

Politechnika Śląska
Wydział Automatyki Elektroniki i Informatyki
Kierunek: Automatyka i Robotyka sem.3

Gliwice
Rok akademicki 2007/2008
Semestr zimowy

Metody Numeryczne

Laboratorium

Ćw.2 : Obliczanie wartości funkcji

Wykonali:

Adam DZIENDZIEL
Adrian BIELEC

Grupa 4 Sekcja 1

Data odbycia ćwiczenia:
19.10.07

Zadanie 1

```
#include <iostream>
#include <math.h>
#include <conio.h>
#include <time.h>
using namespace std;

int main()
{
    int n,i,j,k;
    float *tab,*tabl,wyn=0,x0;

//zczytanie i przedstawienie danych na monitorze

    cout<<"Podaj stopien wielomianu >>> "; cin>>n;
    cout<<"Podaj argument dla ktorego wyznaczyć wartość k-tej pochodnej wielomianu
>>> "; cin>>x0;
    cout<<"Podaj rząd pochodnej (k) >>> "; cin>>k;
    cout<<endl;

    tab=(float*)malloc(n+1*sizeof(float));
    tabl=(float*)malloc(n+1*sizeof(float));

    for(i=0;i<=n;i++)
    {
        cout<<"podaj współczynnik przy "<<i<<" potędze >>> ";
        cin>>tab[i];
    }
    cout<<endl<<"wielomian postaci : W(x)=";

    for(i=n;i>1;i--)
    {
        cout<<tab[i]<<"x^"<<i<<" + ";
    }
    cout<<tab[1]<<"x + "<<tab[0]<<endl<<endl;

//Schemat Hornera

    tabl[n]=tab[n];
    for(i=n-1;i>=0;i--)
    {
        tabl[i]=tab[i]+(tabl[i+1]*x0);
    }

    cout<<"Za pomocą schematu Hornera W("<<x0<<" ) = "<<tabl[0]<<endl;
    cout<<"policzony w "<<t2<<" s "<<endl<<endl;

    //system("PAUSE");
    //return 0;
    getch();
}
```

Program został przetestowany dla kolejnych stopni wielomianu w celu porównania czasu zliczeń:

STOPIEŃ WIELOMINU	Czas obliczeń w sekundach	
	Za pomocą wzoru ogólnego	Za pomocą schematu Hornera
2	1.3×10^{-6}	1×10^{-7}
5	2.6×10^{-6}	1×10^{-7}
10	4.6×10^{-6}	3×10^{-7}
15	6.8×10^{-6}	4×10^{-7}
20	8.8×10^{-6}	5×10^{-7}
30	1.3×10^{-5}	7×10^{-7}
40	1.71×10^{-5}	1.1×10^{-6}

