Национальный исследовательский ядерный университет «МИФИ»

Институт интеллектуальных кибернетических систем Кафедра №12 «Компьютерные системы и технологии»







ОТЧЕТ

**О выполнении лабораторной работы №2**

**«Вычисление значений числовых рядов и функций с заданной точностью»**

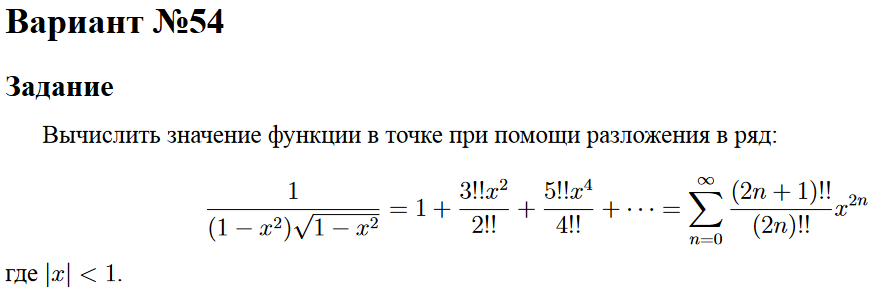
Москва — 2022

**Cтудент:** Баранов А.Т.

**Группа:** Б22-534

**Преподаватель:** Широких Т.А.

# Формулировка индивидуального задания



# Описание использованных типов данных

При выполнении данной лабораторной работы использовался встроенный тип данных int, пред­ назначенный для работы с целыми числами, и встроенный тип данных long double, предназначенный для работы с числами с плавающей точкой повышенной точности.

# Описание использованного алгоритма

Алгоритм 1: вычисления значения функции при возможности указать количество членов ряда, которое необходимо использовать при вычислениях.

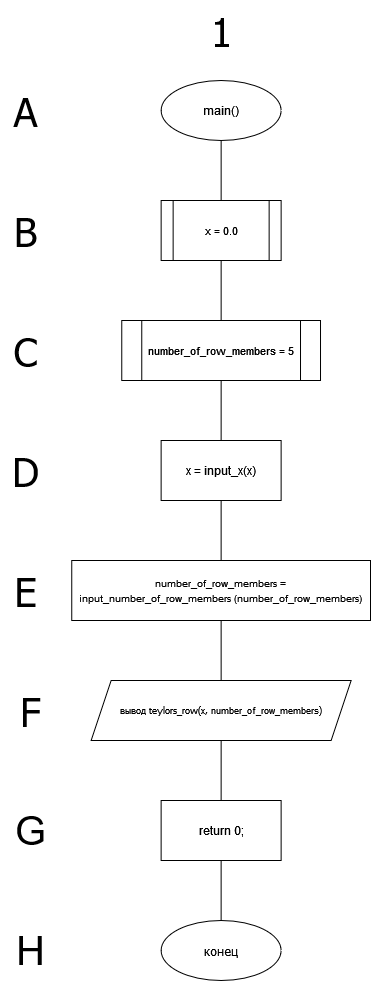


Рис. 1: Блок­схема алгоритма работы функции main()

