσεις χωρίς ορίσματα

Στο παράδειγμα του αλγορίθμου 1 η συνάρτηση `printMessage()` καλείται σε πολλά σημεία της `main()`.

Η λέξη κλειδί `void` στην συνάρτηση σημαίνει πως αυτή η συνάρτηση δεν πρόκειται να επιστρέψει τιμή. Φυσικά μια συνάρτηση μπορεί να γίνει πιο χρήσιμη από το να τυπώνει μηνύματα στην οθόνη. Στο παράδειγμα του αλγορίθμου 2 η συνάρτηση `computeAvg()` διαβάζει 5 αριθμούς από το πληκτρολόγιο, βρίσκει τον μέσο όρο τους και τον εμφανίζει.

1.2 Συναρτήσεις με ορίσματα

Αν και οι συναρτήσεις χωρίς ορίσματα χρησιμοποιούνται αρκετά, συνήθως εμφανίζονται με ορίσματα. Αυτό γίνεται γιατί θέλουμε η εκτέλεση της συνάρτησης να διαφοροποιείται ανάλογα με την είσοδο που παρέχει ο χρήστης. Στο παράδειγμα 3 αλλάζουμε την συνάρτηση `computeAvg()` ώστε να δέχεται είσοδο από τον χρήστη. Οι παράμετροι στις συναρτήσεις είναι τοπικές μεταβλητές σε αυτές και γράφονται μέσα στις παρενθέσεις της συνάρτησης. Ο χρήστης θα πρέπει να δώσει στην συνάρτηση από την οποία γίνεται η κλήση (η `main` στην περίπτωση μας) τιμές για κάθε όρισμα της συνάρτησης.

1.3 Συναρτήσεις με ορίσματα και επιστροφή

Η συνάρτηση `computeAvg()` μπορεί να γίνει πιο χρήσιμη αν επιστρέφει στην καλούσα συνάρτηση την τιμή που υπολογίζει αντί να την εμφανίζει. Αυτό είναι η συνήθης πρακτική όπως και στα μαθηματικά. Στον αλγόριθμο 4 παρουσιάζεται μια εκδοχή της συνάρτησης `computeAvg()` με επιστροφή τιμών στον χρήστη. Μπορούμε να

Algorithm 1 Πολλαπλή κλήση της ίδιας συνάρτησης για εμφάνιση μηνυμάτων.

```
1 # include <stdio.h>
2
3 void    printMessage ()
4 {
5     printf ("Message_from_function\n");
6     printf ("Another_message_from_function\n");
7 }
8
9 int main ()
10 {
11     int counter;
12     printMessage ();
13     for (counter=1;counter <=5;counter++)
14     {
15         printf ("Kaloume_tin_synartisi_kai_pali\n");
16         printMessage ();
17     }
18     return 0;
19 }
```

κάνουμε το πρόγραμμα ακόμα πιο δομημένο μεταφέροντας την ανάγνωση των δεκαδικών αριθμών σε ξεχωριστή συνάρτηση με το όνομα readDouble(), όπως παρουσιάζεται στον αλγόριθμο 5.

Μια ακόμα χρήσιμη συνάρτηση που εμφανίζεται σε πολλά προβλήματα είναι η συνάρτηση του παραγωντικού. Σε αυτήν θα πρέπει ο χρήστης να υπολογίσει το

$$N!=1\times 2\times 3\ldots\times N$$

Στον αλγόριθμο 6 παρουσιάζεται μια πρώτη εκδοχή για το πρόβλημα αυτό με επαναληπτικό τρόπο. Ωστόσο τις περισσότερες φορές αυτό το πρόβλημα αντιμετωπίζεται με την χρήση αναδρομής, δηλαδή με κλήση της συνάρτησης από την ίδια, όπως παρουσιάζεται και στον αλγόριθμο 7.