Zadanie 2

```
#include <iostream>
#include <math.h>
#include <conio.h>
using namespace std;

int main()
{
    int n,i,j,powt=1,war1,war2;
    float *tab,*tab1,wx=0,x0,x1;

    //zczytanie i przedstawienie danych na monitorze

    cout<<"Podaj stopien wielomianu >>> "; cin>>n;
    cout<<endl;

    tab=(float*)malloc((n+1)*sizeof(float));
    tab1=(float*)malloc((n+1)*sizeof(float));

    for(i=0;i<=n;i++)
    {
        cout<<"podaj wspolczynnik przy "<<i<<" potedze >>> ";
        cin>>tab[i];
    }
    cout<<endl<<"wielomian postaci : W(x)=";

    for(i=n;i>1;i--)
    {
        cout<<tab[i]<<"x^"<<i<<" + ";
    }
    cout<<tab[1]<<"x + "<<tab[0]<<endl<<endl;

    //zamiana znaków współczynników na przeciwne dla ujemnego a(n)
    if (tab[n]<0)
    {
        for(i=0;i<=n;i++)
        {
            tab[i]=-tab[i];
        }
    }
}
```

```

//wyznaczania wart przedz //ograniczenie górne
x0=0;
do
{
x0=x0+0.1;
war1=0;
for(i=0;i<=n;i++)
{
tabl[i]=0;
}

tabl[n]=tab[n];
for(i=n-1;i>=0;i--)
{
tabl[i]=tab[i]+(tabl[i+1]*x0);
}

for(i=0;i<=n;i++)
{
if (tabl[i]>=0) war1++;
}

}while(war1<=n);

cout<<endl<<endl<<"ograniczenie górne >>> "<<x0<<endl<<endl;

//ograniczenie dolne

for(i=0;i<=n;i++)
{
tabl[i]=tab[i]*pow((-1.0),n);
tabl[i]=tab[i]*pow((-1.0),(i+2));
}
x1=0;

do
{
x1=x1+0.1;
war2=0;
for(i=0;i<=n;i++)
{
tabl[i]=0;
}

tabl[n]=tab[n];
for(i=n-1;i>=0;i--)
{
tabl[i]=tab[i]+(tabl[i+1]*x1);
}

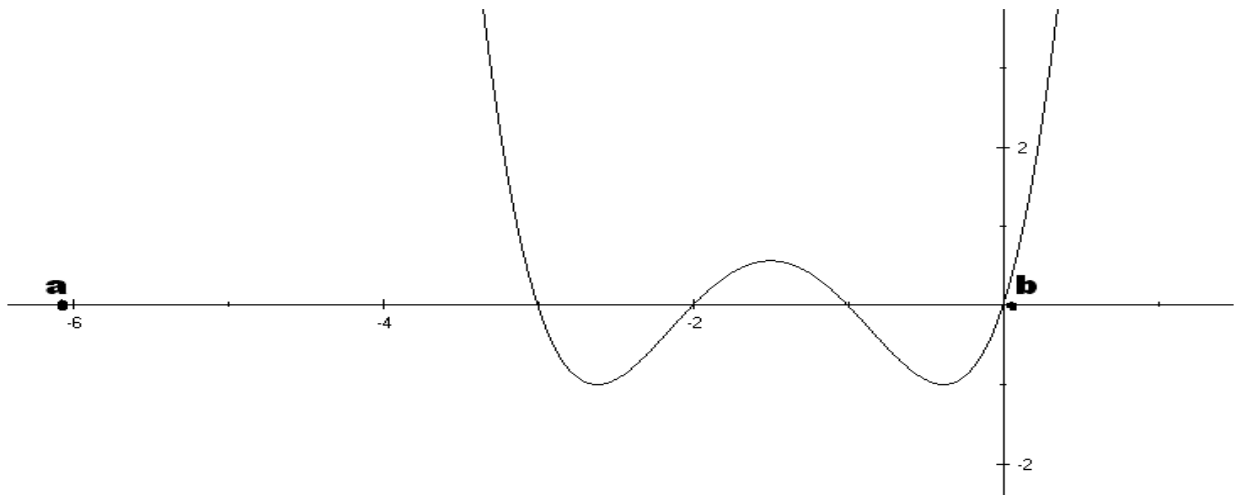
for(i=0;i<=n;i++)
{
if (tabl[i]>=0) war2++;
}
}while(war2<=n);
cout<<endl<<endl<<"ograniczenie dolne >>> "<<-x1<<endl<<endl;

getch();
}

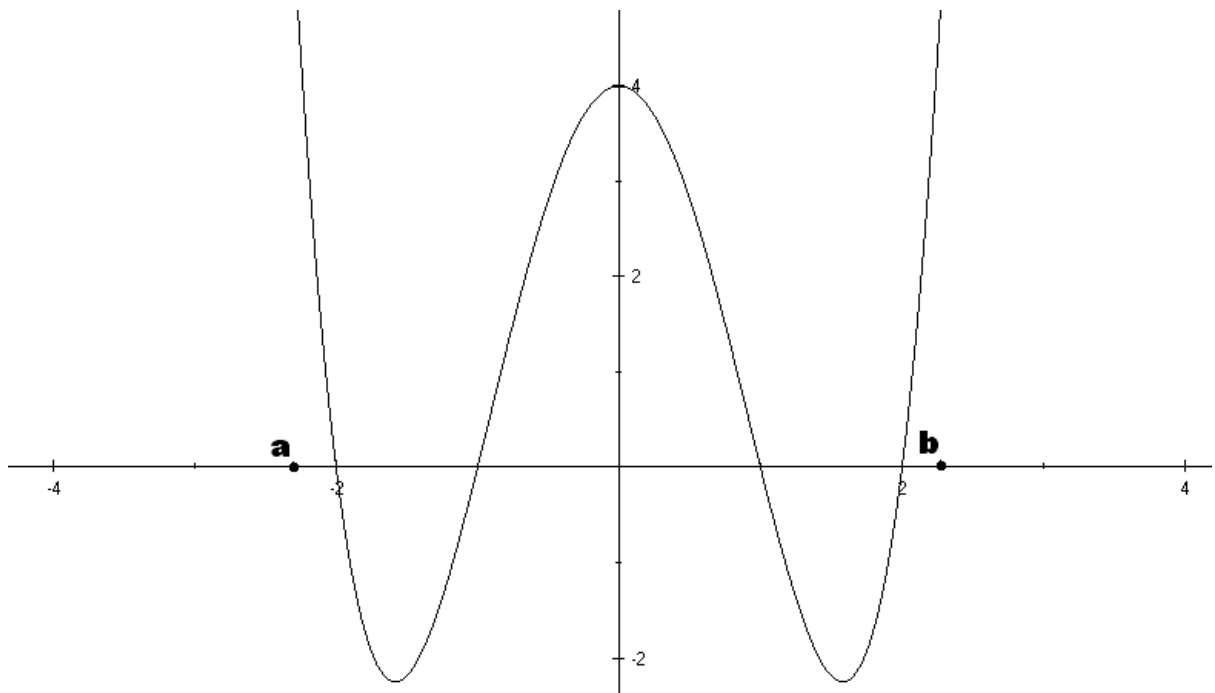
```

