

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра систем автоматизированного проектирования

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Программирование»
Тема: Реализация программы с циклами для вычисления
последовательности значений

Студентка гр. 3353

Карпенко А.Ю

Преподаватель

Калмычков В.А

Санкт-Петербург

2023

ОГЛАВЛЕНИЕ:

Исходная формулировка задания	1
Цели работы	1
Математическая постановка задачи	1
Контрольный пример	2
Особенности выполнения на компьютере	2
Реализация ввода/вывода	3
Организация хранения данных	4
Представление алгоритма решения задачи. Блок-схема	4
Текст программы	5
Результаты работы программы	6
Вывод	8

Исходная формулировка задания

Составить программу, которая будет вычислять значения суммы ряда с заданной точностью. Требуется вывести таблицу значений, содержащую n , $U(n)$, $S(n)$, а также вывести аналогичную таблицу в файл расширения txt.

Цели работы

- изучение циклов
- изучение библиотеки. fstream
- закрепление условных конструкций (ветвления)

Математическая постановка задачи

Дано: уравнение суммы конечного числа членов $U(n)$, которую нужно посчитать для точности $S(n) < E$, где E точность.

$$\sum_{n=1}^{\infty} \frac{3n^2 + 5}{4n^2 - 3n + 1} x^n$$

Для упрощения вычислений необходимо вычислить первый член последовательности

$$\frac{3 * 1^2 + 5}{4 * 1^2 - 3 * 1 + 1} x^1 = \frac{8}{2} x = 4x$$

А остальные представить через переходный коэффициент

$$\frac{\frac{3(n+1)^2 + 5}{4(n+1)^2 - 3(n+1) + 1} x^{(n+1)}}{\frac{3n^2 + 5}{4n^2 - 3n + 1} x^n} = \frac{x(12n^4 + 15n^3 + 17n^2 - 18n + 8)}{12n^4 + 15n^3 + 26n^2 + 25n + 10}$$

Тогда каждый следующий член будет равен предыдущему, умноженному на переходный коэффициент

$$U_2 = 4x * \frac{x(12*2^4 + 15*2^3 + 17*2^2 - 18*2 + 8)}{12*2^4 + 15*2^3 + 26*2^2 + 25*2 + 10}$$

$$U_3 = U_2 * \frac{x(12*3^4 + 15*3^3 + 17*3^2 - 18*3 + 8)}{12*3^4 + 15*3^3 + 26*3^2 + 25*3 + 10}$$

