Zadanie 10:

8. Znaleźć wartości funkcji
$$f(x) = \frac{1}{1+5x^2} \tag{7}$$
w punktach $-1,\ -1+\frac{1}{32},\ -1+\frac{2}{32},\ \ldots,\ 1-\frac{1}{32},\ 1$, a następnie skonstruować wielomian interpolacyjny Lagrange'a oparty na tych węzłach i wartościach funkcji (7) w tych węzłach. Narysować wykres wielomianu interpolacyjnego.

9 O. Skonstruować naturalny splajn kubiczny dla funkcji i węzłów z zadania 8. Sporządzić jego wykres.

10. Skonstruować interpolację funkcjami wymiernymi według algorytmu Floatera i Hormanna z parametrem $d = 3$ dla funkcji i węzłów z zadania 8. Sporządzić odpowiedni wykres.

Kod w języku C++:

```cpp
#include<iostream>
#include<array> //std::array
#include<math.h> //std::pow
#include<fstream> //std::ofstream

using std::endl;
using std::cout;

#define Vector std::array<long double,65>

void fillXY(Vector& x, Vector& y);
long double newtonSymbol(int x, int y);
void calculateW(Vector& w, const int d);
void calculateResult(const Vector& x, const Vector& y, const Vector& w);
long double silnia (int x);
void writeToFile(const Vector& x, const Vector& y, const char* f);


int main(int argc, char const *argv[]) {
    if(argc!=2){
        cout<<"Argumenty wywolania programu : <parametr d>"<<endl;
        exit(-1);
    }
    int d=std::stoi(argv[1]);
    Vector x; //x-y funkcji zmieniajace sie o 1/32
    Vector y ; //wartosci funkcji 1/(1+5*x*x)
    Vector w = {}; //wektor wag
    //najpierw zajmiemy sie funkcja interpolowana
    fillXY(x,y);
    writeToFile(x,y,"function.txt");
    //teraz wielomian interpolacyjny
    calculateW(w,d);
    calculateResult(x,y,w);
    return 0;
}
```

```
//wypelnia wektory wartosciami x i wartosciami funkcji odpowiednio
void fillXY(Vector& x, Vector& y){
    int i=0;
    for(long double j=-1; j<=1; j=j+1.0/32){
        x[i]=j;
        y[i]=(1/(1+5*j*j));
        i++;
    }
}



//zapisuje wektor x i wartosci funkcji y do pliku f
void writeToFile(const Vector& x, const Vector& y, const char* f){
    std::ofstream file;
    file.open(f);
    for(int i=0; i<x.size(); i++){
        file<<x[i]<<" "<<y[i]<<endl;
    }
    file.close();
}

//liczy wagi do algorytmu
void calculateW(Vector& w, const int d){
    long double sum;
    for(int k=0; k<w.size(); k++){
        w[k]=std::pow(-1,k-d);
        sum=0;
        for(int i=k-d; i<=k; i++){
            if(i<0) continue;
            if(i==w.size()-d) break;
            sum+=newtonSymbol(d,k-i);
        }
        w[k]=w[k]*sum;
    }
}


//oblicza symbol newtona
long double newtonSymbol(int x, int y){
    return silnia(x)/(silnia(y)*silnia(x-y));
}

//oblicza silnie
long double silnia (int x){
    if(x==0) return 1;
    if (x<2) return x;
    return x*silnia(x-1);
}


//oblicza wartosci y i zapisuje je do pliku
void calculateResult(const Vector& x, const Vector& y, const Vector& w){
    long double upperSum;
    long double lowerSum;
    long double temp;
```

```
    std::ofstream file;
    file.open("flohorman.txt");
    for(long double i=-1; i<=1; i=i+0.01){
        upperSum=0;
        lowerSum=0;
        temp=0;
        for(int k=0; k<w.size(); k++){
            temp=w[k]/(i-x[k]);
            upperSum+=temp*y[k];
            lowerSum+=temp;
        }
        file<<i<<" "<<upperSum/lowerSum<<endl;
    }
    file.close();
}
```

