

Mateusz Sławiński 222B

Sprawozdanie - lab1

1) Napisz procedurę/funkcję, która obliczy wyróżnik i wyznaczy miejsca zerowe zadanej funkcji kwadratowej:

$x(t) = \hat{A}t^2 + \hat{B}t + \hat{C}$. Wykonaj wykres tej funkcji dla $t \in \langle -10; 10 \rangle$, gdzie $x, t \in \mathbb{R}$, przy $\Delta_t = \frac{1}{100}$.

```
int a = 9, b = 9, c = 4;
float poczatek1 = -10.0;
float krok1 = 0.01;

double delta()
{
    float wynik = (b*b)-(4*(a*c));
    return wynik;
}
void pierwiastki()
{
    double x1, x2;

    if (delta() < 0)
    {
        cout<<"nie ma pierwiastkow"<<endl;
        return;
    }
    else if (delta() > 0)
    {
        x1 = (-b+sqrt(delta()))/(2*a);
        x2 = (-b-sqrt(delta()))/(2*a);
        cout << "pierwiastki= " << x1 << ", " << x2 << endl;
        return;
    }
    else
    {
        x1 = x2 = (-b)/(2*a);
        cout << "pierwiastek= " << x1 << endl;
        return;
    }
}
```

```

}

float wartosc(float t)
{
    float x = a * (t*t) + b * t + c;
    return x;
}

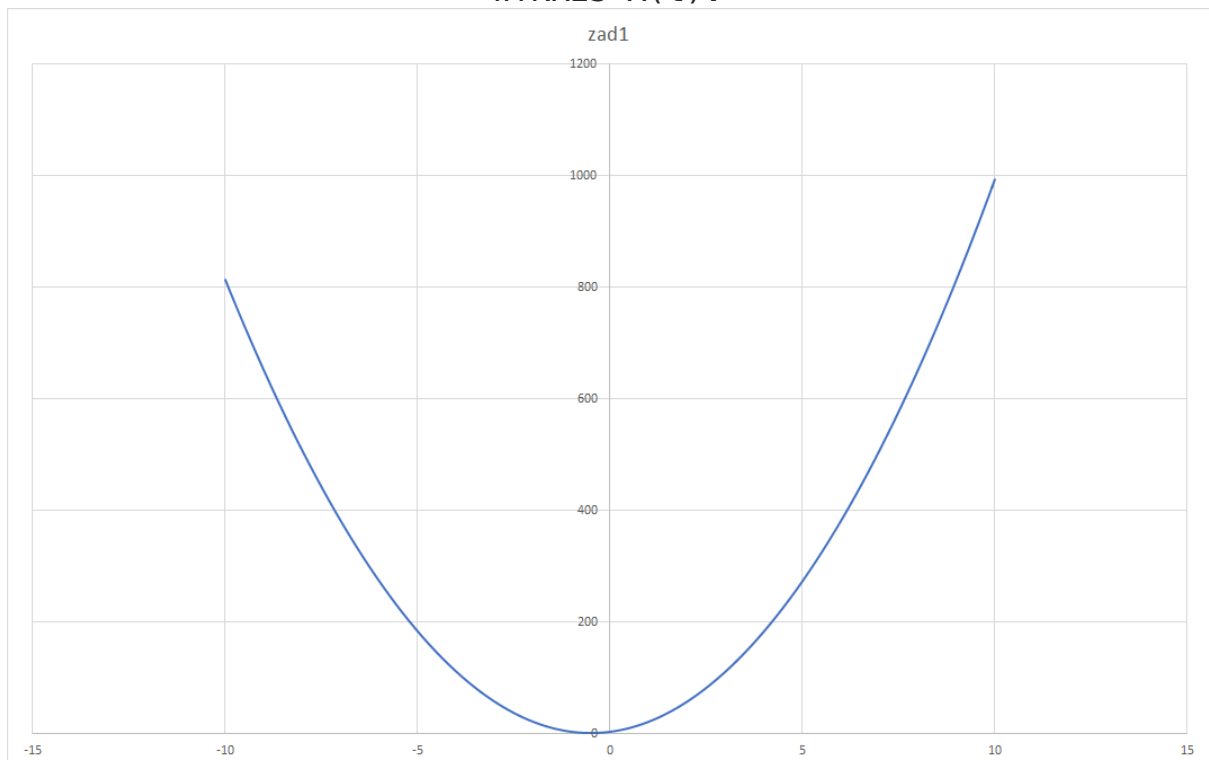
int main(int argc, const char * argv[])
{
    ofstream plik;

    double x = delta();
    cout << "delta= " << x << endl;
    pierwiastki();
    int n = 0;
    float t = poczatek1 + (n * krok1);

    plik.open("punkty1.txt");
    while (t <= 10)
    {
        plik << t << " " << wartosc(t) << "\n";
        n++;
        t = poczatek1 + (n * krok1);
    }
    plik.close();
}