Аналогично поступаем с суммой

$$S_2 = U_1 + U_2$$

$$S_3 = S_2 + U_3$$

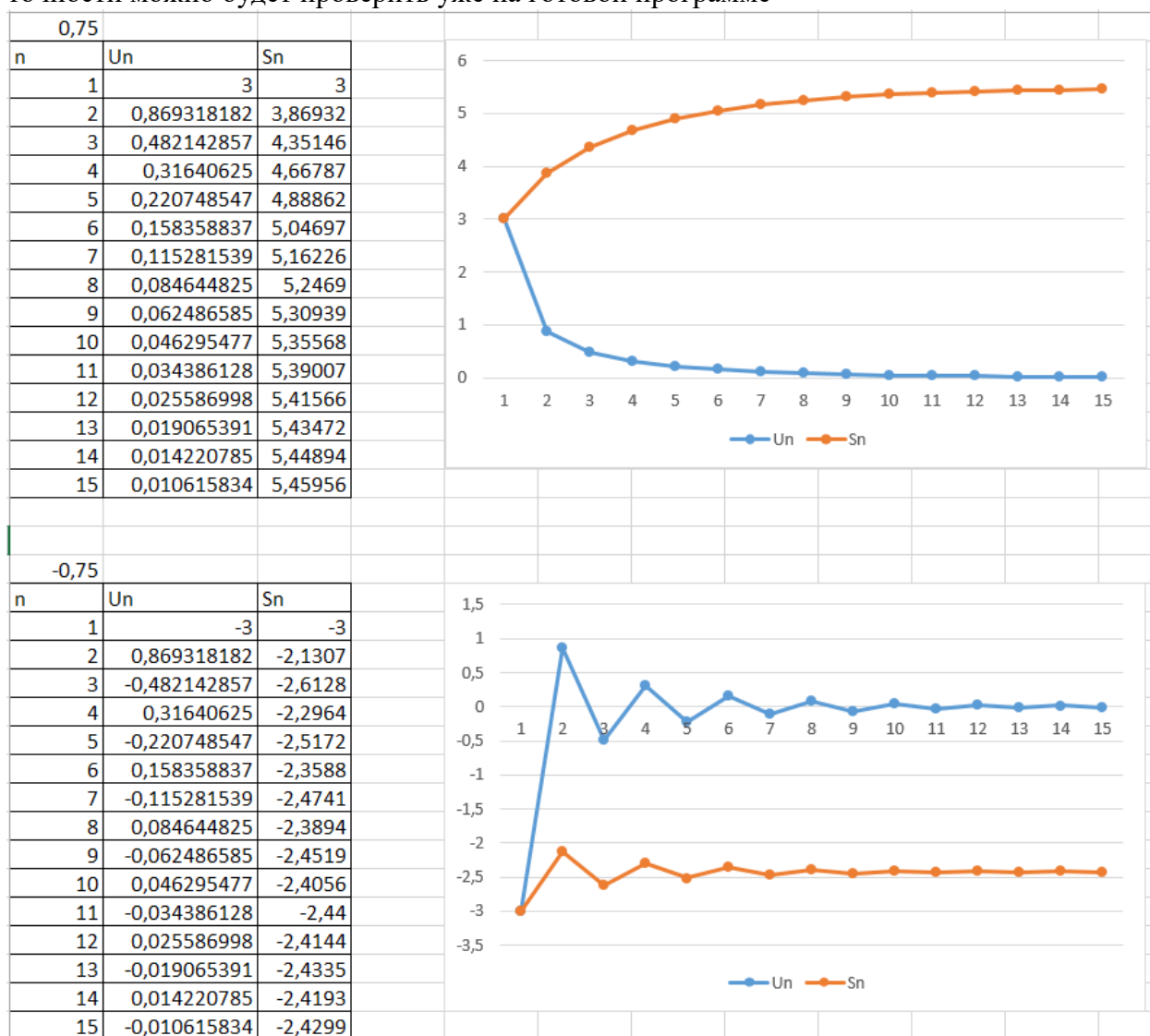
Делаем так до тех пор, пока

$$|U_n| < E$$

Кроме того, следует отметить, что ряд имеет радиус сходимости и ряд сходится при $|x| < 1$, поэтому на x следует наложить соответствующие ограничения

Контрольный пример

Для контрольного примера, приведем два результата ручного расчета и график для $x=0.75$ и $x=-0.75$ произведем вычисление первых 15 членов в обоих случаях, выполнения условия точности можно будет проверить уже на готовой программе



Особенности выполнения на компьютере

В первую очередь в самой программе x , E зададим переменной типа float. float представляет вещественное число с плавающей точкой в диапазоне от $-3.4E-38$ до $3.4E38$ и в памяти занимает 4 байта (32 бита), что накладывает свои ограничения на величину E , x .

В программе значения суммы S_n и значения элемента ряда U_n зададим переменной типа double. double представляет вещественное число с плавающей точкой в диапазоне от $-1.7E-308$ до $1.7E308$ и в памяти занимает 8 байт (64 бита), что делает труднодостижимый возникновение придельных для типа переменной значений U_n и S_n . Переменные n, p требуют целочисленного значения зададим тип переменной int, который хранит целочисленные значения в диапазоне от -32768 до 32767 и занимает 2 байта (16 бит)

В процессе выполнения программы потребуется реализация условных конструкций:

```

if (условие 1) {(выполняется если условие 1 истина);}
else if(условие 2) {выполняется если условие 2 истина;}
else {выполняется если условие 1 И условие 2 ложь;};

```

А также потребуется циклы. Будем использовать цикл:

```

do {
    (тело цикла)
}while(условие выхода из цикла)

```

Реализация ввода/вывода

Программа представляется, выводит исходную формулу и просит ввести X, E, после чего выводит шапку таблицы, выводит строки таблицы и в конце подводит итог.

Вывод приветствия и исходной формулы:

Карпенко Анастасия
Группа:3353
Начало:10.10.23

```

oo
\--- 3n^2 + 5
> ----- x^n
/___ 4n^2-3n+1
n=1

```

Необходимо задать x и точность E
Ряд сходится при $|x| < 1$
введите x в промежутке $(-1;1)$

Ввод x– введите x, при $x \in (-1;1)$:

(Если пользователь сделал это верно – «значение введено верно».

Если ошибочно – «значение введено не верно».

Если пользователь ввел ошибочно 3 раза – «перезапустите программу и обратите внимание на условия ввода данных»)

Ввод E– введите степень < -10 числа E:

(Если пользователь сделал это верно – «значение введено верно».

Если ошибочно – «значение введено не верно».

Если пользователь ввел ошибочно 3 раза – «перезапустите программу и обратите внимание на условия ввода данных»)

Вывод в консоль:

Вывод шапки таблицы:

n	Un	Sn

Вывод первого ряда $n=1$, $Un=U(n)$ $Sn=U(n)$:

1	[±]d.dddde-dd	[±]d.dddde-dd
---	---------------	---------------

Вывод последующих членов ряда $n=n+1$, $Un=U(n)$ $Sn=U(n)$

(повторяется по количеству шагов):

(n+1)	[±]d.dddde-dd	[±]d.dddde-dd
-------	---------------	---------------

Вывод результата:

Результат:

Кол-во шагов:dd

Sn:d.dddde-dd

Последний Un:dddde-d

Вывод в файл:

Вывод шапки таблицы:

n	Un	Sn

Вывод первого ряда $n=1$, $Un=U(n)$ $Sn=U(n)$:

1	[±]d.dddde-dd	[±]d.dddde-dd
---	---------------	---------------

Вывод последующих членов ряда $n=n+1$, $Un=U(n)$ $Sn=U(n)$

(повторяется по количеству шагов):

(n+1)	[±]d.dddde-dd	[±]d.dddde-dd
-------	---------------	---------------

Вывод результата:

Результат:

Кол-во шагов:dd

Sn:d.ddddddddddddddddde-dd
Последний Un:ddddddddddddddddddde-d

Для ввода значений мы в первых будем использовать:

`cin>>(имя переменной)`

Для ввода в консоль мы используем библиотеке конструкции `iostream`

`cout<<«Текст»<<(имя переменной);`

Для вывода в файл-библиотеку `fstream`

`out<<«Текст»<<(имя переменной);`

В обоих случая для вывода будем использовать модификаторы потока:

`<<setiosflags(ios::showpos)<<`-устанавливает флаг обязательной публикации знака;

`<<resetiosflags(ios::showpos)<<`-снимает флаг обязательной публикации знака;

`<<setiosflags(ios::scientific)<<`-устанавливает тип выводимого значения-научный

`<<setprecision(17)<<`-для установки количества знаков после точки;

`<<setw(24)<<`-количество выделенных ячеек для значения.