Program testowaliśmy dla różnych funkcji o różnym rozkładzie miejsc zerowych (**a**-to dolne pgraniczenie przedziału wyliczone za pomocą programu a **b** to ograniczenie górne) oto niektóre z analizowanych wielomianów:

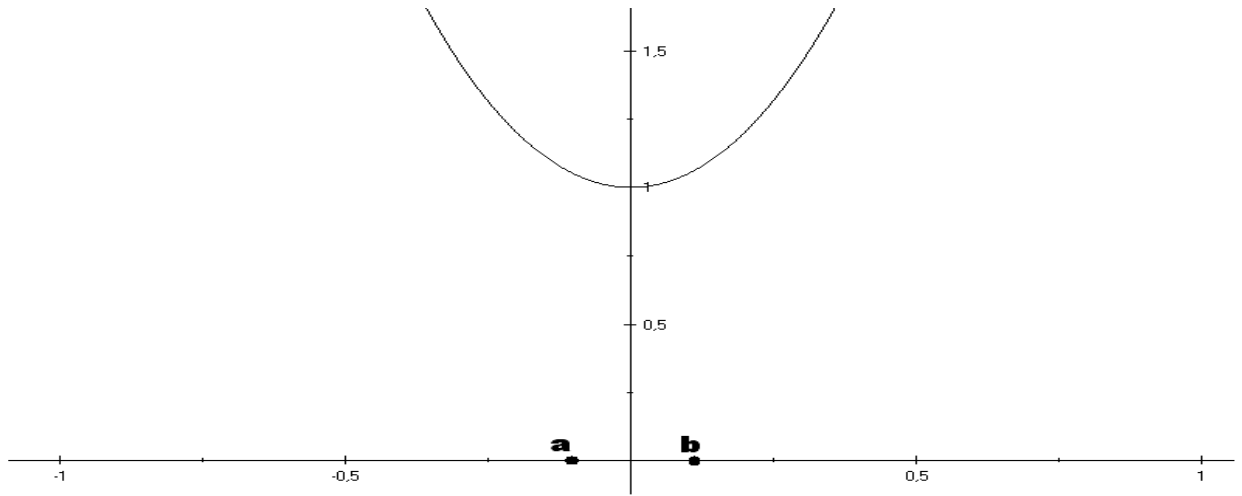
- $F_1(x) = x^4 + 6x^3 + 11x^2 + 6x$
 $x_0 = \{-3 ; -2 ; -1 ; 0 \}$
 $\mathbf{a} = -6,1$
 $\mathbf{b} = 0,1$