```

WYKRES $X(t)$:



2) Napisz program obliczający poniżej zadane funkcje dla $t \in \langle 0; 1 \rangle$, gdzie $\Delta_t = \frac{1}{22050}$. Wykonaj wykresy tych funkcji.

$$y(t) = 2 \cdot x(t)^2 + 12 \cdot \cos(t)$$

$$z(t) = \sin(2\pi \cdot 7 \cdot t) \cdot x(t) - 0.2 \cdot \log_{10}(|y(t)| + \pi)$$

$$u(t) = \sqrt{|y(t) \cdot y(t) \cdot z(t)|} - 1.8 \cdot \sin(0.4 \cdot t \cdot z(t) \cdot x(t))$$

$$v(t) = \begin{cases} (1 - 7t) \cdot \sin(\frac{2\pi \cdot t \cdot 10}{t+0.04}) & \text{dla } 0.22 > t \geq 0 \\ 0.63 \cdot t \cdot \sin(125 \cdot t) & \text{dla } 0.22 \leq t < 0.7 \\ t^{-0.662} + 0.77 \sin(8t) & \text{dla } 1.0 \geq t \geq 0.7 \end{cases}$$

$$p(t) = \sum_{n=1}^N \frac{\cos(12t \cdot n^2) + \cos(16t \cdot n)}{n^2}$$

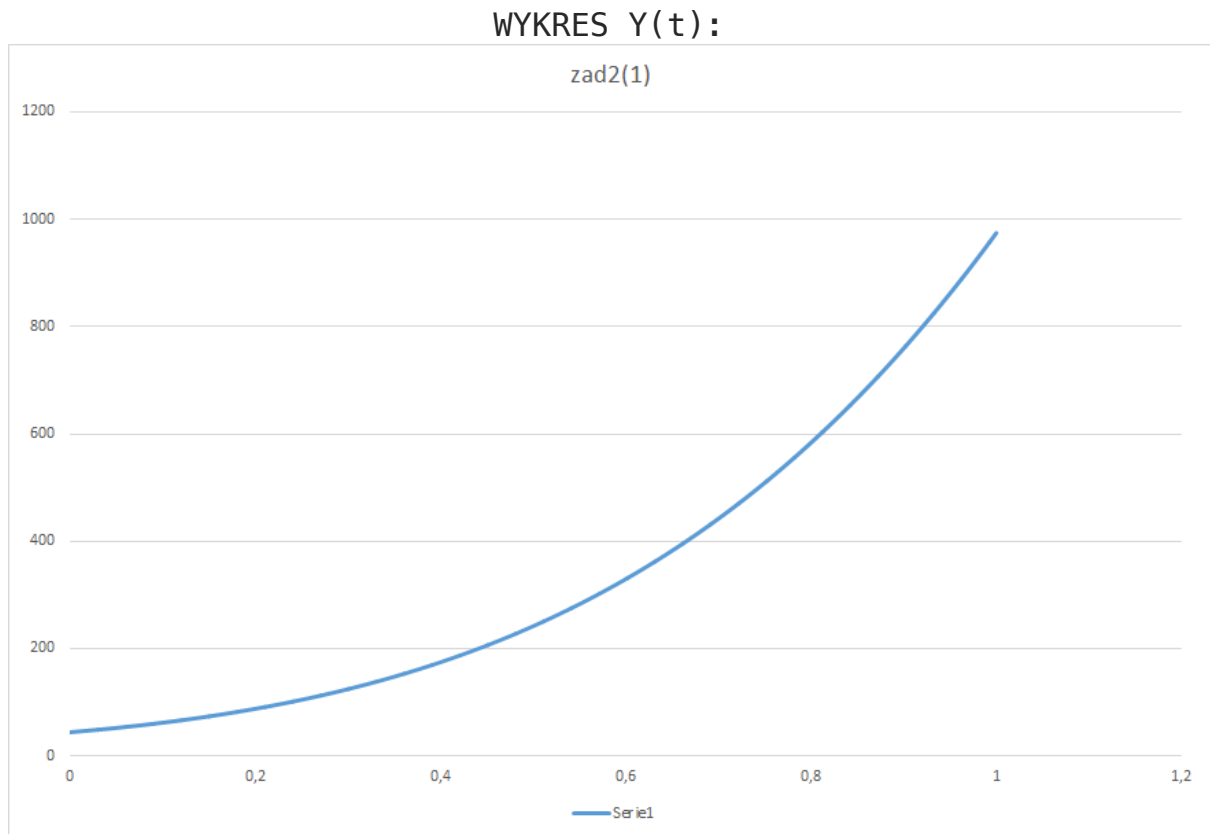
$$\text{dla } N \in \{2, 4, \widehat{A}\widehat{B}\}$$

```