Так же стоит упомянуть, что для работы с файлом потребуется о открыть поток вывода и открыть файл:

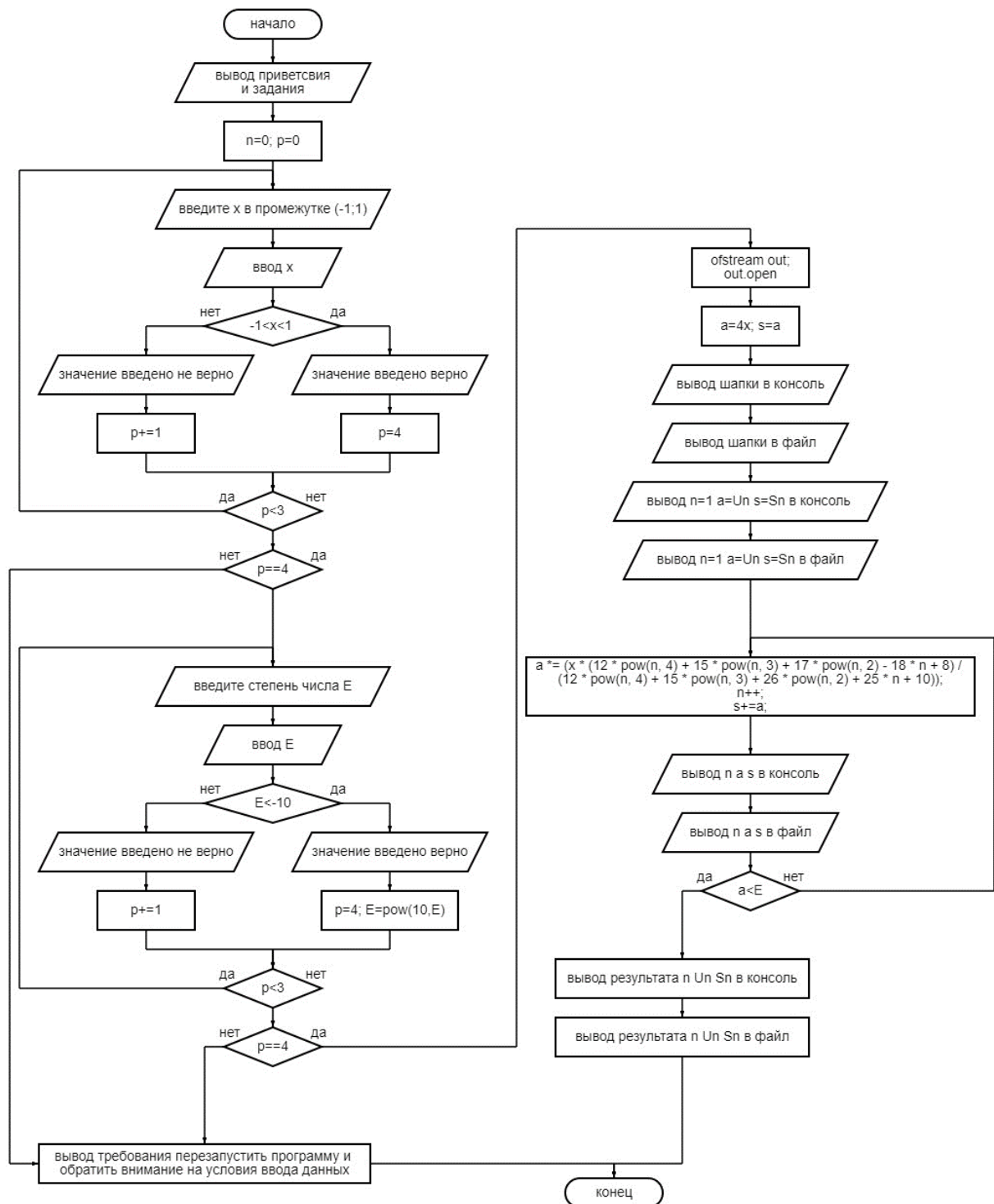
`ofstream out;
out.open("Таблица.txt");`

Организация хранения данных

Название переменной	Тип переменной	Пояснение
x	float	Хранит значение x
E	float	Хранит значение E
n	int	Хранит значение n (номер члена ряда)
p	int	Хранит значение p (необходимо для проверки E, x)
a	double	Хранит значение Un (значение n-го члена ряда)
s	double	Хранит значение Sn (сумма n-членов ряда)

Представление алгоритма решения задачи. Блок-схема

В первую очередь программа должна представиться, показать задание, и запрашивает x, выполняет проверку, если соответствует требованию-говорит «принято», если не соответствует-просит ввести еще раз, если пользователь ошибся 3 раза-выводит сообщение требующие перезапуска программы, аналогично запрашивается и проверяется степень числа E. Если все хорошо и программа приняла 2 значения то программа считает 1 член ряда выводит шапку и первую строчку значений для таблицы (и в консоль и в файл), потом внутри цикла начинает вычислять значения Un, Sn и n++ и выводить новую строчку в таблицу и в консоль соответствующие текущему значению n, выход из цикла осуществляется по условию $|Un| < E$, но так же не более 1000 шагов следовательно $n > 1000$ После завершения цикла выходит результат количество шагов(n), Sn и последний Un.