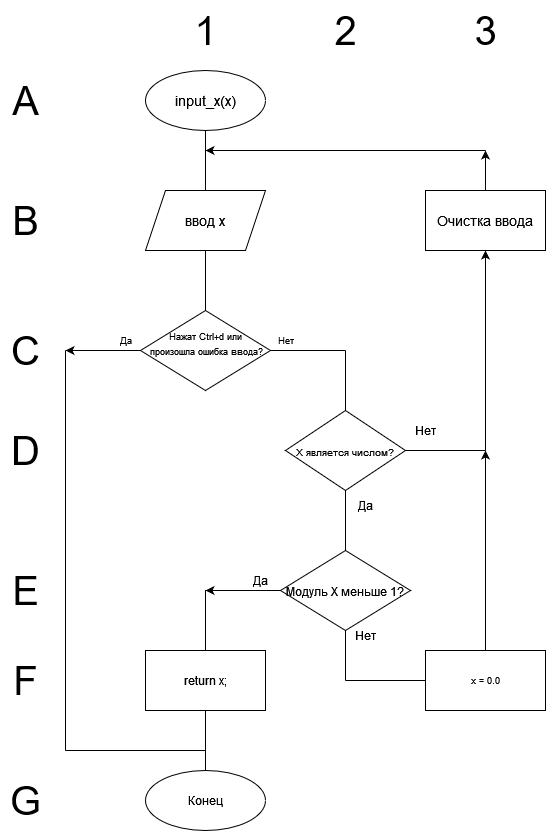


Рис. 2: Блок­схема алгоритма работы функции input\_x()

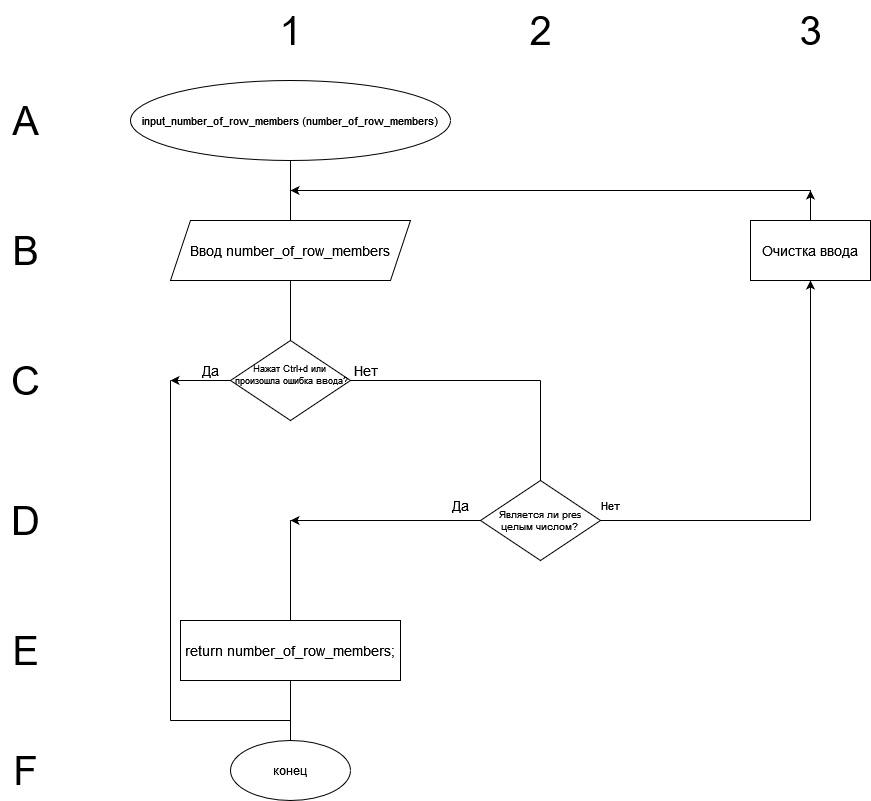


Рис. 3: Блок­схема алгоритма работы функции input\_number\_of\_row\_members()

