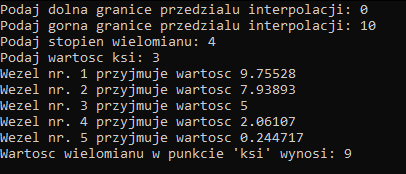
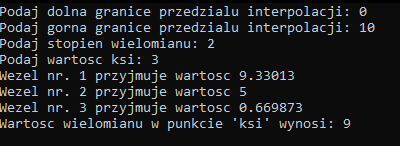
Zad 2

Wartości funkcji dla f(x) = x^2, ξ = 3, a=0, b=10 (funkcja kwadratowa została wybrana aby szybko można było obliczyć jej wartość punkcie i porównać z tą otrzymaną z programu)

1. Dla n=4



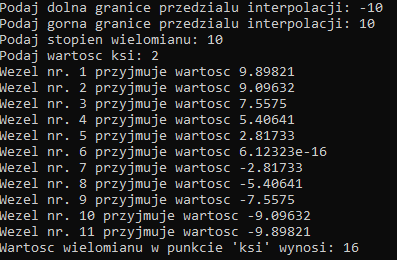
1. Dla n=2



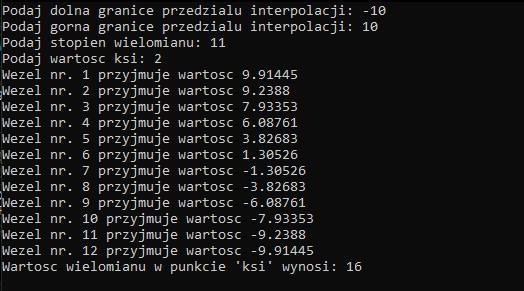
dla f(x) = x^4, a=-10, b=-10, ksi = 2,

wartości węzłów wynoszą:

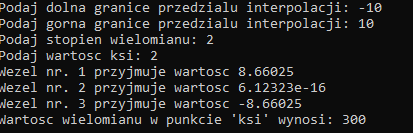
1. dla n=10



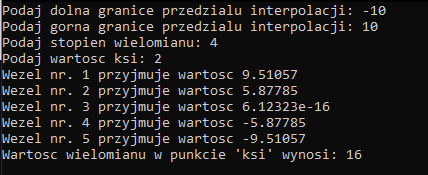
1. dla n=11



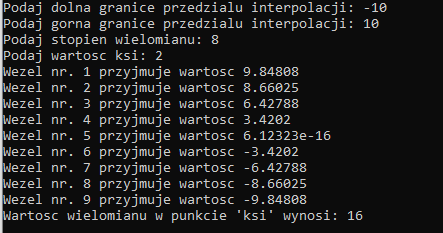
1. n=2



1. n=4

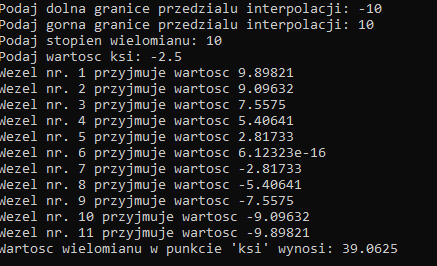


1. n=8

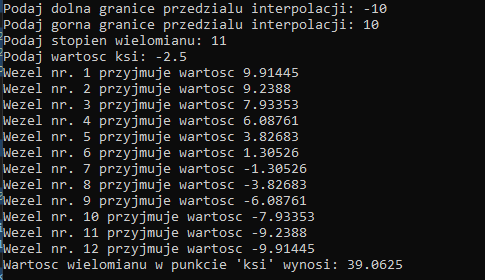


dla f(x) = x^4, a=-10, b=-10, ksi = -2 .5

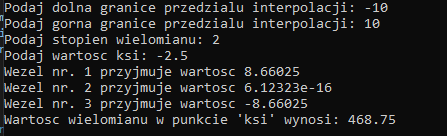
1. dla n=10



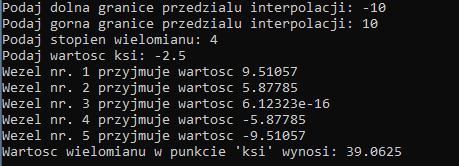
1. dla n=11



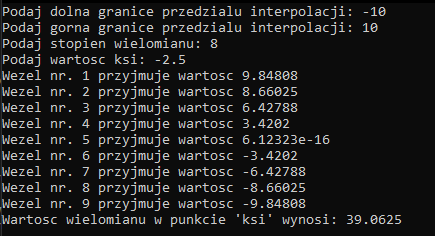
1. dla n=2



1. dla n=4



1. dla n=8



Kod:

//Zad. 2.

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#include <iostream>

using namespace std;

int main(void)

{

double a, b, n, e, pomoc, wynik = 0, wynik1 = 1; //wprowadzenie danych wejsciowych

cout << "Podaj dolna granice przedzialu interpolacji: ";

cin >> a;

cout << "Podaj gorna granice przedzialu interpolacji: ";

cin >> b;

cout << "Podaj stopien wielomianu: ";

cin >> n;

cout << "Podaj wartosc ksi: ";

cin >> e;

double krok;

krok = (b - a) / n; //wyznaczenie kroku potrzebnego do wyznaczenia nowych wezlow

double\* x, \* wartosc;

x = new double[n + 1]; // wprowadzenie tablicy dla kolejnych wezlow

wartosc = new double[n + 1]; // wprowadzenie tablicy wartosci funkcji

for (int i = 0; i <= n; i++) //metoda Czybyszewa

{

pomoc = cos((2.0 \* i + 1.0) / (2.0 \* n + 2.0) \* 3.14159265358979323846);

x[i] = 1.0 / 2.0 \* ((b - a) \* pomoc + (a + b));

wartosc[i] = pow(x[i], 4); // wartosc zadanej funkcji

}

for (int i = 0; i <= n; i++)//obliczenie wartosci zadanego wielomianu

{

wynik1= 1;

for (int j = 0; j <= n; j++)

{

if (j != i)

wynik1 = wynik1 \* (e - x[j]) / (x[i] - x[j]);

}

wynik1 = wynik1 \* wartosc[i];

wynik = wynik + wynik1;

}

for (int i = 0; i <= n; i++) //wypisywanie kolejnych wezlow

{

cout << "Wezel nr. " << i + 1 << " przyjmuje wartosc " << x[i] << endl;

}

cout << "Wartosc wielomianu w punkcie 'ksi' wynosi: " << wynik<< endl;

return 0;

}