Текст программы

```

#include <iostream>
#include <iomanip>
#include <math.h>
#include <fstream>
using namespace std;
int main()
{
    setlocale(LC_ALL, "RU");
    float E, x;
    double a, s;
    int n, p;
    n = 1;
    p = 0;

    cout
    << "Карпенко Анастасия " << "\n"
    << "Группа:3353" << "\n"
    << "Начало:10.10.23" << "\n"

```

```
<< "0"
<< " \\--- 3n^2 + 5 " << "\n"
<< " > ----- x^n " << "\n"
<< " /--- 4n^2-3n+1 " << "\n"
<< " n=1 " << "\n"
<< "Необходимо задать х и точность Е" << "\n"
<< "Ряд сходится при |х|<1" << "\n";
do {
cout << "введите х в промежутке (-1;1)" << "\n";
cin >> x;
if (-1 < x && x < 1) { cout << "значение введено верно" << "\n"; p = 4; }
else { cout << "значение введено не верно" << "\n"; p += 1; }
} while (p < 3);
if (p == 4) {
p = 0;
do {
cout << "введите степень числа Е <-10" << "\n";
cin >> E;
if (E < -10) { cout << "значение введено верно" << "\n"; p = 4; E = pow(10, E); }
else { cout << "значение введено не верно" << "\n"; p += 1; }
} while (p < 3);
}
if (p == 4) {
ofstream out;
out.open("Таблица.txt");
cout << "|      n          |      Un          |      Sn          |" << "\n"
<< "-----|-----|-----|" << "\n";
out << "|      n          |      Un          |      Sn          |" << "\n"
<< "-----|-----|-----|" << "\n";
a = 4*x;
s = a;
cout << resetiosflags(ios::showpos) << "|      " << setw(5) << n << "    |      " << setiosflags(ios::showpos) << setprecision(12) << setw(20)
<< setiosflags(ios::scientific) <<
a << "    |      " << setprecision(17) << setw(24) << setiosflags(ios::showpos) << setiosflags(ios::scientific) << s << "    |" << "\n";
out << resetiosflags(ios::showpos) << "|      " << setw(5) << n << "    |      " << setiosflags(ios::showpos) << setprecision(12) << setw(20)
<< setiosflags(ios::scientific) <<
a << "    |      " << setprecision(17) << setw(24) << setiosflags(ios::showpos) << setiosflags(ios::scientific) << s << "    |" << "\n";
do
{
a *= (x * (12 * pow(n, 4) + 15 * pow(n, 3) + 17 * pow(n, 2) - 18 * n + 8) / (12 * pow(n, 4) + 15 * pow(n, 3) + 26 * pow(n, 2) + 25 * n + 10));
n++;
s += a;
cout << resetiosflags(ios::showpos) << "|      " << setw(5) << n << "    |      " << setiosflags(ios::showpos) << setprecision(12) << setw(20)
<< setiosflags(ios::scientific) <<
a << "    |      " << setprecision(17) << setw(24) << setiosflags(ios::showpos) << setiosflags(ios::scientific) << s << "    |" << "\n";
out << resetiosflags(ios::showpos) << "|      " << setw(5) << n << "    |      " << setiosflags(ios::showpos) << setprecision(12) << setw(20)
<< setiosflags(ios::scientific) <<
a << "    |      " << setprecision(17) << setw(24) << setiosflags(ios::showpos) << setiosflags(ios::scientific) << s << "    |" << "\n";
} while (n<1001 && fabs(a)>E);
cout << resetiosflags(ios::showpos) << "Результат:" << "\n" << "n:" << n << "\n" << "Sn:" << s << "\n" << setprecision(22) << "Последний Un:" << a;
out << resetiosflags(ios::showpos) << "Результат:" << "\n" << "n:" << n << "\n" << "Sn:" << s << "\n" << setprecision(22) << "Последний Un:" << a;
out.close();
}
else { cout << "перезапустите программу и обратите внимание на условия ввода данных"; }
```

Результаты работы программы

Результат работы программы при неверно введенных значениях x и E

Карпенко Анастасия

Группа: 3353

Начало: 10.10.23

$$\sum_{n=1}^{\infty} \frac{3n^2 + 5}{4n^2 - 3n + 1} x^n$$

Необходимо задать x и точность ϵ

Ряд сходится при $|x| < 1$

введите x в промежутке $(-1;1)$

3

значение введено не верно

введите x в промежутке $(-1;1)$

-1.1

значение введено не верно

введите x в промежутке $(-1;1)$

0.75

значение введено верно

введите степень числа E < -10

9

значение введено не верно

введите степень числа $E \leq -10$

-10

значение введено не верно

значение введено не верно
введите степень числа $E \leq -10$

10

значение введено не верно

перезапустите программу и обр

В качестве тестов возьмем значения из контрольных примеров для того, чтобы убедиться в правильности работы программы. Ниже представлен вывод в консоль и в файл.