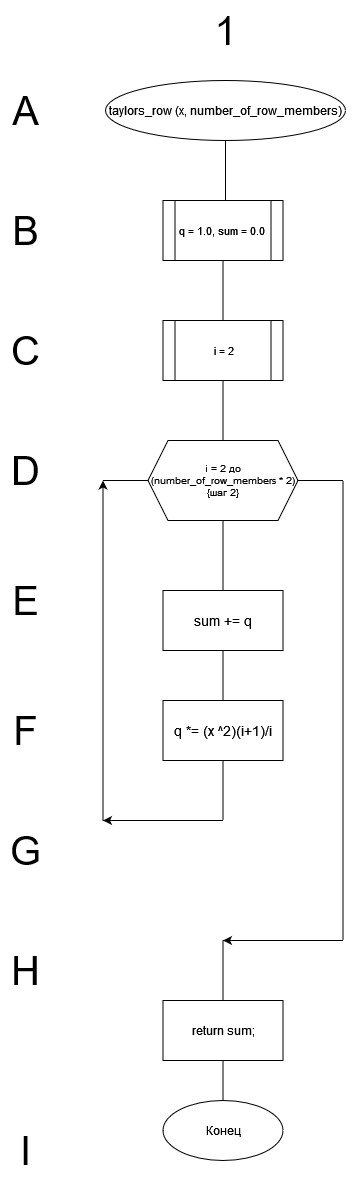


Рис. 4: Блок­схема алгоритма работы функции teylors\_row()

Алгоритм 2: вычисления значения функции при возможности указать точность, с которой должно быть вычислено значение функции.

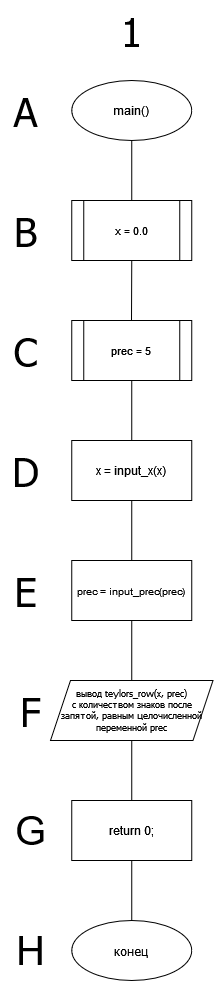
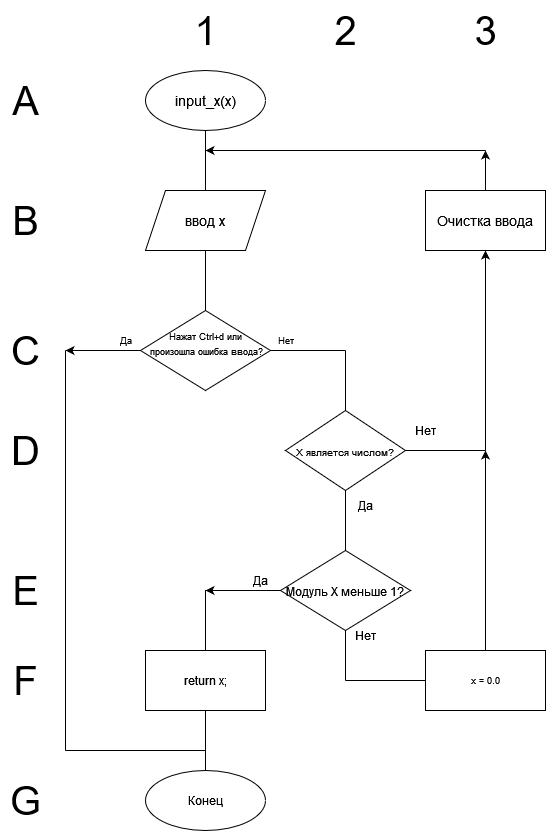
****

Рис. 5: Блок­схема алгоритма работы функции main()

Рис. 6: Блок­схема алгоритма работы функции input\_x()