float pi = 3.141592;
float poczatek2 = 0.0;
float krok2 = 0.0000453;

float Y(float t)
{
    float x = wartosc(t);
    float y = 2 * (x*x) + 12 * cos(t);

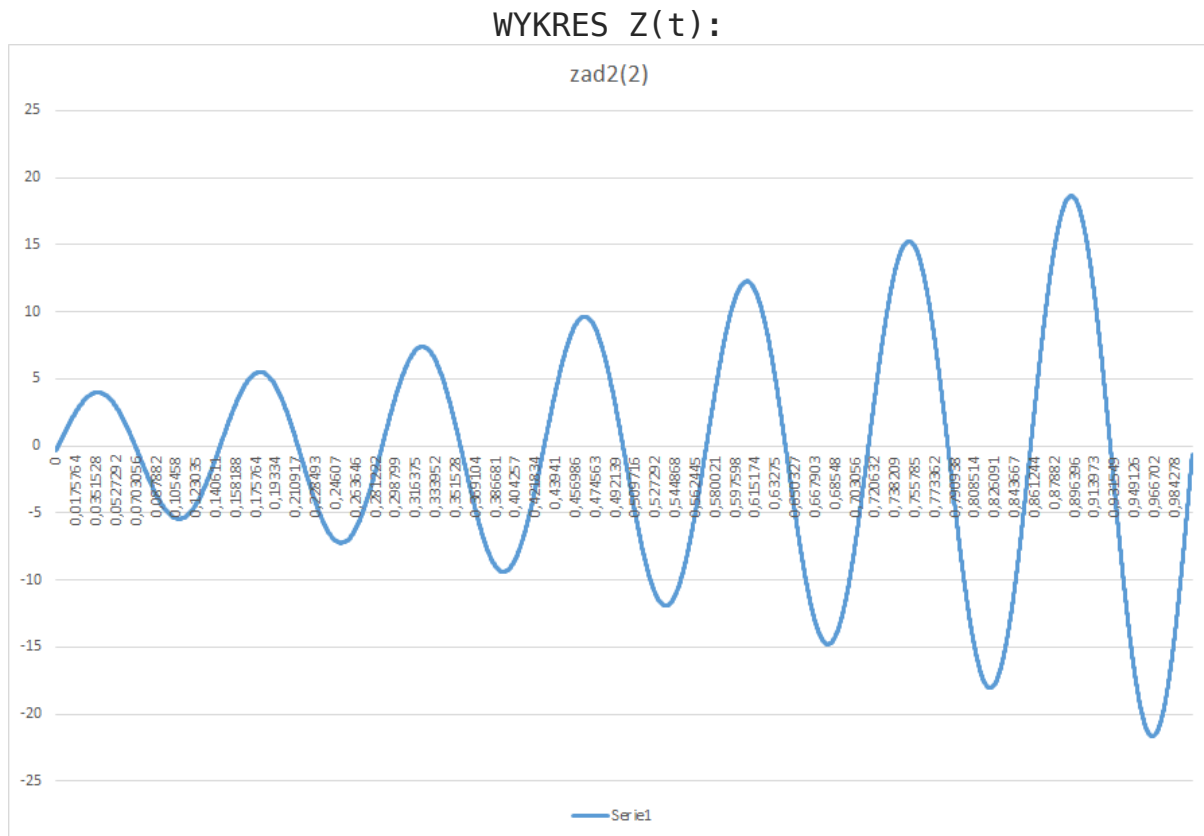
    return y;
}

int main(int argc, const char * argv[])
{
    ofstream plik;
    n = 0;
    t = poczatek2 + (n * krok2);
    plik.open("punkty2.txt");
    while (t <= 1)
    {
        plik << t << " " << Y(t) << "\n";
        n++;
        t = poczatek2 + (n * krok2);
    }
    plik.close();
}
```



```
float Z(float t)
{
    float x = wartosc(t);
    float y = Y(t);
    float z = sin(2 * pi * 7 * t) * x - (0.2*log10(abs(y) +
pi));
    return z;
}

int main(int argc, const char * argv[])
{
    ofstream plik;
    n = 0;
    t = poczatek2 + (n * krok2);
    plik.open("punkty3.txt");
    while (t <= 1)
    {
        plik << t << " " << Z(t) << "\n";
        n++;