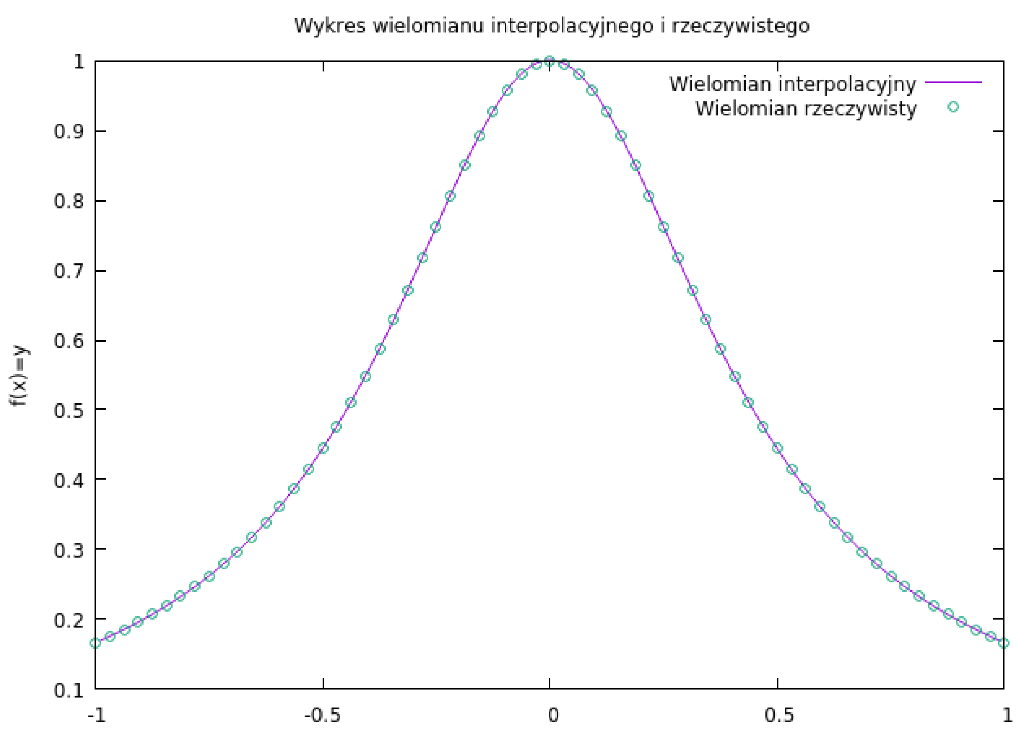
Skrypt gunplota do stworzenia wykresu:

```
set title "Wykres wielomianu interpolacyjnego i rzeczywistego"
set xlabel "x"
set ylabel "f(x)=y"
set xrange [-1: 1]
plot 'flohorman.txt' with lines title 'Wielomian interpolacyjny' , 'function.txt' title 'Wielomian rzeczywisty' with points pointtype 6
```

Prawidłowe przykładowe wywołanie programu:

```
Zadanie10]$ g++ -o zadanie10.x zadanie10.cpp
Zadanie10]$ ./zadanie10.x 3
Zadanie10]$
```

Wyniki:



Opis metody:

Algorytm Floatera-Hormanna jest typem interpolacji za pomocą funkcji wymiernych. Oferuje taką samą dokładność jak interpolacje splajnami, ale jest mniej złożony numerycznie. Aby obliczyć wartość wielomianu interpolacyjnego w punkcie x korzystam z poniższych wzorów:

- Do obliczania wartości funkcji wzór w postaci barycentrycznej (w funkcji calculateResult()):

$$r(x) = \frac{\sum_{k=0}^{n} fk \frac{w_k}{x-w_k}}{\sum_{k=0}^{n} \frac{w_k}{x-x_k}}$$

- Do obliczenia wag dla równoodległych węzłów (w funkcji calculateW()):

$$w_k = (-1)^{k-d} \sum_{i \in J_k} \binom{d}{k-i}$$

Z warunkiem dla zbioru Jk:

$$\begin{aligned} J_k = \{i \in I : k - d \leq i \leq k\} \\ zograniczeniem : \\ I = \{0, 1, ..., n - d\} \end{aligned}$$

Jak możemy zauważyć wielomian interpolacyjny praktycznie pokrywa się z funkcją interpolowaną. Algorytm Floatera-Hormanna daje szybkie i dokładne wyniki.