

**Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

Лабораторна робота №1
з дисципліни
«Алгоритми і структури даних»

Виконав:

студент групи ІМ-42
Федоренко Іван Русланович
номер у списку групи: 29

Перевірив:

Сергієнко А. М.

Київ 2025

Завдання

Дане натуральне число n . Знайти суму перших n членів ряду чисел, заданого рекурентною формулою. Розв'язати задачу трьома способами:

- 1) у програмі використати рекурсивну функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному спуску;
- 2) у програмі використати рекурсивну функцію, яка виконує обчислення і членів ряду, і суми на рекурсивному поверненні;
- 3) у програмі використати рекурсивну функцію, яка виконує обчислення членів ряду на рекурсивному спуску, а обчислення суми на рекурсивному поверненні

Варіант № 29

$$F_1 = x/(0.525 + 0.5x)^2 - 1; \quad F_{i+1} = F_i \cdot F_1(3 - 2i)/(2i), \quad i > 0;$$
$$\sum_{i=1}^n F_i = \sqrt{x}, \quad 0.5 < x < 1.$$

На жаль, формула не працює коректно для апроксимації функції \sqrt{x} , тому для обчислень я використав ряд Тейлора:

$$\sqrt{x} \approx 1 + \left(\frac{1}{2}\right)(x - 1) - \left(\frac{1}{8}\right)(x - 1)^2 + \left(\frac{1}{16}\right)(x - 1)^3 - \left(\frac{5}{128}\right)(x - 1)^4 + \dots$$
$$\sqrt{x} \approx \sum_{n=0}^N f_n, \text{ де } f_0 = 1, f_n = f_{n-1} \cdot \frac{3 - 2n}{2n} \cdot (x - 1)$$

Обчислення \sqrt{x} за допомогою біноміального ряду навколо точки $x=1$.

Код програми:

```
#include <stdio.h>
#include <math.h>

double sqrt_series_descent(double x, int n, int i, double term, double sum) {
    sum += term;
    printf("f_%d = %.10lf, sum = %.10lf\n", i, term, sum);

    if (i == n - 1) {
        return sum;
    }

    double next_term = term * ((3.0 - 2.0 * (i + 1)) / (2.0 * (i + 1))) * (x - 1);
    return sqrt_series_descent(x, n, i + 1, next_term, sum);
}

typedef struct {
```

```

    double term;
    double sum;
} Result;

Result sqrt_series_return(double x, int n, int i) {
    if (i == 0) {
        printf("f_0 = %.10lf\n", 1.0);
        return (Result){1.0, 1.0};
    }

    Result prev = sqrt_series_return(x, n, i - 1);
    double t = x - 1;
    double term = prev.term * ((3.0 - 2.0 * i) / (2.0 * i)) * t;
    double sum = prev.sum + term;

    printf("f_%d = %.10lf, sum = %.10lf\n", i, term, sum);
    return (Result){term, sum};
}

double sqrt_series_hybrid(double x, int n, int i, double prev_term) {
    double term = (i == 0) ? 1.0 :
        prev_term * ((3.0 - 2.0 * i) / (2.0 * i)) * (x - 1);

    if (i == n - 1) {
        printf("f_%d = %.10lf, sum = %.10lf\n", i, term, term);
        return term;
    }

    double rest_sum = sqrt_series_hybrid(x, n, i + 1, term);
    double total = term + rest_sum;

    printf("f_%d = %.10lf, sum = %.10lf\n", i, term, total);
    return total;
}

double sqrt_series_loop(int n, double x) {
    double sum = 1.0; // f_0 = 1
    double term = 1.0;

    for (int i = 1; i < n; i++) {
        term *= ((3.0 - 2.0 * i) / (2.0 * i)) * (x - 1);
        sum += term;
        printf("f_%d = %.10lf, sum = %.10lf\n", i, term, sum);
    }

    return sum;
}

int main() {