X=0.75	X=-0.75
E=-11	E=-11

Карпенко Анастасия Группа: 3353 Начало: 10.10.23 oo \\--- 3n^2 + 5 > ----- x^n /_ 4n^2-3n+1 n=1 Необходимо задать x и точность E Ряд сходится при x <1 введите x в промежутке (-1;1) 0.75 значение введено верно введите степень числа E < -10 -11 значение введено верно			Карпенко Анастасия Группа: 3353 Начало: 10.10.23 oo \\--- 3n^2 + 5 > ----- x^n /_ 4n^2-3n+1 n=1 Необходимо задать x и точность E Ряд сходится при x <1 введите x в промежутке (-1;1) -0.75 значение введено верно введите степень числа E < -10 -11 значение введено верно		
nUnSn			nUnSn		
1	+3.000000000000e+00	+3.0000000000000000e+00	1	-3.000000000000e+00	-3.0000000000000000e+00
2	+8.693181818182e-01	+3.86931818181818166e+00	2	+8.693181818182e-01	-2.13068181818181834e+00
3	+4.821428571429e-01	+4.3514610389610391e+00	3	-4.821428571429e-01	-2.61282467532467555e+00
4	+3.164062500000e-01	+4.6678672889610391e+00	4	+3.164062500000e-01	-2.29641842532467555e+00
5	+2.207485465116e-01	+4.8886158354726675e+00	5	-2.207485465116e-01	-2.51716697183630345e+00
6	+1.583588367372e-01	+5.04697467220987228e+00	6	+1.583588367372e-01	-2.35880813509909881e+00
7	+1.152815385298e-01	+5.16225621073970231e+00	7	-1.152815385298e-01	-2.47408967362892840e+00
8	+8.464482516178e-02	+5.246901803590148471e+00	8	+8.464482516178e-02	-2.38944484846714600e+00
9	+6.248658455458e-02	+5.30938762045606705e+00	9	-6.248658455458e-02	-2.45193143302172878e+00
10	+4.629547705226e-02	+5.35568309750832938e+00	10	+4.629547705226e-02	-2.40563595596946689e+00
11	+3.438612845092e-02	+5.39006922595924642e+00	11	-3.438612845092e-02	-2.44002208442083939e+00
12	+2.558699784570e-02	+5.41565622380494371e+00	12	+2.558699784570e-02	-2.41443508657468664e+00
13	+1.906539055995e-02	+5.43472161436489021e+00	13	-1.906539055995e-02	-2.43350047713463558e+00
14	+1.422078488833e-02	+5.44894239925322310e+00	14	+1.422078488833e-02	-2.41927969224630024e+00
15	+1.061583351274e-02	+5.45955823276596508e+00	15	-1.061583351274e-02	-2.42989552575904266e+00
16	+7.929853142926e-03	+5.46748808590889190e+00	16	+7.929853142926e-03	-2.42196567261611628e+00
17	+5.926562048357e-03	+5.47341464795724875e+00	17	-5.926562048357e-03	-2.42789223466447131e+00
18	+4.431249220472e-03	+5.47784589717771997e+00	18	+4.431249220472e-03	-2.42346098544080183e+00
19	+3.314388654717e-03	+5.48116028583243686e+00	19	-3.314388654717e-03	-2.42677537409871835e+00
20	+2.47960146928e-03	+5.48364004597936461e+00	20	+2.47960146928e-03	-2.42429561395179061e+00
21	+1.855773849106e-03	+5.48549581982847023e+00	21	-1.855773849106e-03	-2.42615138780089667e+00
22	+1.389100512166e-03	+5.4868849198063643e+00	22	+1.389100512166e-03	-2.42476228764873047e+00
23	+1.039973251214e-03	+5.48792489323185073e+00	23	-1.039973251214e-03	-2.42580226089994477e+00
24	+7.787179059561e-04	+5.48870361113780714e+00	24	+7.787179059561e-04	-2.42502354299398881e+00
25	+5.831746501856e-04	+5.48928678578799278e+00	25	-5.831746501856e-04	-2.42560671764417446e+00
26	+4.367874525455e-04	+5.48972357324053828e+00	26	+4.367874525455e-04	-2.42516993019162896e+00
27	+3.271812707748e-04	+5.49005075451131308e+00	27	-3.271812707748e-04	-2.42549711146240377e+00
28	+2.451027326306e-04	+5.49029585724394398e+00	28	+2.451027326306e-04	-2.42525200872977331e+00
29	+1.836304754728e-04	+5.49047948771941652e+00	29	-1.836304754728e-04	-2.42543563920524630e+00
30	+1.375860307348e-04	+5.49061707375015118e+00	30	+1.375860307348e-04	-2.42529805317451164e+00
31	+1.030940317986e-04	+5.49072016778194971e+00	31	-1.030940317986e-04	-2.42540114720631017e+00
32	+7.725371505683e-05	+5.49079742149700678e+00	32	+7.725371505683e-05	-2.42532389349125355e+00
33	+5.789343652382e-05	+5.49085531493353063e+00	33	-5.789343652382e-05	-2.42538178692777739e+00
34	+4.338715275743e-05	+5.49089870208628827e+00	34	+4.338715275743e-05	-2.42533839977501975e+00
35	+3.251717909326e-05	+5.49093121926538164e+00	35	-3.251717909326e-05	-2.42537091695411311e+00
36	+2.437152596865e-05	+5.49095559079135054e+00	36	+2.437152596865e-05	-2.42534654542814465e+00
37	+1.826708220357e-05	+5.49097385787355385e+00	37	-1.826708220357e-05	-2.42536481251034841e+00
38	+1.369212544463e-05	+5.4909875499899885e+00	38	+1.369212544463e-05	-2.42535112038490386e+00
39	+1.026328887810e-05	+5.49099781328787717e+00	39	-1.026328887810e-05	-2.42536138367378173e+00
40	+7.693343687577e-06	+5.49100550663156461e+00	40	+7.693343687577e-06	-2.42535369033009429e+00
41	+5.767075344132e-06	+5.49101127370690900e+00	41	-5.767075344132e-06	-2.42535945740534823e+00
42	+4.323218009764e-06	+5.4910159692491836e+00	42	+4.323218009764e-06	-2.42535513418742843e+00