        t = poczatek2 + (n * krok2);
    }
    plik.close();
}
```



```
float U(float t)
{
    float x = wartosc(t);
    float y = Y(t);
    float z = Z(t);
    float u = sqrt(abs(y * y * z)) - (1.8 * sin(0.4 * t * z *
x));
    return u;
}

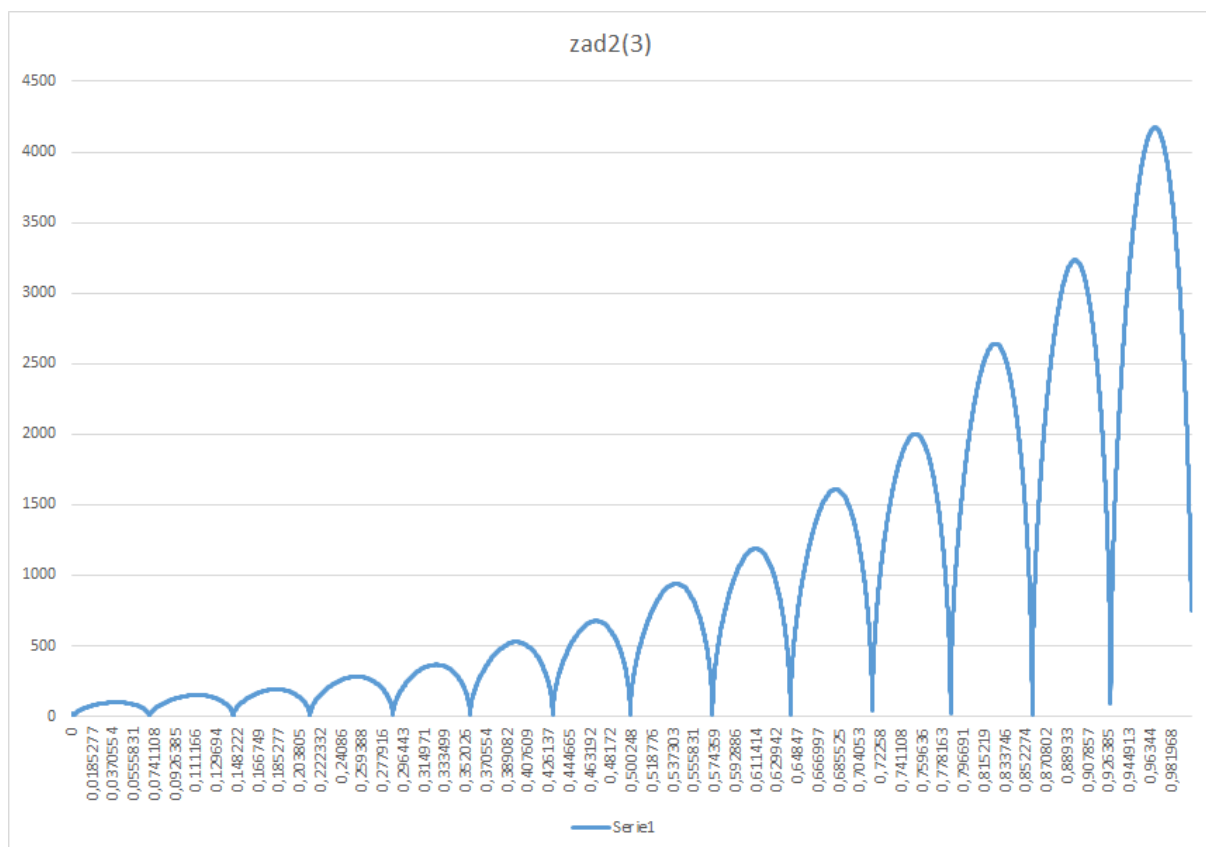
int main(int argc, const char * argv[])
{
    ofstream plik;
    n = 0;
    t = poczatek2 + (n * krok2);
    plik.open("punkty4.txt");
    while (t <= 1)
    {
        plik << t << "\t" << U(t) << "\n";
        n++;
        t = poczatek2 + (n * krok2);
    }
    plik.close();
}
```

```

int main(int argc, const char * argv[])
{
    ofstream plik;
    n = 0;
    t = poczatek2 + (n * krok2);
    plik.open("punkty4.txt");
    while (t <= 1)
    {
        plik << t << "\t" << U(t) << "\n";
        n++;
        t = poczatek2 + (n * krok2);
    }
    plik.close();
}