```

```

int n = 5;
double x = 0.79;
if(x > 0.5 && x < 1) {
    printf("\nN = %d, X = %.3lf.", n, x);
    printf("\n--- [1] Method1 ---\n");
    double res1 = sqrt_series_descent(x, n, 0, 1.0, 0.0);
    printf("Final sum = %.10lf\n", res1);

    printf("\n--- [2] Method2 ---\n");
    Result res2 = sqrt_series_return(x, n, n - 1);
    printf("Final sum = %.10lf\n", res2.sum);

    printf("\n--- [3] Method3 ---\n");
    double res3 = sqrt_series_hybrid(x, n, 0, 0.0);
    printf("Final sum = %.10lf\n", res3);

    printf("\n--- [4] Method4 ---\n");
    double res4 = sqrt_series_loop(n, x);
    printf("Final sum = %.10lf\n", res4);

    double actual = sqrt(x);
    printf("\nActual sqrt(%.2lf) = %.10lf\n", x, actual);
    printf("\nError: %.10lf", fabs(res1 - actual));
} else {
    printf("\nmInvalid data: %d, %.3lf", n, x);
}

return 0;
}

```

Результати тестування:

```
N = 5, X = 0.520.
--- [1] Method1 ---
f_0 = 1.0000000000, sum = 1.0000000000
f_1 = -0.2400000000, sum = 0.7600000000
f_2 = -0.0288000000, sum = 0.7312000000
f_3 = -0.0069120000, sum = 0.7242880000
f_4 = -0.0020736000, sum = 0.7222144000
Final sum = 0.7222144000

--- [2] Method2 ---
f_0 = 1.0000000000
f_1 = -0.2400000000, sum = 0.7600000000
f_2 = -0.0288000000, sum = 0.7312000000
f_3 = -0.0069120000, sum = 0.7242880000
f_4 = -0.0020736000, sum = 0.7222144000
Final sum = 0.7222144000

--- [3] Method3 ---
f_4 = -0.0020736000, sum = -0.0020736000
f_3 = -0.0069120000, sum = -0.0089856000
f_2 = -0.0288000000, sum = -0.0377856000
f_1 = -0.2400000000, sum = -0.2777856000
f_0 = 1.0000000000, sum = 0.7222144000
Final sum = 0.7222144000

--- [4] Method4 ---
f_1 = -0.2400000000, sum = 0.7600000000
f_2 = -0.0288000000, sum = 0.7312000000
f_3 = -0.0069120000, sum = 0.7242880000
f_4 = -0.0020736000, sum = 0.7222144000
Final sum = 0.7222144000

Actual sqrt(0.52) = 0.7211102551
Error: 0.0011041449
```

```
N = 5, X = 0.640.
--- [1] Method1 ---
f_0 = 1.0000000000, sum = 1.0000000000
f_1 = -0.1800000000, sum = 0.8200000000
f_2 = -0.0162000000, sum = 0.8038000000
f_3 = -0.0029160000, sum = 0.8008840000
f_4 = -0.0006561000, sum = 0.8002279000
Final sum = 0.8002279000

--- [2] Method2 ---
f_0 = 1.0000000000
f_1 = -0.1800000000, sum = 0.8200000000
f_2 = -0.0162000000, sum = 0.8038000000
f_3 = -0.0029160000, sum = 0.8008840000
f_4 = -0.0006561000, sum = 0.8002279000
Final sum = 0.8002279000

--- [3] Method3 ---
f_4 = -0.0006561000, sum = -0.0006561000
f_3 = -0.0029160000, sum = -0.0035721000
f_2 = -0.0162000000, sum = -0.0197721000
f_1 = -0.1800000000, sum = -0.1997721000
f_0 = 1.0000000000, sum = 0.8002279000
Final sum = 0.8002279000

--- [4] Method4 ---
f_1 = -0.1800000000, sum = 0.8200000000
f_2 = -0.0162000000, sum = 0.8038000000
f_3 = -0.0029160000, sum = 0.8008840000
f_4 = -0.0006561000, sum = 0.8002279000
Final sum = 0.8002279000

Actual sqrt(0.64) = 0.8000000000
Error: 0.0002279000
```

```

N = 5, X = 0.710.
--- [1] Method1 ---
f_0 = 1.0000000000, sum = 1.0000000000
f_1 = -0.1450000000, sum = 0.8550000000
f_2 = -0.0105125000, sum = 0.8444875000
f_3 = -0.0015243125, sum = 0.8429631875
f_4 = -0.0002762816, sum = 0.8426869059
Final sum = 0.8426869059

--- [2] Method2 ---
f_0 = 1.0000000000
f_1 = -0.1450000000, sum = 0.8550000000
f_2 = -0.0105125000, sum = 0.8444875000
f_3 = -0.0015243125, sum = 0.8429631875
f_4 = -0.0002762816, sum = 0.8426869059
Final sum = 0.8426869059

--- [3] Method3 ---
f_4 = -0.0002762816, sum = -0.0002762816
f_3 = -0.0015243125, sum = -0.0018005941
f_2 = -0.0105125000, sum = -0.0123130941
f_1 = -0.1450000000, sum = -0.1573130941
f_0 = 1.0000000000, sum = 0.8426869059
Final sum = 0.8426869059

--- [4] Method4 ---
f_1 = -0.1450000000, sum = 0.8550000000
f_2 = -0.0105125000, sum = 0.8444875000
f_3 = -0.0015243125, sum = 0.8429631875
f_4 = -0.0002762816, sum = 0.8426869059
Final sum = 0.8426869059

Actual sqrt(0.71) = 0.8426149773
Error: 0.0000719285