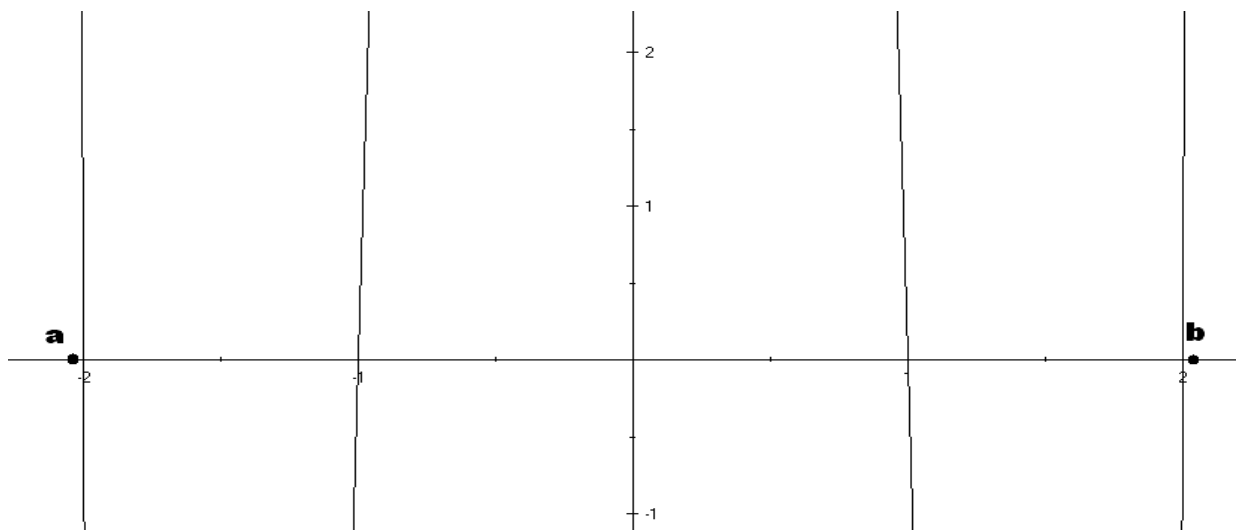
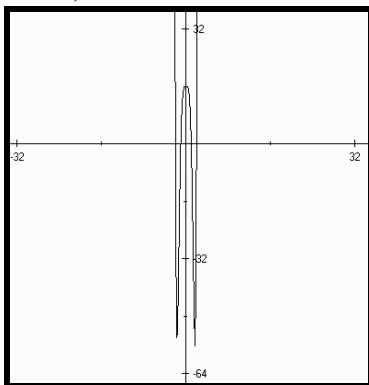
- $F_2(x) = x^4 - 5x^2 + 4$
 $x_0 = \{-2 ; -1 ; 1 ; 2 \}$
 $\mathbf{a} = -2.3$
 $\mathbf{b} = 2.3$