Algorithm 2 Ανάγνωση 5 αριθμών και εμφάνιση του μέσου όρου τους με την χρήση συνάρτησης.

```
1 #include <stdio.h>
2
3 void computeAvg()
4 {
5     int counter;
6     double sum=0.0;
7     double x, average;
8     for(counter=1;counter<=5;counter++)
9     {
10         printf("Doste_arithmo_");
11         scanf("%lf",&x);
12         sum=sum+x;
13     }
14     average=sum/counter;
15     printf("O_mesos_oros_5_arithmon_einai_%lf\n",average);
16 }
17
18 int main()
19 {
20     computeAvg();
21     return 0;
22 }
```

Algorithm 3 Μέσος όρος N αριθμών με υπολογισμό από συνάρτηση.

```
1 # include <stdio.h>
2
3 void computeAvg(int n)
4 {
5     int counter;
6     double sum=0.0;
7     double x, average;
8     for(counter=1;counter<=n;counter++)
9     {
10         printf("Doste_arithmo_");
11         scanf("%lf",&x);
12         sum=sum+x;
13     }
14     average=sum/n;
15     printf("O_mesos_oros_%d_arithmon_einai_%lf\n",n, average);
16 }
17
18 int main()
19 {
20     int numbers;
21     printf("Gia_posous_arithmous_thelete_meso_oro_");
22     scanf("%d",&numbers);
23     computeAvg(numbers);
24     return 0;
25 }
```

Algorithm 4 Υπολογισμός μέσου όρου αριθμών με συνάρτηση και επιστροφή τιμής.

```
1  # include <stdio.h>
2
3  double computeAvg(int n)
4  {
5      int counter;
6      double sum=0.0;
7      double x, average;
8      for(counter=1;counter<=n;counter++)
9      {
10         printf("Doste_arithmo_";
11         scanf("%lf",&x);
12         sum=sum+x;
13     }
14     average=sum/n;
15     return average;
16 }
17
18 int main()
19 {
20     int n1,n2;
21     double avg1,avg2;
22     printf("Gia_posous_arithmous_thelete_meso_oro_");
23     scanf("%d",&n1);
24     avg1=computeAvg(n1);
25     printf("Gia_posous_arithmous_thelete_meso_oro_");
26     scanf("%d",&n2);
27     avg2=computeAvg(n2);
28     printf("Oi_mesoi_oi_einai_%lf_%lf_\n",avg1,avg2);
29     return 0;
30 }
```

Algorithm 5 Χρήση ξεχωριστής συνάρτησης για την ανάγνωση δεκαδικών αριθμών για το πρόβλημα του μέσου όρου.

```
1 # include <stdio.h>
2
3 double readDouble()
4 {
5     double x;
6     printf("Doste_arithmo_");
7     scanf("%lf",&x);
8     return x;
9 }
10
11 double computeAvg(int n)
12 {
13     int counter;
14     double sum=0.0;
15     double x, average;
16     for(counter=1;counter<=n;counter++)
17     {
18         x=readDouble();
19         sum=sum+x;
20     }
21     average=sum/n;
22     return average;
23 }
24
25 int main()
26 {
27     int n1,n2;
28     double avg1,avg2;
29     printf("Gia_posous_arithmous_thelete_meso_oro_");
30     scanf("%d",&n1);
31     avg1=computeAvg(n1);
32     printf("Gia_posous_arithmous_thelete_meso_oro_");
33     scanf("%d",&n2);
34     avg2=computeAvg(n2);
35     printf("Oi_mesoi_oi_einai_%lf_%lf_\n",avg1,avg2);
36     return 0;
37 }
```

Algorithm 6 Υπολογισμός του παραγοντικού με επαναληπτικό τρόπο.

```
1 # include <stdio.h>
2
3 int factorial(int n)
4 {
5     int p=1;
6     int i;
7     for(i=1;i<=n;i++)
8     {
9         p=p*i;
10    }
11    return p;
12 }
13
14 int main()
15 {
16     int number;
17     printf("Doste_arithmo_");
18     scanf("%d",&number);
19     printf("To_paragontiko_tou_einai_%d\n",factorial(number));
20     return 0;
21 }
```

Algorithm 7 Υπολογισμός παραγωντικού με την χρήση αναδρομής.

```
1 # include <stdio.h>
2
3 int factorial(int n)
4 {
5     if(n<=1) return 1;
6     else return n*factorial(n-1);
7 }
8
9 int main()
10 {
11     int number;
12     printf("Doste_arithmo_");
13     scanf("%d",&number);
14     printf("To_paragontiko_tou_einai_%d\n",factorial(number));
15     return 0;
16 }
```