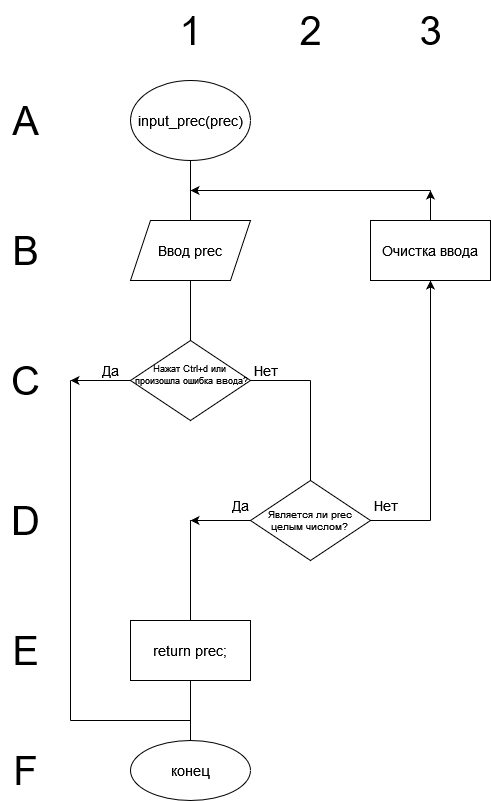


Рис. 7: Блок­схема алгоритма работы функции input\_prec()

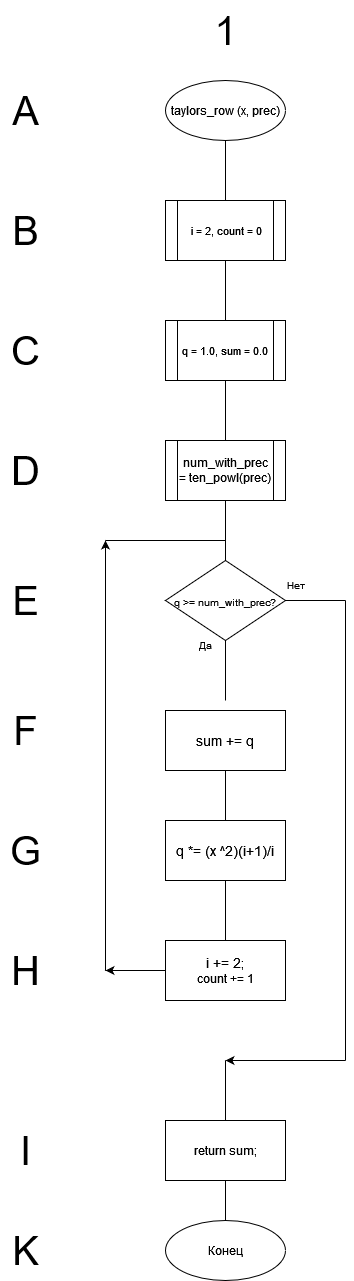


Рис. 8: Блок­схема алгоритма работы функции teylors\_row()

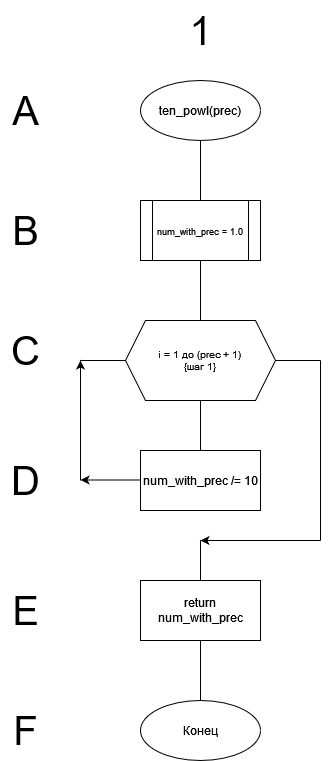


Рис. 9: Блок­схема алгоритма работы функции ten\_powl()

# Исходные коды разработанных программ

Листинг 1: Исходные коды программы code1 (файл: code1.c)