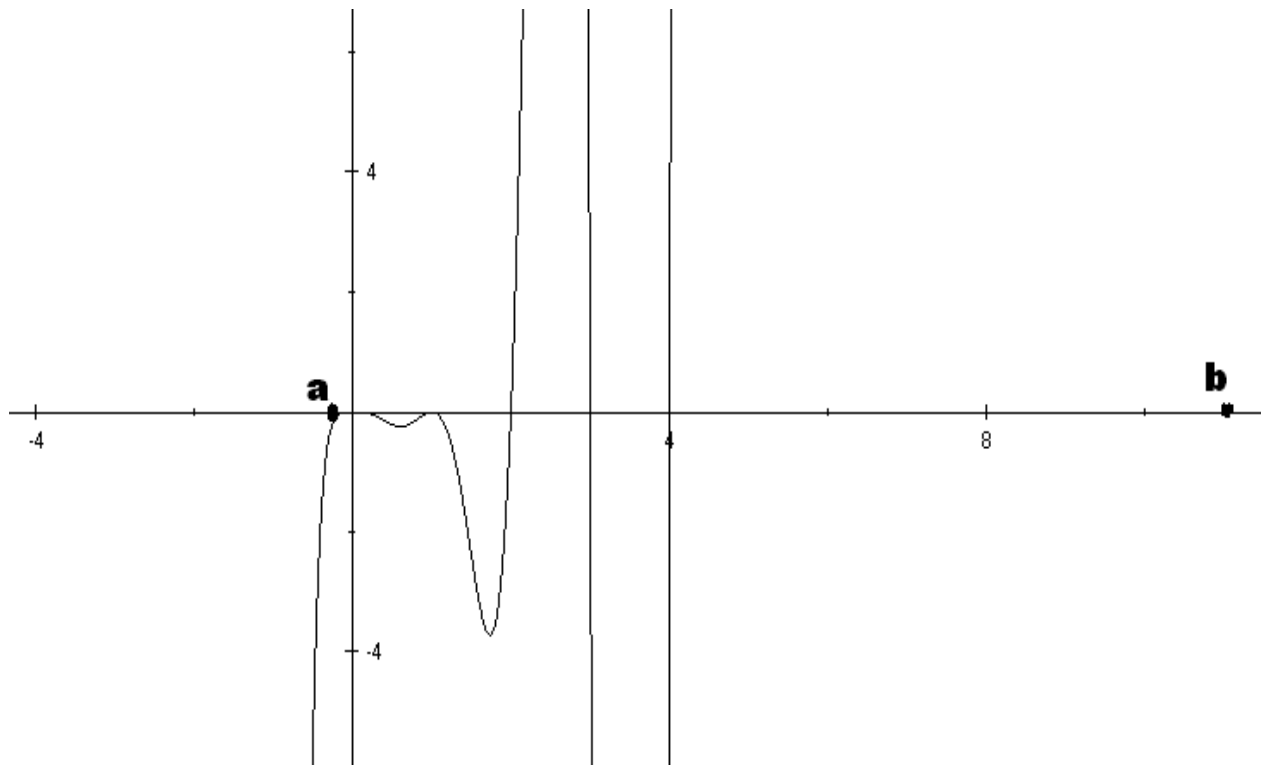
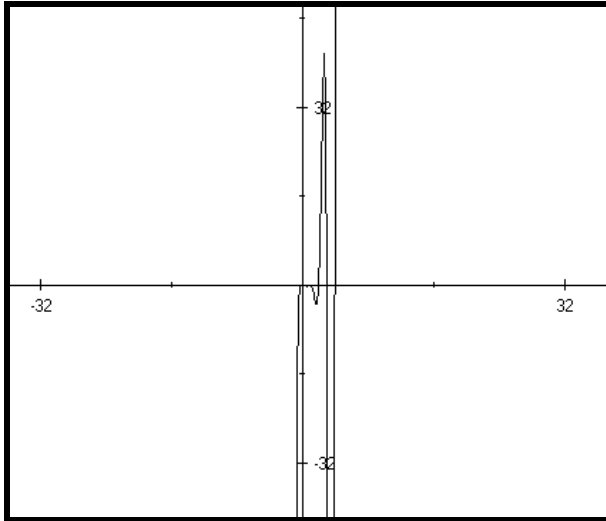
- $F_3(x) = x^4 - 5x^2 + 1$
 $x_0 = \{ \text{brak} \}$
 $a = -0.1$
 $b = 0.1$