43	+3.240924132477e-06	+5.49101883784905098e+00	43	-3.240924132477e-06	-2.42535837511156105e+00
44	+2.429629669804e-06	+5.49102126747872088e+00	44	+2.429629669804e-06	-2.42535945481891151e+00
45	+1.821462061075e-06	+5.49102308894078206e+00	45	-1.821462061075e-06	-2.42535776694395233e+00
46	+1.365552515793e-06	+5.49102445449329757e+00	46	+1.365552515793e-06	-2.42535640139143638e+00
47	+1.023774636389e-06	+5.49102547826793419e+00	47	-1.023774636389e-06	-2.42535742516607256e+00
48	+7.675514696852e-07	+5.49102624581940368e+00	48	+7.675514696852e-07	-2.42535665761460306e+00
49	+5.754629589801e-07	+5.49102682128236275e+00	49	-5.754629589801e-07	-2.42535723037562121e+00
50	+4.314530536478e-07	+5.49102725273541648e+00	50	+4.314530536478e-07	-2.42535680162450840e+00
51	+3.234861119817e-07	+5.49102757622152815e+00	51	-3.234861119817e-07	-2.42535712511062052e+00
52	+2.425399585344e-07	+5.49102781876148693e+00	52	+2.425399585344e-07	-2.42535688257066218e+00
53	+1.818512107172e-07	+5.49102800061269747e+00	53	-1.818512107172e-07	-2.42535706447827721e+00
54	+1.363496513665e-07	+5.49102813696234904e+00	54	+1.363496513665e-07	-2.42535692807222114e+00
55	+1.022342758061e-07	+5.49102823919662519e+00	55	-1.022342758061e-07	-2.42535703030649850e+00
56	+7.665551110805e-08	+5.49102831585214268e+00	56	+7.665551110805e-08	-2.42535695365979618e+00
57	+5.747705005474e-08	+5.49102837332919247e+00	57	-5.747705005474e-08	-2.42535701112803004e+00
58	+4.309724020164e-08	+5.49102841642643291e+00	58	+4.309724020164e-08	-2.42535696803079004e+00
59	+3.231529914531e-08	+5.49102844874173179e+00	59	-3.231529914531e-08	-2.42535700034608936e+00
60	+2.423094982443e-08	+5.49102847297268148e+00	60	+2.423094982443e-08	-2.425356976711513967e+00
61	+1.816921038617e-08	+5.49102849114189162e+00	61	-1.816921038617e-08	-2.42535699428435025e+00
62	+1.362400703401e-08	+5.49102850476589843e+00	62	+1.362400703401e-08	-2.42535698066034300e+00
63	+1.021590152070e-08	+5.49102851498180033e+00	63	-1.021590152070e-08	-2.42535699087624446e+00
64	+7.660399562714e-09	+5.49102852264219976e+00	64	+7.660399562714e-09	-2.42535698321584503e+00
65	+5.744191327268e-09	+5.49102852838639066e+00	65	-5.744191327268e-09	-2.42535698896003638e+00
66	+4.307338384876e-09	+5.49102853269372915e+00	66	+4.307338384876e-09	-2.42535698465269789e+00
67	+3.229918656610e-09	+5.49102853592364770e+00	67	-3.229918656610e-09	-2.42535698788261644e+00
68	+2.422013528056e-09	+5.49102853834566140e+00	68	+2.422013528056e-09	-2.42535698546060274e+00
69	+1.816200634278e-09	+5.491028540616186193e+00	69	-1.816200634278e-09	-2.42535698727680327e+00
70	+1.361925212863e-09	+5.49102854152378672e+00	70	+1.361925212863e-09	-2.42535698591487803e+00
71	+1.021279890481e-09	+5.49102854254506667e+00	71	-1.021279890481e-09	-2.42535698693615798e+00
72	+7.658404402115e-10	+5.49102854331090739e+00	72	+7.658404402115e-10	-2.42535698617031770e+00
73	+5.742932617873e-10	+5.49102854388520090e+00	73	-5.742932617873e-10	-2.42535698674461075e+00
74	+4.306564701681e-10	+5.49102854431585730e+00	74	+4.306564701681e-10	-2.42535698631395435e+00
75	+3.229460577756e-10	+5.49102854488097947e+00	75	-3.229460577756e-10	-2.4253569863690047e+00
76	+2.421757662929e-10	+5.49102854506258620e+00	76	+2.421757662929e-10	-2.42535698639472486e+00
77	+1.8168071715437e-10	+5.49102854519877326e+00	77	-1.8168071715437e-10	-2.4253569865763203e+00
78	+1.361873782537e-10	+5.49102854530090045e+00	78	+1.361873782537e-10	-2.42535698644014452e+00
79	+1.021273861189e-10	+5.49102854537748630e+00	79	-1.021273861189e-10	-2.42535698654227172e+00
80	+7.658593323214e-11	+5.49102854547798813e+00	80	+7.658593323214e-11	-2.4253569864565857e+00
81	+5.743242861713e-11	+5.49102854551028585e+00	81	-5.743242861713e-11	-2.42535698652311815e+00
82	+4.306918789218e-11	+5.49102854553450648e+00	82	+4.306918789218e-11	-2.42535698648004905e+00
83	+3.229813635172e-11	+5.4910285455266973e+00	83	-3.229813635172e-11	-2.42535698651234721e+00
84	+2.42085540954e-11	+5.4910285455629803e+00	84	+2.42085540954e-11	-2.42535698648812614e+00
85	+1.81636313821e-11	+5.49102854557650577e+00	85	-1.81636313821e-11	-2.42535698650628984e+00
86	+1.362125193694e-11	+5.49102854558416631e+00	86	+1.362125193694e-11	-2.42535698649266873e+00
87	+1.021486139055e-11		87	-1.021486139055e-11	-2.42535698650288367e+00
88	+7.660356769436e-12		88	+7.660356769436e-12	-2.42535698649522313e+00
Результат: n: 88 Sn: 5.49102854558416631e+00 Последний Un: 7.6603567694357588181321e-12			Результат: n: 88 Sn: -2.42535698649522313e+00 Последний Un: 7.6603567694357588181321e-12		