```

```

N = 5, X = 0.790.
--- [1] Method1 ---
f_0 = 1.0000000000, sum = 1.0000000000
f_1 = -0.1050000000, sum = 0.8950000000
f_2 = -0.0055125000, sum = 0.8894875000
f_3 = -0.0005788125, sum = 0.8889086875
f_4 = -0.0000759691, sum = 0.8888327184
Final sum = 0.8888327184

--- [2] Method2 ---
f_0 = 1.0000000000
f_1 = -0.1050000000, sum = 0.8950000000
f_2 = -0.0055125000, sum = 0.8894875000
f_3 = -0.0005788125, sum = 0.8889086875
f_4 = -0.0000759691, sum = 0.8888327184
Final sum = 0.8888327184

--- [3] Method3 ---
f_4 = -0.0000759691, sum = -0.0000759691
f_3 = -0.0005788125, sum = -0.0006547816
f_2 = -0.0055125000, sum = -0.0061672816
f_1 = -0.1050000000, sum = -0.1111672816
f_0 = 1.0000000000, sum = 0.8888327184
Final sum = 0.8888327184

--- [4] Method4 ---
f_1 = -0.1050000000, sum = 0.8950000000
f_2 = -0.0055125000, sum = 0.8894875000
f_3 = -0.0005788125, sum = 0.8889086875
f_4 = -0.0000759691, sum = 0.8888327184
Final sum = 0.8888327184

Actual sqrt(0.79) = 0.8888194417
Error: 0.0000132766

```

```

N = 5, X = 0.810.
--- [1] Method1 ---
f_0 = 1.0000000000, sum = 1.0000000000
f_1 = -0.0950000000, sum = 0.9050000000
f_2 = -0.0045125000, sum = 0.9004875000
f_3 = -0.0004286875, sum = 0.9000588125
f_4 = -0.0000509066, sum = 0.9000079059
Final sum = 0.9000079059

--- [2] Method2 ---
f_0 = 1.0000000000
f_1 = -0.0950000000, sum = 0.9050000000
f_2 = -0.0045125000, sum = 0.9004875000
f_3 = -0.0004286875, sum = 0.9000588125
f_4 = -0.0000509066, sum = 0.9000079059
Final sum = 0.9000079059

--- [3] Method3 ---
f_4 = -0.0000509066, sum = -0.0000509066
f_3 = -0.0004286875, sum = -0.0004795941
f_2 = -0.0045125000, sum = -0.0049920941
f_1 = -0.0950000000, sum = -0.0999920941
f_0 = 1.0000000000, sum = 0.9000079059
Final sum = 0.9000079059

--- [4] Method4 ---
f_1 = -0.0950000000, sum = 0.9050000000
f_2 = -0.0045125000, sum = 0.9004875000
f_3 = -0.0004286875, sum = 0.9000588125
f_4 = -0.0000509066, sum = 0.9000079059
Final sum = 0.9000079059

Actual sqrt(0.81) = 0.9000000000
Error: 0.0000079059