#include <stdio.h>  
#include <iso646.h>  
#include <math.h>  
  
long double input\_x(long double);  
int input\_number\_of\_row\_members(int);  
long double teylors\_row(long double, int);  
  
int main()  
{  
 long double x = 0.0;  
 int number\_of\_row\_members = 10;  
  
 x = input\_x(x);  
 number\_of\_row\_members = input\_number\_of\_row\_members(number\_of\_row\_members);  
  
 printf("\nCalculated value - %.25Lf\n", teylors\_row(x, number\_of\_row\_members));  
  
 return 0;  
}  
  
long double input\_x (long double x)  
{  
 printf("\nWrite number x in the range from -1 to 1 (except -1 and 1):");  
 while(scanf("%Lf", &x) != 1 or fabsl(x) >= 1)  
 {  
 if (fabsl(x) >= 1)  
 {  
 printf("\nYou haven`t written number in the range from -1 to 1"  
 "(except -1 and 1).Try again:");  
 x = 0.0;  
 }  
 else //as for "if (scanf("%Lf", &x) != 1)"  
 if (getchar() == -1)  
 printf("\nError or EOF\n");  
 else  
 printf("\nYou haven`t written number.Try again:");  
  
 while(getchar() != '\n');  
 }  
  
 return x;  
}  
  
int input\_number\_of\_row\_members(int number\_of\_row\_members)  
{  
 char check;  
  
 printf("\nWrite integer required number of row members:");  
 while ((scanf("%d%c", &number\_of\_row\_members, &check) != 2) or (check != 10))  
 {  
 if (getchar() == -1)  
 printf("\nError or EOF.\n");  
 else  
 printf("\nYou haven`t written integer. Try again:");  
  
 while (getchar() != '\n');  
 }  
  
 return number\_of\_row\_members;  
}  
  
long double teylors\_row(long double x, int number\_of\_members)  
{  
 long double q = 1.0, sum = 0.0;  
  
 for(int i = 2; i/2 <= number\_of\_members; i += 2)  
 {  
 sum += q;  
 q \*= (x\*x)\*(i+1)/(i);  
 }  
  
 return sum;  
}

Листинг 2: Исходные коды программы code (файл: code.c)

#include <stdio.h>  
#include <iso646.h>  
#include <math.h>  
  
long double input\_x(long double);  
int input\_prec(int);  
long double teylors\_row(long double, int);  
long double ten\_powl(int);  
  
int main()  
{  
 long double x = 0.0;  
 int prec = 5;  
  
 x = input\_x(x);  
 prec = input\_prec(prec);  
  
 printf("\nCalculated value - %.\*Lf\n", prec, teylors\_row(x, prec));  
  
 return 0;  
}  
  
long double input\_x (long double x)  
{  
 printf("\nWrite number x in the range from -1 to 1 (except -1 and 1):");  
 while(scanf("%Lf", &x) != 1 or fabsl(x) >= 1)  
 {  
 if (fabsl(x) >= 1)  
 {  
 printf("\nYou haven`t written number in range from -1 to 1"  
 "(except -1 and 1). Try again:");  
 x = 0.0;  
 }  
 else if (getchar() == -1)  
 printf("\n\nError or EOF\n");  
 else  
 printf("\nYou haven`t written number. Try again:");  
  
 while(getchar() != '\n');  
 }  
  
 return x;  
}

int input\_prec(int prec)  
{  
 char check;  
  
 printf("\nWrite integer precision (number of decimal places):");  
 while ((scanf("%d%c", &prec, &check) != 2) or (check != 10))  
 {  
 if (check == -1)  
 printf("\n\nError or EOF.\n");  
 else  
 printf("\nYou haven`t written integer. Try again:");  
  
 while(getchar() != '\n');  
 }  
  
 return prec;  
}  
  
long double teylors\_row(long double x, int prec)  
{  
 int i = 2, count = 0;  
 long double q = 1.0, num\_with\_prec = ten\_powl(prec), sum = 0.0;  
  
 while (q >= num\_with\_prec)  
 {  
 sum += q;  
 q \*= (x\*x)\*(i+1)/(i);  
 i += 2;  
 count += 1;  
 }  
  
 printf("\nNumber of row members - %d\n", count);  
  
 return sum;  
}  
  
long double ten\_powl(int prec) /\*Example: pres = 3 -> num\_with\_pres = 0.0001 to better  
 \* precision\*/  
{  
 long double num\_with\_prec = 1;  
 for (int i = 1; i <= (prec + 1); i++)  
 num\_with\_prec /= 10;  
  
 return num\_with\_prec;  
}

# Описание тестовых примеров

Таблица 1: Тестовые примеры программы №1.