n	Un	Sn	n	Un	Sn
1	+3.000000000000e+00	+3.0000000000000000e+00	1	-3.000000000000e+00	-3.0000000000000000e+00
2	+8.693181818182e-01	+8.6931818181818166e+00	2	+8.693181818182e-01	-2.130681818181834e+00
3	+4.821428571429e-01	+4.35146103896103931e+00	3	-4.821428571429e-01	-2.61282467532467555e+00
4	+3.164062500000e-01	+4.66786728896103931e+00	4	+3.164062500000e-01	-2.29641842532467555e+00
5	+2.207485465116e-01	+4.88861583547266765e+00	5	-2.207485465116e-01	-2.51716697183630345e+00
6	+1.583588367372e-01	+5.04697467220987228e+00	6	+1.583588367372e-01	-2.35880813509909881e+00
7	+1.152815385298e-01	+5.16225621073970231e+00	7	-1.152815385298e-01	-2.47408967362892840e+00
8	+8.464482516178e-02	+5.24690103590148471e+00	8	+8.464482516178e-02	-2.38944484846714600e+00
9	+6.248658455458e-02	+5.30938762045606705e+00	9	-6.248658455458e-02	-2.45193143302172878e+00
10	+4.629547705226e-02	+5.35568309750832938e+00	10	+4.629547705226e-02	-2.4056359559646689e+00
11	+3.438612845092e-02	+5.39006922595924642e+00	11	-3.438612845092e-02	-2.44002208442038393e+00
12	+2.558699784570e-02	+5.41565622380494371e+00	12	+2.558699784570e-02	-2.41443508657468664e+00
13	+1.906539055955e-02	+5.43472161436489021e+00	13	-1.906539055955e-02	-2.43350047713463358e+00
14	+1.422078488833e-02	+5.44894239925322310e+00	14	+1.422078488833e-02	-2.41927969224630024e+00
15	+1.061583351274e-02	+5.45955823276596508e+00	15	-1.061583351274e-02	-2.42989552575904266e+00
16	+7.929853142926e-03	+5.46748808590889190e+00	16	+7.929853142926e-03	-2.42196567261611628e+00
17	+5.926562048357e-03	+5.47341464795724875e+00	17	-5.926562048357e-03	-2.42789223466447131e+00
18	+4.431249220472e-03	+5.47784589717771997e+00	18	+4.431249220472e-03	-2.42346098544400146e+00
19	+3.314388654717e-03	+5.48116028583243686e+00	19	-3.314388654717e-03	-2.42677537408971835e+00
20	+2.479760146928e-03	+5.48364004597936461e+00	20	+2.479760146928e-03	-2.42429561395179061e+00
21	+1.855773849106e-03	+5.48549581982847023e+00	21	-1.855773849106e-03	-2.42615138780089667e+00
22	+1.389100152166e-03	+5.48688491998063643e+00	22	+1.389100152166e-03	-2.42476228764873047e+00
23	+1.039973251214e-03	+5.48792489323185073e+00	23	-1.039973251214e-03	-2.42580226089994477e+00
24	+7.787179059561e-04	+5.48870361113780714e+00	24	+7.787179059561e-04	-2.42502354299398881e+00
25	+5.831746501856e-04	+5.48928678578799278e+00	25	-5.831746501856e-04	-2.42560671764417446e+00
26	+4.367874525455e-04	+5.48972357324053828e+00	26	+4.367874525455e-04	-2.425169930116262896e+00
27	+3.271812707748e-04	+5.49005075451131308e+00	27	-3.271812707748e-04	-2.42549711146240377e+00
28	+2.451027326306e-04	+5.49029585724394398e+00	28	+2.451027326306e-04	-2.42525200872977331e+00
29	+1.836304754728e-04	+5.49047948771941652e+00	29	-1.836304754728e-04	-2.42543563920524630e+00
30	+1.375860307348e-04	+5.49061707375015118e+00	30	+1.375860307348e-04	-2.42529805317451164e+00
31	+1.030940317986e-04	+5.49072016778194971e+00	31	-1.030940317986e-04	-2.42540114720631017e+00
32	+7.725371505683e-05	+5.49079742149700678e+00	32	+7.725371505683e-05	-2.42532389349125355e+00
33	+5.789343652382e-05	+5.49085531493353063e+00	33	-5.789343652382e-05	-2.42538178692777739e+00
34	+4.338715275743e-05	+5.49089870208628827e+00	34	+4.338715275743e-05	-2.42533839977501975e+00
35	+3.251717909326e-05	+5.49093121926538164e+00	35	-3.251717909326e-05	-2.42537091695411311e+00
36	+2.437152596865e-05	+5.49095559079135054e+00	36	+2.437152596865e-05	-2.42534654542814465e+00
37	+1.826708220357e-05	+5.49097385787355385e+00	37	-1.826708220357e-05	-2.42536481251034841e+00
38	+1.369212544463e-05	+5.49098754999899885e+00	38	+1.369212544463e-05	-2.42535112038490386e+00
39	+1.026328887810e-05	+5.49099781328787717e+00	39	-1.026328887810e-05	-2.42536138367378173e+00
40	+7.693343687577e-06	+5.49100550663156461e+00	40	+7.693343687577e-06	-2.42535369033009429e+00
