

Ασκήσεις επί των μεθόδων Διχοτόμησης & Regula falsi

1) Να γραφεί πρόγραμμα στη C++, το οποίο βρίσκει τη ρίζα της $f(x) = x^3 + x + 1$ στο διάστημα $[-1,1]$ με ακρίβεια $\varepsilon = 10^{-10}$.

Όταν γράψετε και “τρέξετε” το πρόγραμμα θα πρέπει να βρείτε τη ρίζα $-0,6823278038$ και 35 επαναλήψεις.

2) Να γραφεί πρόγραμμα διχοτόμησης στη C++ για τη συνάρτηση $f(x) = x^5 + x + 1$ στο διάστημα $[-1,1]$ με ακρίβεια $\varepsilon = 10^{-10}$.

Στο πρόγραμμα αυτό αντί του επαναληπτικού βρόχου for να χρησιμοποιηθεί η εντολή while (που είναι επίσης δομή επανάληψης), με συνθήκη ισχύος των περικλειόμενων εντολών της, τη : $|b - a| > \varepsilon$ (όπου a και b τα άκρα του διαστήματος ορισμού της συνάρτησης). Οι επαναλήψεις n της μεθόδου για την εύρεση ρίζας, να μην υπολογιστούν από το γενικό τύπο της άσκησης 1, αλλά να βρεθούν μέσα από την εντολή while. Τέλος, να χρησιμοποιηθεί και πάλι το κριτήριο ελέγχου if – else, για την ισχύ της μεθόδου (If $(f(a)*f(b) \leq 0)$ then else....).

Όταν γράψετε και “τρέξετε” το πρόγραμμα θα πρέπει να βρείτε τη ρίζα $-0,7548776663$ και 35 επαναλήψεις.

3) Να γραφεί πρόγραμμα στη C++ με τη μέθοδο Regula falsi για τον υπολογισμό ρίζας της $f(x) = x^3 + x + 1$ στο διάστημα $[-1,1]$. Να χρησιμοποιηθεί η εντολή do – while με συνθήκες ισχύος των περικλειόμενων εντολών της, τις: $|b - a| > \varepsilon_1$ και $|f(x)| > \varepsilon_2$, όπου $\varepsilon_1 = \varepsilon_2 = 10^{-10}$

(Στη μέθοδο αυτή ένα υποψήφιο άκρο των διαστημάτων σας κάθε φορά, θα είναι:

$$x = a - \frac{f(a) * (b - a)}{f(b) - f(a)} \quad \text{και} \quad \text{όχι} \quad x = \frac{(b + a)}{2}, \quad \text{που είχαμε στη μέθοδο της διχοτόμησης).}$$

Όταν γράψετε και “τρέξετε” το πρόγραμμα θα πρέπει να βρείτε τη ρίζα $-0,6823278038$ και 17 επαναλήψεις (πολύ λιγότερες από αυτές της μεθόδου διχοτόμησης).

Λύσεις

1)

```
#include <stdio.h>
#include<stdlib.h>
#include <math.h>
double f(double);
int main()
{
    double a,b,x,err;
    int n, i;
    a=-1.0;
    b=1.0;
    err=1.e-10;
    n=(int) (log((b-a)/err)/log(2.0))+1;
    if (f(a)*f(b)<=0)
    {
        for(i=1;i<=n;i++)
        {
            x=(a+b)/2;
            if (f(x)==0) goto label;
            if (f(x)*f(b)<0) a=x;
            else b=x;
        }
        label: printf("root is %14.10lf\n",x);
        printf("oi epanalhpseis einai:%d\n",n);
    }

    else

        printf("criterion not valid\n");
    system("pause");
    return 0;
} //end main

double f(double x)
{
    return x*x*x+x+1;
}
```

2)

```
#include <stdio.h>
#include<stdlib.h>
#include <math.h>
double f(double);
int main()
{
    double a,b,x,err;
    int n;
    a=-1.0;
    b=1.0;
    err=1.e-10;
    n=0;

    if (f(a)*f(b)<=0)
    {
        while (fabs(b-a)>err)
        {
            n++;
            x=(a+b)/2;
            if (f(x)==0) goto label;
            if (f(x)*f(b)<0) a=x;
            else b=x;
        }
        label: printf("root is %14.10lf\n",x);
        printf(" kai prokypetei meta apo %d epanalhpseis\n",n);
    }

    else
        printf("criterion not valid\n");
    system("pause");
    return 0;

} //end main
double f(double x)
{
    double fx;
    fx=pow(x,5)+x+1;
    return fx;
}
```

3)

```
#include <stdio.h>
#include<stdlib.h>
#include <math.h>
double f(double);
int main()
{
    double a,b,e1,e2,x,ar,par;
    int n;
    a=-1.0;
    b=1.0;
    e1=1.e-10;
    e2=1.e-10;
    n=0;

    if (f(a)*f(b)<=0)
    {
        do
        {
            ar=(b-a)*f(a);
            par=f(b)-f(a);
            x=a-(ar/par);
            if (f(x)*f(b)<0) a=x;
            else b=x;
            n++;
        }
        while (fabs (b-a) > e1 && fabs(f(x))>e2);
        printf("root is %14.10lf\n",x);
        printf(" kai prokypetei meta apo %d epanalhpseis\n",n);
    }

    else
        printf("criterion not valid\n");
    system("pause");
    return 0;
}

//end main

double f(double x)
{
    double fx;
```

```
    fx=pow(x,3)+x+1;  
    return fx;  
}
```