```

WYKRES U(t):



```

float V(float t)
{
    if (0.22 > t && t >= 0)
    {
        return (1 - (7 * t)) * sin((2 * pi*t * 10) / (t +
0.04));
    }
}

```

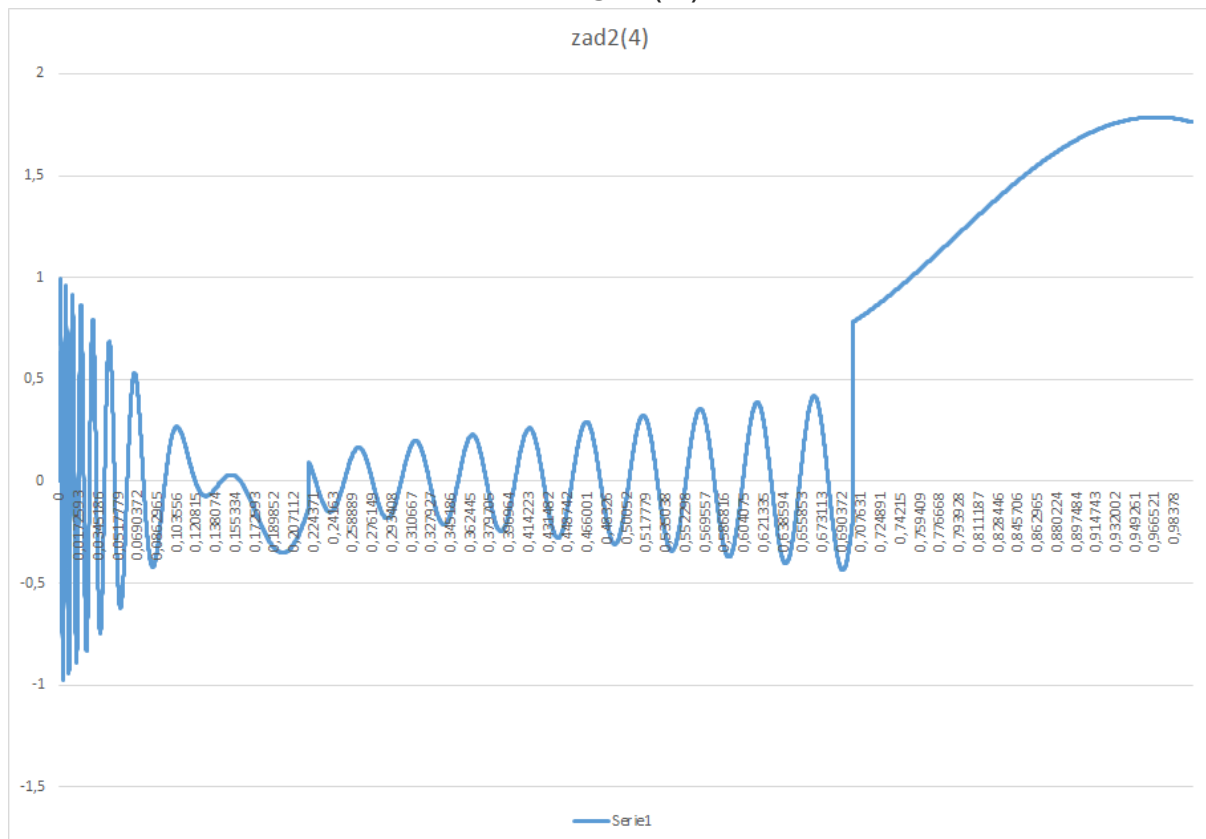
```

else if (0.22 <= t && t < 0.7)
{
    return 0.63 * t * sin(125 * t);
}
else
{
    return pow(t, -0.662) + 0.77 * sin(8 * t);
}
}

int main(int argc, const char * argv[])
{
    ofstream plik;
    n = 0;
    t = poczatek2 + (n * krok2);
    plik.open("punkty5.txt");
    while (t <= 1)
    {
        plik << t << "\t" << V(t) << "\n";
        n++;
        t = poczatek2 + (n * krok2);
    }
    plik.close();
}