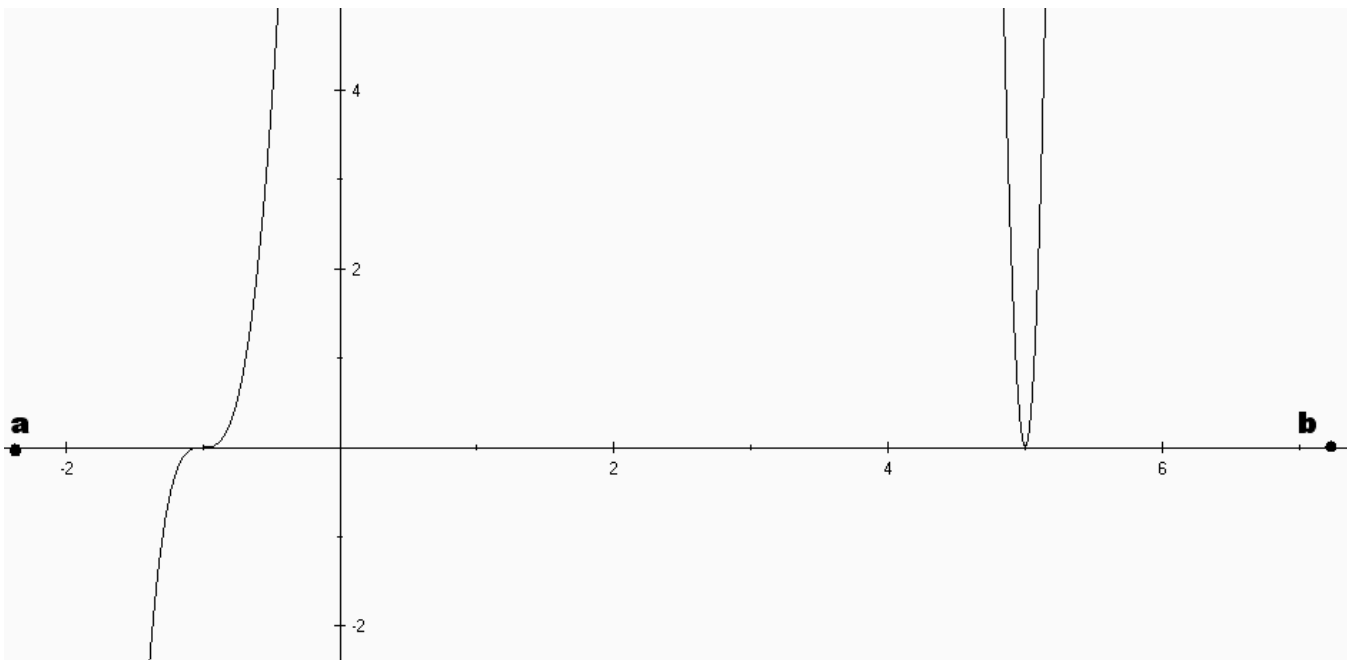
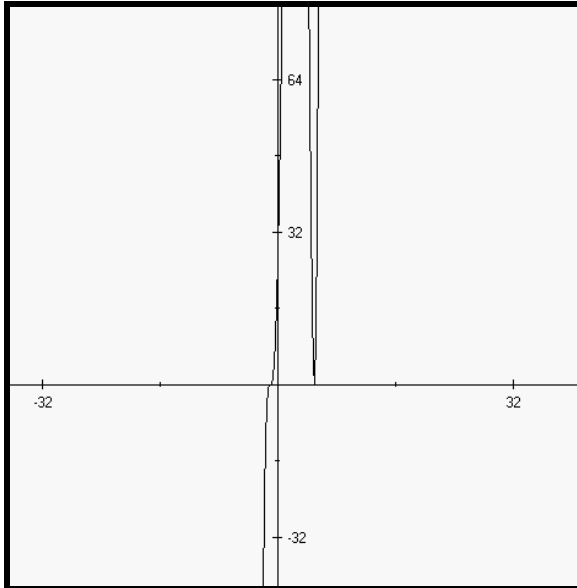
- $F_4(x) = x^8 - 17x^4 + 16$
 $x_0 = \{-2; -1; 1; 2\}$
 $a = -2,1$
 $b = 2,1$