```

```

N = 5, X = 0.860.
--- [1] Method1 ---
f_0 = 1.0000000000, sum = 1.0000000000
f_1 = -0.0700000000, sum = 0.9300000000
f_2 = -0.0024500000, sum = 0.9275500000
f_3 = -0.0001715000, sum = 0.9273785000
f_4 = -0.0000150063, sum = 0.9273634938
Final sum = 0.9273634938

--- [2] Method2 ---
f_0 = 1.0000000000
f_1 = -0.0700000000, sum = 0.9300000000
f_2 = -0.0024500000, sum = 0.9275500000
f_3 = -0.0001715000, sum = 0.9273785000
f_4 = -0.0000150063, sum = 0.9273634938
Final sum = 0.9273634938

--- [3] Method3 ---
f_4 = -0.0000150063, sum = -0.0000150063
f_3 = -0.0001715000, sum = -0.0001865063
f_2 = -0.0024500000, sum = -0.0026365063
f_1 = -0.0700000000, sum = -0.0726365063
f_0 = 1.0000000000, sum = 0.9273634937
Final sum = 0.9273634937

--- [4] Method4 ---
f_1 = -0.0700000000, sum = 0.9300000000
f_2 = -0.0024500000, sum = 0.9275500000
f_3 = -0.0001715000, sum = 0.9273785000
f_4 = -0.0000150063, sum = 0.9273634938
Final sum = 0.9273634938

Actual sqrt(0.86) = 0.9273618495
Error: 0.0000016442

```

```

N = 5, X = 0.900.
--- [1] Method1 ---
f_0 = 1.0000000000, sum = 1.0000000000
f_1 = -0.0500000000, sum = 0.9500000000
f_2 = -0.0012500000, sum = 0.9487500000
f_3 = -0.0000625000, sum = 0.9486875000
f_4 = -0.0000039062, sum = 0.9486835937
Final sum = 0.9486835937

--- [2] Method2 ---
f_0 = 1.0000000000
f_1 = -0.0500000000, sum = 0.9500000000
f_2 = -0.0012500000, sum = 0.9487500000
f_3 = -0.0000625000, sum = 0.9486875000
f_4 = -0.0000039062, sum = 0.9486835937
Final sum = 0.9486835937

--- [3] Method3 ---
f_4 = -0.0000039062, sum = -0.0000039062
f_3 = -0.0000625000, sum = -0.0000664062
f_2 = -0.0012500000, sum = -0.0013164062
f_1 = -0.0500000000, sum = -0.0513164062
f_0 = 1.0000000000, sum = 0.9486835937
Final sum = 0.9486835937

--- [4] Method4 ---
f_1 = -0.0500000000, sum = 0.9500000000
f_2 = -0.0012500000, sum = 0.9487500000
f_3 = -0.0000625000, sum = 0.9486875000
f_4 = -0.0000039062, sum = 0.9486835937
Final sum = 0.9486835937

Actual sqrt(0.90) = 0.9486832981
Error: 0.0000002957

```

```

N = 5, X = 0.970.
--- [1] Method1 ---
f_0 = 1.0000000000, sum = 1.0000000000
f_1 = -0.0150000000, sum = 0.9850000000
f_2 = -0.0001125000, sum = 0.9848875000
f_3 = -0.0000016875, sum = 0.9848858125
f_4 = -0.0000000316, sum = 0.9848857809
Final sum = 0.9848857809

--- [2] Method2 ---
f_0 = 1.0000000000
f_1 = -0.0150000000, sum = 0.9850000000
f_2 = -0.0001125000, sum = 0.9848875000
f_3 = -0.0000016875, sum = 0.9848858125
f_4 = -0.0000000316, sum = 0.9848857809
Final sum = 0.9848857809

--- [3] Method3 ---
f_4 = -0.0000000316, sum = -0.0000000316
f_3 = -0.0000016875, sum = -0.0000017191
f_2 = -0.0001125000, sum = -0.0001142191
f_1 = -0.0150000000, sum = -0.0151142191
f_0 = 1.0000000000, sum = 0.9848857809
Final sum = 0.9848857809

--- [4] Method4 ---
f_1 = -0.0150000000, sum = 0.9850000000
f_2 = -0.0001125000, sum = 0.9848875000
f_3 = -0.0000016875, sum = 0.9848858125
f_4 = -0.0000000316, sum = 0.9848857809
Final sum = 0.9848857809


Actual sqrt(0.97) = 0.9848857802
Error: 0.0000000007

```


Перевірка результатів обчислень через WolframAlpha:



$$X=0.52$$

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA



$1 + ((1/2)*(0.52 - 1)) + ((-1/8)*(0.52 - 1)^2) + ((1/16)*(0.52 - 1)^3) + ((-5/128)*(0.52 - 1)^4)$