|  |  |  |  |
| --- | --- | --- | --- |
| Значение  x | Значение number\_of\_row\_members | Ожидаемое значение  sum | Полученное значение  sum |
| 0.8 | 25 | 4.629395089770481181714380 | 4.629395089770481181714380 |
| 0.312879 | 30 | 1.1671135597614613957362373 | 1.1671135597614613957362373 |
| 0 | 500 | 1.0000000000000000000000000 | 1.0000000000000000000000000 |

# Скриншоты

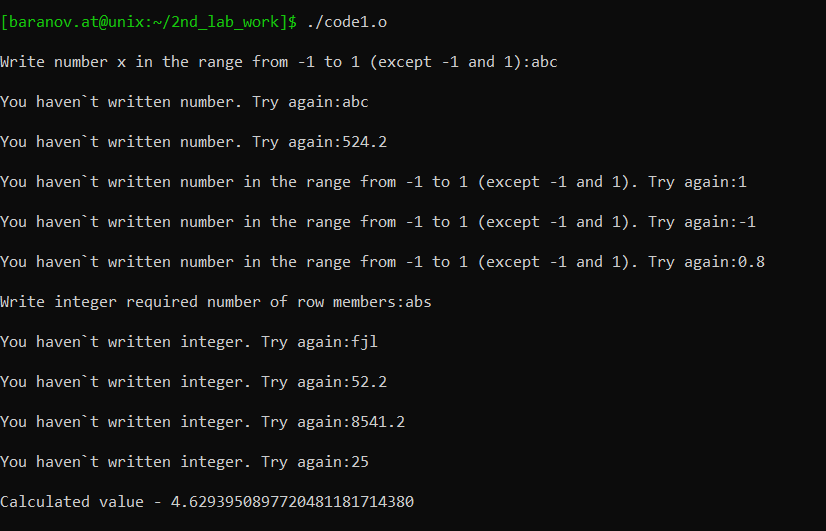
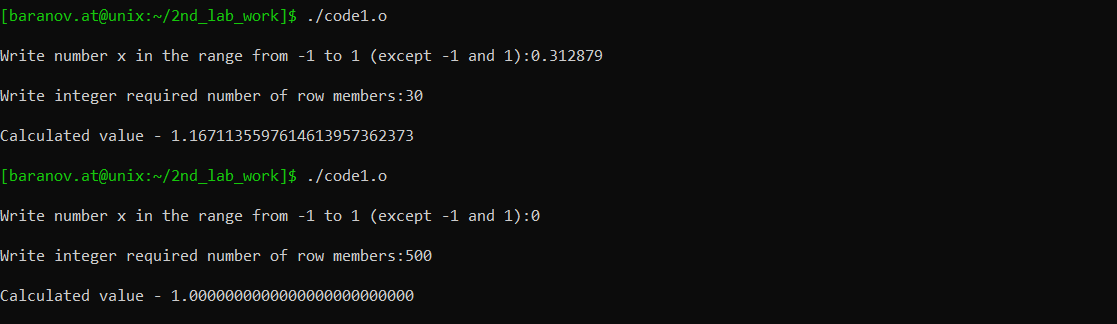
****

Рис. 10: Запуск программы prog1

Таблица 2: Тестовые примеры программы №2.