- $F_5(x) = x^9 - 11x^8 + 45x^7 - 85x^6 + 74x^5 - 24x^4$
 $x_0 = \{0; 0; 0; 0; 1; 2; 3; 4\}$
 $a = -0,1$
 $b = 11$



- $F_6(x) = x^5 - 7x^4 - 2x^3 + 46x^2 + 65x + 25$
 $x_0 = \{-1; -1; -1; 5; 5\}$
 $a = -2,4$
 $b = 7,3$



Zadanie 3

```
#include <iostream>
#include <math.h>
#include <conio.h>
#include <time.h>
using namespace std;

float silnia(int x)
{
    float s=1;
    int i;
    for(i=1;i<=x;i++)
    {
        s=s*i;
    }
    return s;
}

int main()
{
    int n,i,k,m,s;
    float x,x0,cosx1,cosx2,cosxe,wyn,eps,il,il1;

    cout<<"ten program wyznacza (cos(x/5))^2 "<<endl<<"x0= "; cin>>x0;
    cout<<"z jaka dokladnoscia liczyc ? >>> "; cin>>eps;

//metoda 1, bez rekurencji
    cout<<endl<<endl<<"Metoda I (bez rekurencji) : "<<endl<<endl;
    i=0;
    cosx1=0;
    cosx2=0;
    do
    {
        cosx2=cosx1;
        cosx1=cosx1+pow((-1),i)*pow(((2*x0)/5),(2*i))/silnia(2*i);
        i++;
        cosxe=cosx2-cosx1;
        if (cosxe<0) cosxe=-cosxe;
    }while(cosxe>eps);

    wyn=(1+cosx1)/2.0;
    cout<<endl<<"(cos(x/5))^2 dla x0 = "<<x0<<" wynosi "<<wyn<<" (wykonano w
"<<i+1<<" krokow) ";
```

```

//metoda 2, rekurencyjnie
cout<<endl<<endl<<"Metoda II (rekurencyjnie) : "<<endl<<endl;
i=1;
cosx1=1;
cosx2=0;
wyn=0;
m=1;
il=(-4*x0*x0/25.0)/2.0;
ill=il;
do
{
  cosx2=cosx1;
  cosx1=cosx1+ill;
  m=m+2;
  s=m+1;
  ill=ill*(il)/float(m*s);
  i++;
  cosxe=cosx2-cosx1;
  if (cosxe<0) cosxe=-cosxe;
}while(cosxe>eps);
wyn=(1+cosx1)/2.0;
cout<<endl<<"(cos(x/5))^2 dla x0 = "<<x0<<" wynosi "<<wyn<<" (wykonano w
"<<i+1<<" krokow) ";

//system("PAUSE");
//return 0;
getch();
}

```

Dokładność	ARGUMENT					
	x ₀ = -3	x ₀ = -1	x ₀ = 0	x ₀ = 1	x ₀ = 7	x ₀ = 15
0.1	4 / 0.6 / 0.6	3 / 0.9 / 0.9	3 / 1 / 1	3 / 0.9 / 0.9	6 / 0 / -0.3	11 / 0.9 / -0.4
0.01	5 0.68 0.66	4 0.96 0.96	3 / 1 / 1	4 0.96 0.96	7 0.02 -0.39	12 0.98 -0.45
0.001	6 0.681 0.661	5 0.960 0.960	3 / 1 / 1	5 0.960 0.960	8 0.028 -0.397	13 0.980 -0.452
0.0001	7 0.6811 0.6610	5 0.9605 0.9602	3 / 1 / 1	5 0.9605 0.9602	9 0.0288 -0.3977	14 0.9800 -0.4526
0.00001	7 0.68117 0.66108	5 0.96053 0.96026	3 / 1 / 1	5 0.96053 0.96026	10 0.02888 -0.39778	14 0.98008 -0.45266
0.000001	7 0.68117 0.66108	6 0.960531 0.960266	3 / 1 / 1	6 0.960531 0.960266	10 0.028888 -0.397786	15 0.980085 -0.452662