```

WYKRES V(t):

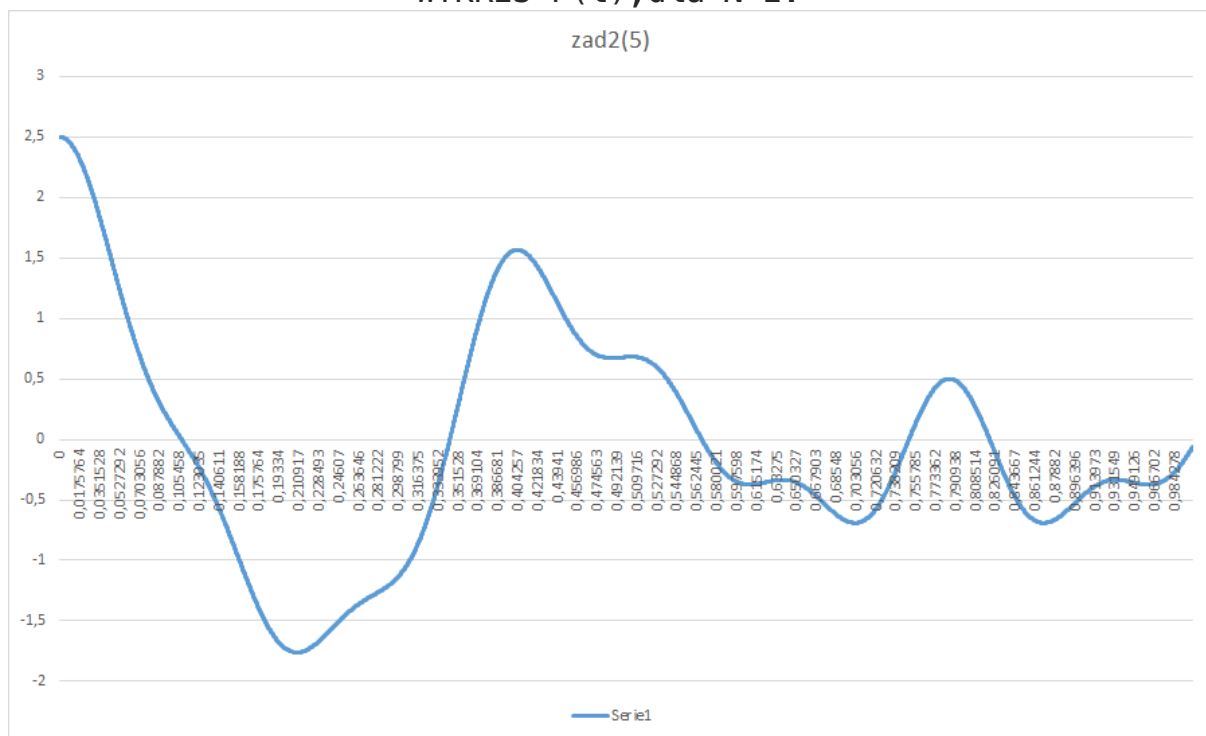


```

float P(float t, int N)
{
    float p = 0, suma = 0;
    for (int i = 1; i <= N; i++)
    {
        p = (cos(12 * t * i * i) + cos(16 * t * i)) / (i *
i);
        suma = suma + p;
    }
    return suma;
}
int main(int argc, const char * argv[])
{
    ofstream plik;
    n = 0;
    t = poczatek2 + (n * krok2);
    plik.open("punkty6.txt");
    while (t <= 1)
    {
        plik << t << "\t" << P(t, 2) << "\n";
        n++;
        t = poczatek2 + (n * krok2);
    }
    plik.close();
}