↳ = **0.7222144**

 Extended Keyboard  Upload

Input


$$1 + \frac{1}{2} (0.52 - 1) - \frac{1}{8} (0.52 - 1)^2 + \frac{1}{16} (0.52 - 1)^3 + -\frac{5}{128} (0.52 - 1)^4$$

Result



0.7222144





$$X = 0.64$$

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA



$1 + ((1/2)*(0.64 - 1)) + ((-1/8)*(0.64 - 1)^2) + ((1/16)*(0.64 - 1)^3) + ((-5/128)*(0.64 - 1)^4)$

 NATURAL LANGUAGE  MATH INPUT

 EXTENDED KEYBOARD  EXAMPLES  UPLOAD  RANDOM

Input


$$1 + \frac{1}{2} (0.64 - 1) - \frac{1}{8} (0.64 - 1)^2 + \frac{1}{16} (0.64 - 1)^3 + -\frac{5}{128} (0.64 - 1)^4$$

Result







0.8002279

X=0.71

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA



$1 + ((1/2)*(0.71 - 1)) + ((-1/8)*(0.71 - 1)^2) + ((1/16)*(0.71 - 1)^3) + ((-5/128)*(0.71 - 1)^4)$

 NATURAL LANGUAGE  MATH INPUT  EXTENDED KEYBOARD  EXAMPLES  UPLOAD  RANDOM

Input


$$1 + \frac{1}{2} (0.71 - 1) - \frac{1}{8} (0.71 - 1)^2 + \frac{1}{16} (0.71 - 1)^3 + -\frac{5}{128} (0.71 - 1)^4$$

Result


0.842686905859375



X = 0.79

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA



$1 + ((1/2)*(0.79 - 1)) + ((-1/8)*(0.79 - 1)^2) + ((1/16)*(0.79 - 1)^3) + ((-5/128)*(0.79 - 1)^4)$

 = **0.8888327183593749**

 Extended Keyboard  Upload

Input


$$1 + \frac{1}{2} (0.79 - 1) - \frac{1}{8} (0.79 - 1)^2 + \frac{1}{16} (0.79 - 1)^3 + -\frac{5}{128} (0.79 - 1)^4$$

Result

0.888832718359375

$$X=0.81$$

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA



$1 + ((1/2)*(0.81 - 1)) + ((-1/8)*(0.81 - 1)^2) + ((1/16)*(0.81 - 1)^3) + ((-5/128)*(0.81 - 1)^4)$

NATURAL LANGUAGE
MATH INPUT
EXTENDED KEYBOARD
EXAMPLES
UPLOAD
RANDOM

Input


$$1 + \frac{1}{2} (0.81 - 1) - \frac{1}{8} (0.81 - 1)^2 + \frac{1}{16} (0.81 - 1)^3 + -\frac{5}{128} (0.81 - 1)^4$$

Result

0.900007905859375

$$X=0.86$$

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA



$1 + ((1/2)*(0.86 - 1)) + ((-1/8)*(0.86 - 1)^2) + ((1/16)*(0.86 - 1)^3) + ((-5/128)*(0.86 - 1)^4)$

Extended Keyboard
Upload

Input


$$1 + \frac{1}{2} (0.86 - 1) - \frac{1}{8} (0.86 - 1)^2 + \frac{1}{16} (0.86 - 1)^3 + -\frac{5}{128} (0.86 - 1)^4$$

Result

0.92736349375

$$X=0.90$$

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA



$1 + ((1/2)*(0.9 - 1)) + ((-1/8)*(0.9 - 1)^2) + ((1/16)*(0.9 - 1)^3) + ((-5/128)*(0.9 - 1)^4)$

NATURAL LANGUAGE
MATH INPUT
EXTENDED KEYBOARD
EXAMPLES
UPLOAD
RANDOM

Input


$$1 + \frac{1}{2} (0.9 - 1) - \frac{1}{8} (0.9 - 1)^2 + \frac{1}{16} (0.9 - 1)^3 + -\frac{5}{128} (0.9 - 1)^4$$

Result

0.94868359375

$$X=0.97$$

FROM THE MAKERS OF WOLFRAM LANGUAGE AND MATHEMATICA



$1 + ((1/2)*(0.97 - 1)) + ((-1/8)*(0.97 - 1)^2) + ((1/16)*(0.97 - 1)^3) + ((-5/128)*(0.97 - 1)^4)$

NATURAL LANGUAGE
MATH INPUT
EXTENDED KEYBOARD
EXAMPLES
UPLOAD
RANDOM

Input

$$1 + \frac{1}{2} (0.97 - 1) - \frac{1}{8} (0.97 - 1)^2 + \frac{1}{16} (0.97 - 1)^3 + -\frac{5}{128} (0.97 - 1)^4$$

Result

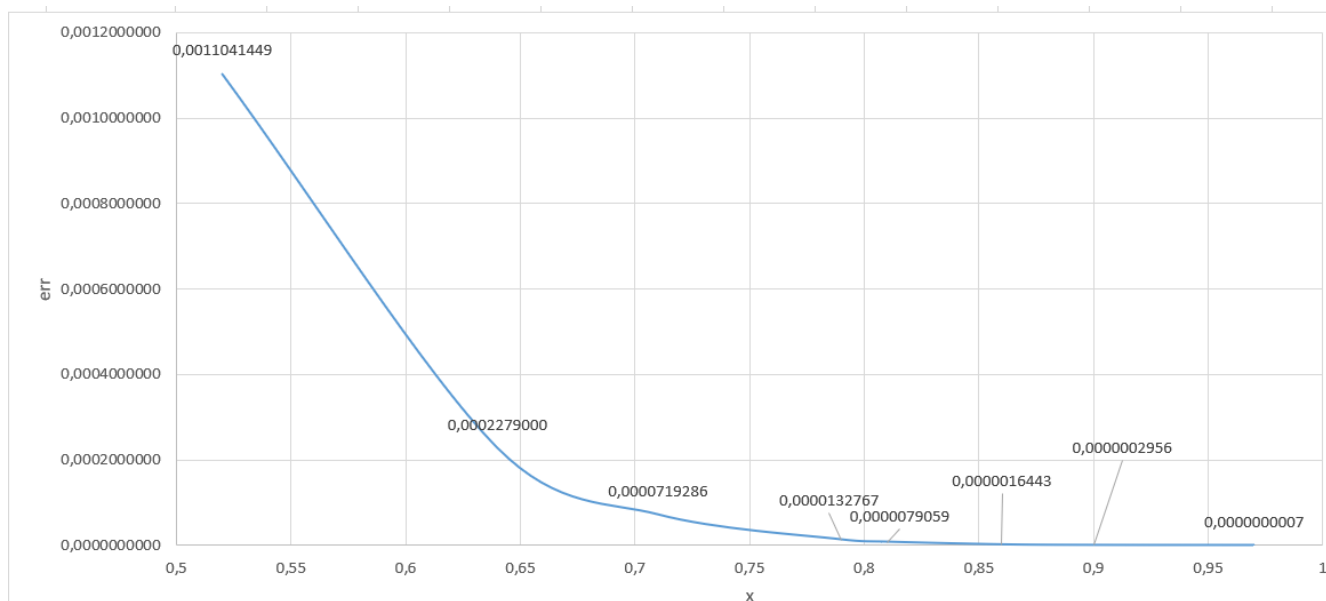
0.984885780859375

Графік похибки:

Табличка зі значеннями:

	A	B	C	D	E	F	G	H	I
1	x	0,52	0,64	0,71	0,79	0,81	0,86	0,9	0,97
2	sum	0,7222144000	0,8002279000	0,8426869059	0,8888327184	0,9000080000	0,9273634938	0,9486835937	0,9848857809
3	actual	0,7211102551	0,8000000000	0,8426149773	0,8888194417	0,9000000000	0,9273618495	0,9486832981	0,9848857802
4	err	0,0011041449	0,0002279000	0,0000719286	0,0000132767	0,0000080000	0,0000016443	0,0000002956	0,0000000007
5									

Графік:



Висновки:

У ході виконання лабораторної роботи я реалізував три рекурсивні функції, які обчислюють наближення квадратного кореня з числа x за допомогою ряду, розкладеного за узагальненою біноміальною формулою.

Для тестування використав циклічний (ітераційний) варіант обчислення функції, який повторює логіку обчислення ряду без використання рекурсії.