|  |  |  |  |
| --- | --- | --- | --- |
| Значение  x | Значение  pres | Ожидаемое значение  sum | Полученное значение  sum |
| 0.8 | 3 | 4.629 | 4.629 |
| 0.8 | 4 | 4.6296 | 4.6296 |
| 0 | 25 | 1.0000000000000000000000000 | 1.0000000000000000000000000 |
| 0.312879 | 6 | 1.167114 | 1.167114 |
| -0.8 | 5 | 4.62963 | 4.62963 |

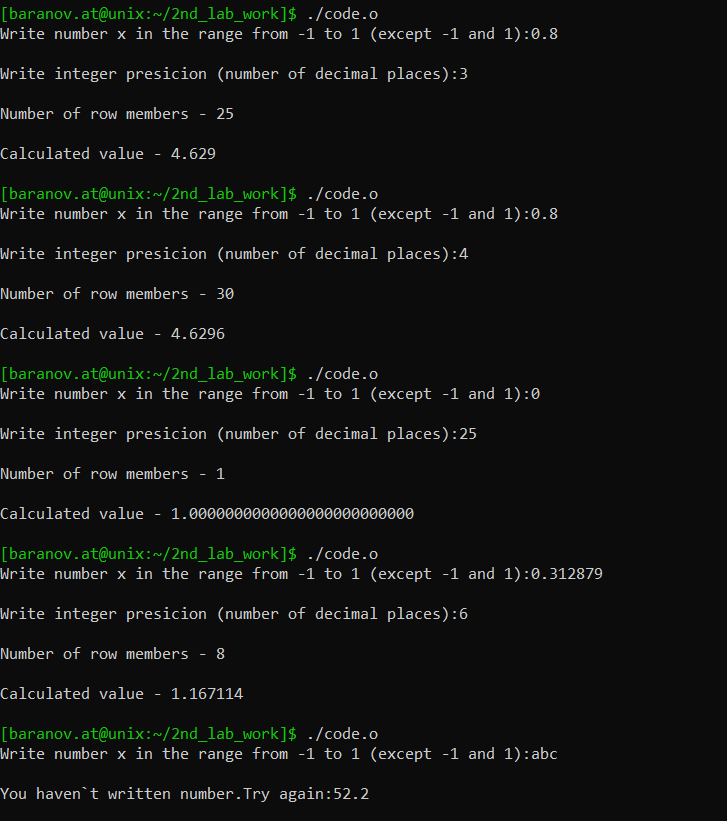
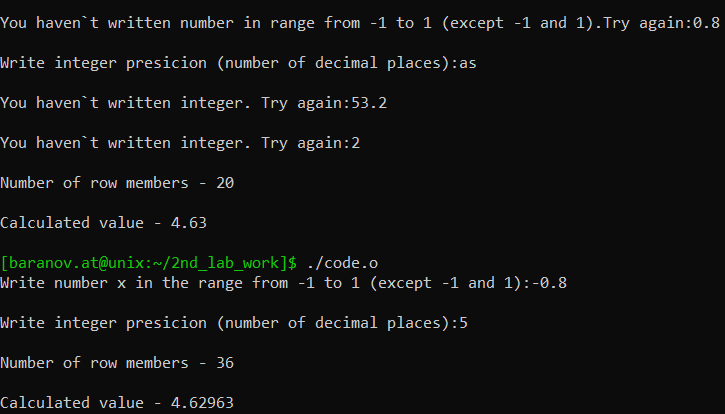


Рис. 11: Запуск программы prog

# Выводы

В ходе выполнения данной работы на примере программ, позволяющих вычислять  
значения некоторой заданной функции, были рассмотрены принципы работы построения программ на языке C и обработки чисел с плавающей точкой:

* 1. Организация ввода.
  2. Разработка функций.
  3. Объявление и использование переменных.
  4. Вычисление значения заданной функции с помощью разложения её в ряд.
  5. Работа с циклами и условными операторами.
  6. Организация вывода с количеством знаков после запятой, введённым пользователем.