```

WYKRES P(t), dla N=2:

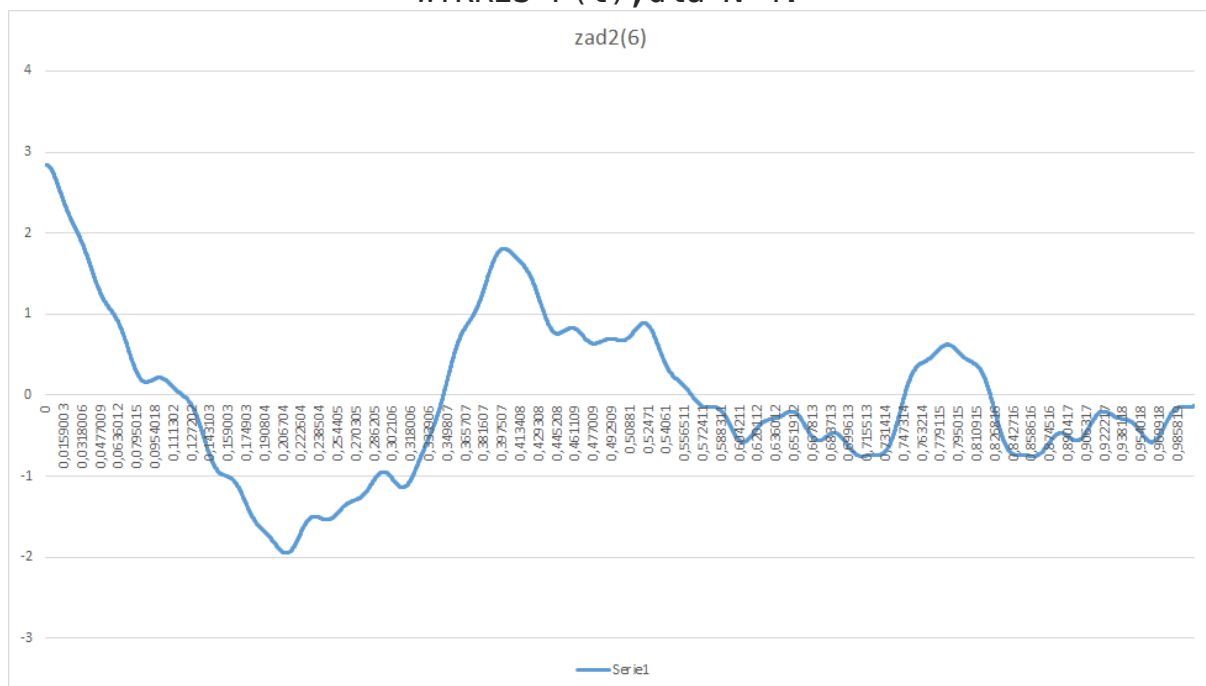



```

int main(int argc, const char * argv[])
{
    ofstream plik;
    n = 0;
    t = poczatek2 + (n * krok2);
    plik.open("punkty7.txt");
    while (t <= 1)
    {
        plik << t << "\t" << P(t, 4) << "\n";
        n++;
        t = poczatek2 + (n * krok2);
    }
    plik.close();
}

```

WYKRES P(t), dla N=4:



```

int main(int argc, const char * argv[])
{
    ofstream plik;
    n = 0;
    t = poczatek2 + (n * krok2);
    plik.open("punkty8.txt");
    while (t <= 1)
    {
        plik << t << "\t" << P(t, 99) << "\n";
        n++;
        t = poczatek2 + (n * krok2);
    }
    plik.close();
}

```

WYKRES $P(t)$, dla $N=99$:

