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

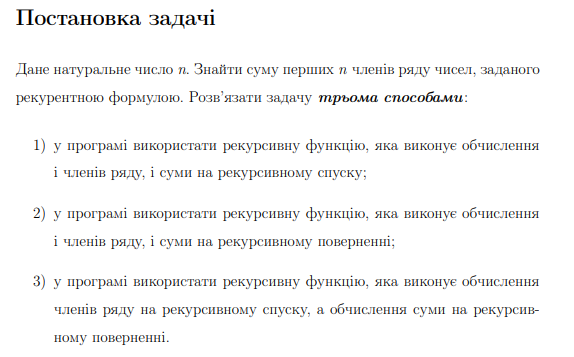
**Лабораторна робота №1**

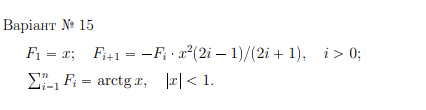
з дисципліни  
 «Алгоритми і структури даних»

Виконав: Перевірила:

студент групи ІМ-32 Молчанова А. А.  
Король Олександр Володимирович  
номер у списку групи: 15

Київ 2024





**Текст програми**

1. **програма, що рахує на спуску**

**#include <stdio.h>**

**double multiply (int i, double x) {**

**return x\*x\*(2\*i-1)/(2\*i+1);**

**}**

**double recursion(double previous, int i, double sum, double x, unsigned int n) {**

**double current;**

**if (i == 1) {**

**current = x;**

**} else {**

**current = -previous \* multiply(i-1,x);**

**}**

**sum += current;**

**if (i == n) {**

**return sum;**

**} else {**

**return recursion(current,i+1, sum, x, n);**

**}**

**}**

**double provideRecursion ( double x, int n) {**

**return recursion(0,1,0,x,n);**

**}**

**int main() {**

**double x;**

**unsigned int n;**

**printf("enter x (hint: it must be |x| < 1) = ");**

**scanf("%lf", &x);**

**if(x >= 1 || x <=-1) {**

**printf("x must be |x| < 1");**

**} else {**

**printf("enter n = ");**

**scanf("%d", &n);**

**double result = provideRecursion(x,n);**

**printf("Result: %lf\n", result);**

**}**

**return 0;**

**}**

1. **програма, що рахує при поверненні**

**#include <stdio.h>**

**double multiply (int i, double x) {**

**return x\*x\*(2\*i-1)/(2\*i+1);**

**}**

**double recursion(double x, int n, double \*sum) {**

**double current;**

**if (n == 1) {**

**current = x;**

**} else {**

**current = -recursion( x,n-1, sum) \* multiply(n-1, x);**

**}**

**\*sum += current;**

**return current;**

**}**

**double provideRecursion ( double x, int n) {**

**double sum = 0;**

**recursion(x, n, &sum);**

**return sum;**

**}**

**int main() {**

**double x;**

**int n;**

**printf("enter x (hint: it must be |x| < 1) = ");**

**scanf("%lf", &x);**

**if(x >= 1 || x <=-1) {**

**printf("x must be |x| < 1");**

**} else {**

**printf("enter n = ");**

**scanf("%d", &n);**

**printf("Result: %.10lf\n", provideRecursion(x, n));**

**}**

**return 0;**

**}**

1. **програма, що рахує в змішаному форматі**

**#include <stdio.h>**

**double multiply (int i,double x) {**

**return x\*x\*(2\*i-1)/(2\*i+1);**

**}**

**double recursion (double previous, int i, double x, int n) {**

**double current;**

**if (i == 1) {**

**current = x;**

**} else {**

**current = -previous \* multiply(i-1, x);**

**}**

**if (i == n) {**

**return current;**

**} else {**

**double sum = current + recursion(current,i+1, x, n);**

**return sum;**

**}**

**}**

**double provideRecursion (double parameter, int amount) {**

**return recursion(0,1,parameter,amount);**

**}**

**int main() {**

**double x;**

**int n;**

**printf("enter x (hint: it must be |x| < 1) = ");**

**scanf("%lf", &x);**

**if(x >= 1 || x <=-1) {**

**printf("x must be |x| < 1");**

**} else {**

**printf("enter n = ");**

**scanf("%d", &n);**

**double sum = provideRecursion(x,n);**

**printf("Result: %.10lf\n", sum);**

**}**

**return 0;**

**}**

1. **програма, що рахує циклом**

**#include <stdio.h>**

**double multiply (int i,double x) {**

**return x\*x\*(2\*i-1)/(2\*i+1);**

**}**

**double loop(double x, int amount) {**

**double sum = x;**

**double current = x;**

**for (int i = 2; i <= amount ; i++) {**

**current \*= -multiply(i-1,x);**

**sum += current;**

**}**

**return sum;**

**}**

**int main() {**

**double x;**

**int n;**

**printf("enter x (hint: it must be |x| < 1) = ");**

**scanf("%lf", &x);**

**if(x >= 1 || x <=-1) {**

**printf("x must be |x| < 1");**

**} else {**

**printf("enter n = ");**

**scanf("%d", &n);**

**printf("Result: %.10lf\n", loop(x,n));**

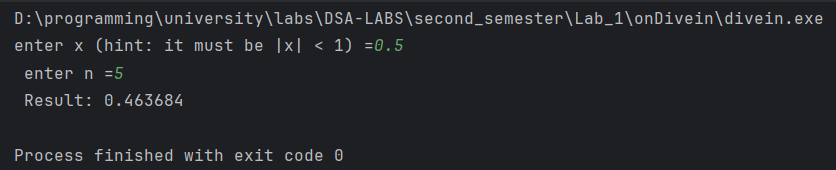
**}**

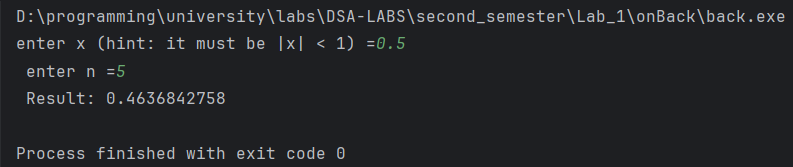
**return 0;**

**}**

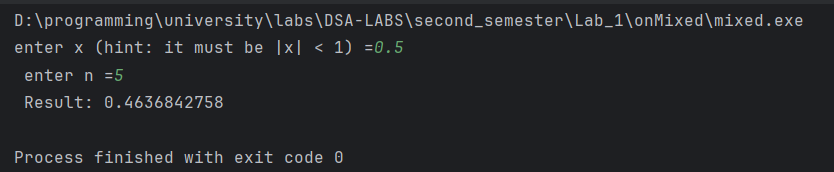
**Результати тестування програми**

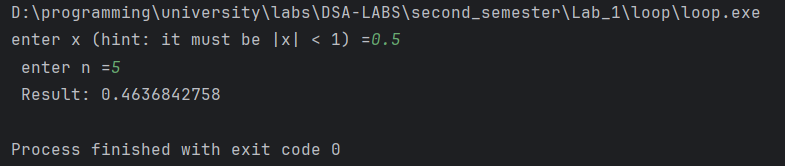
1. **для x = 0.5**

**1)**

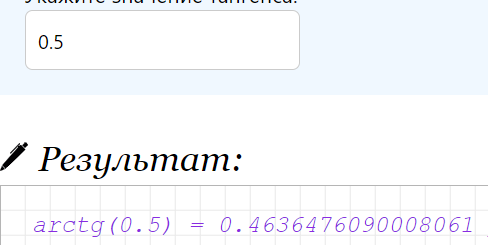
**2)**

3)

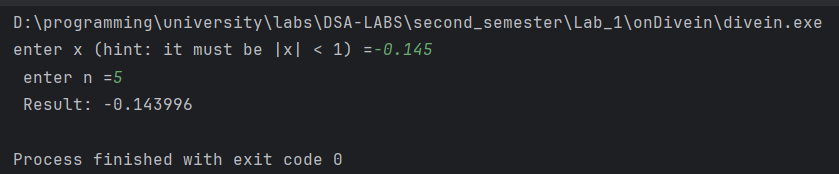


4)

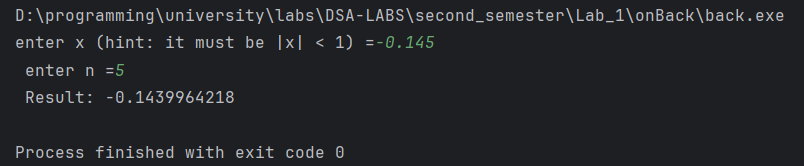
калькулятор:



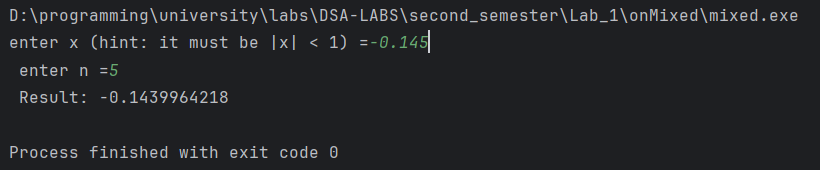
1. **для x = -0.145**

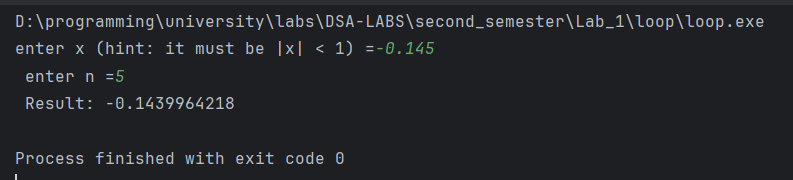
**1)**

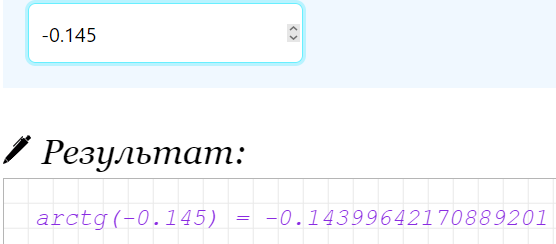
**2)**

****

3)

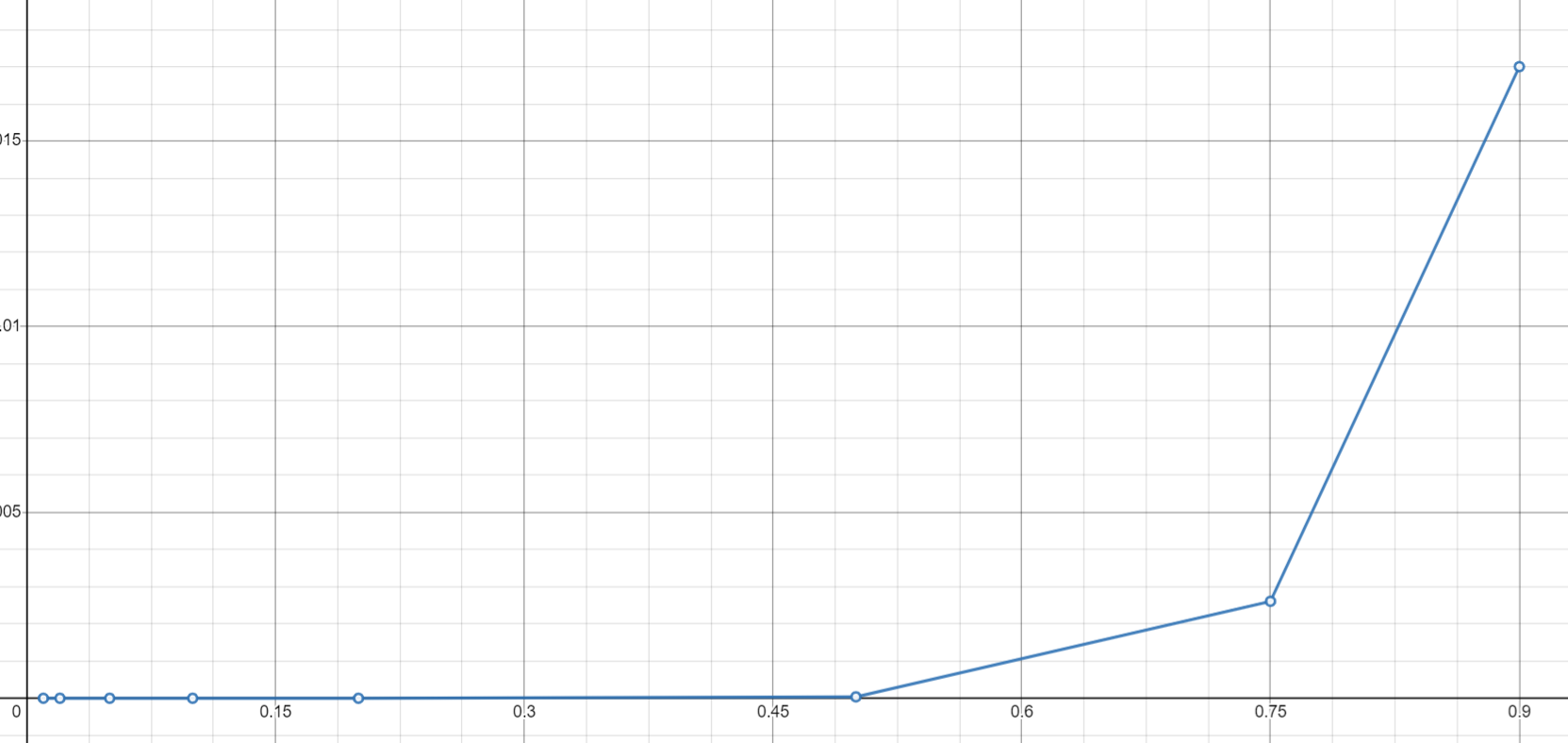


4)

калькулятор:   


Розрахунки похибки

| 0.01 | 0.02 | 0.05 | 0.1 | 0.2 | 0.5 | 0.75 | 0.9 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 11.3334762114×10−11 | 22.6849463625×10−11 | 2.1942767048×10−11 | 8.8379581431×10−12 | 1.8501192245×10−9 | 0.0000366667991939 | 0.00260844490672 | 0.0170014906135 |



Висновки: На цій лабораторній роботі ми навчились розробляти програми з використанням рекурсивних функції. Зокрема, ми вивчили теорію пов’язану з використанням рекурсивних функцій, методику обчислення значень різних функцій за допомогою рекурсії; також у результаті цієї лабораторної роботи ми здобули практичні навички з роботи з рекурсивними функціями та їх застосування на прикладі обчислення складних математичних послідовностей.

Під час лабораторної роботи ми порівняли ефективність трьох підходів: рекурсивного обчислення на спуску, на поверненні та змішаного. За результатами порівняння ми виявили, що підхід на спуску може бути більш ефективним з точки зору часу виконання, особливо для великих значень n, оскільки уникне накопичення додаткових значень на шляху до бази. Крім того, ми здійснили перевірку та тестування розроблених програм з рекурсивними функціями. Це дозволило нам переконатися в правильності реалізації алгоритмів та в їхній здатності до точних обчислень.

У підсумку, ця лабораторна робота дала нам можливість поглибити знання про рекурсивні функції та їх використання в обчисленнях, а також розвинула практичні навички програмування та аналізу алгоритмів.