41	+5.767075344132e-06	+5.49101127370690900e+00	41	-5.767075344132e-06	-2.4253594574053823e+00
42	+4.323218009764e-06	+5.49101559692491836e+00	42	+4.323218009764e-06	-2.42535513418742843e+00
43	+3.240924132477e-06	+5.49101883784905098e+00	43	-3.240924132477e-06	-2.42535837511156105e+00
44	+2.429629669804e-06	+5.49102126747872088e+00	44	+2.429629669804e-06	-2.42535594548189115e+00
45	+1.821462061475e-06	+5.49102308894078206e+00	45	-1.821462061475e-06	-2.42535776694395233e+00
46	+1.365552515793e-06	+5.49102445449329757e+00	46	+1.365552515793e-06	-2.42535640139143638e+00
47	+1.023774636389e-06	+5.49102547826793419e+00	47	-1.023774636389e-06	-2.42535742516607256e+00
48	+7.675514696852e-07	+5.49102624581940368e+00	48	+7.675514696852e-07	-2.42535665761460306e+00
49	+5.754629589801e-07	+5.49102682128236275e+00	49	-5.754629589801e-07	-2.42535723307756121e+00
50	+4.314530536478e-07	+5.49102725273541648e+00	50	+4.314530536478e-07	-2.42535680162450840e+00
51	+3.234861119817e-07	+5.49102757622152815e+00	51	-3.234861119817e-07	-2.42535712511062052e+00
52	+2.425399585344e-07	+5.49102781876148693e+00	52	+2.425399585344e-07	-2.42535688257066218e+00
53	+1.818512107172e-07	+5.49102800061269747e+00	53	-1.818512107172e-07	-2.42535706442187271e+00
54	+1.363496513665e-07	+5.49102813696234904e+00	54	+1.363496513665e-07	-2.42535692807222114e+00
55	+1.023240758061e-07	+5.49102823919662519e+00	55	-1.023240758061e-07	-2.42535703030649685e+00
56	+7.665551710805e-08	+5.49102831585214268e+00	56	+7.665551710805e-08	-2.42535695365097981e+00
57	+5.747705005474e-08	+5.49102837332919247e+00	57	-5.747705005474e-08	-2.42535701112803004e+00
58	+4.309724020164e-08	+5.49102841642643291e+00	58	+4.309724020164e-08	-2.4253569803979004e+00
59	+3.231529914531e-08	+5.491028444874173179e+00	59	-3.231529914531e-08	-2.42535700034608936e+00
60	+2.423094982443e-08	+5.49102847297268148e+00	60	+2.423094982443e-08	-2.42535697611513967e+00
61	+1.816921038617e-08	+5.49102849114189162e+00	61	-1.816921038617e-08	-2.42535699428435025e+00
62	+1.362400703401e-08	+5.49102850476589843e+00	62	+1.362400703401e-08	-2.4253569806034300e+00
63	+1.021590152070e-08	+5.49102851498180033e+00	63	-1.021590152070e-08	-2.42535699087624446e+00
64	+7.660399562714e-09	+5.49102852264219976e+00	64	+7.660399562714e-09	-2.42535698321584503e+00
65	+5.744191327268e-09	+5.49102852838639066e+00	65	-5.744191327268e-09	-2.4253569896003638e+00
66	+4.307338384076e-09	+5.49102853269372915e+00	66	+4.307338384076e-09	-2.42535698465269789e+00
67	+3.229918656610e-09	+5.49102853592364770e+00	67	-3.229918656610e-09	-2.42535698788261644e+00
68	+2.422013528056e-09	+5.49102853834566140e+00	68	+2.422013528056e-09	-2.42535698540600274e+00
69	+1.816200634278e-09	+5.49102854016186193e+00	69	-1.816200634278e-09	-2.42535698727680327e+00
70	+1.361925212863e-09	+5.491028545152378672e+00	70	+1.361925212863e-09	-2.42535698591487803e+00
71	+1.021279890481e-09	+5.49102854254506667e+00	71	-1.021279890481e-09	-2.42535698693165798e+00
72	+7.658404402115e-10	+5.49102854331090739e+00	72	+7.658404402115e-10	-2.42535698617031770e+00
73	+5.742932617873e-10	+5.49102854388520090e+00	73	-5.742932617873e-10	-2.42535698674461075e+00
74	+4.306564701681e-10	+5.49102854431585730e+00	74	+4.306564701681e-10	-2.42535698631395435e+00
75	+3.229460577756e-10	+5.49102854463880341e+00	75	-3.229460577756e-10	-2.42535698663690047e+00
76	+2.421757662929e-10	+5.49102854488097947e+00	76	+2.421757662929e-10	-2.42535698639472486e+00
77	+1.816071715437e-10	+5.49102854506258620e+00	77	-1.816071715437e-10	-2.42535698657633203e+00
78	+1.361873782537e-10	+5.49102854519877326e+00	78	+1.361873782537e-10	-2.42535698644014452e+00
79	+1.021273861189e-10	+5.49102854530090045e+00	79	-1.021273861189e-10	-2.42535698654227172e+00
80	+7.658593323214e-11	+5.49102854537748630e+00	80	+7.658593323214e-11	-2.42535698646568587e+00
81	+5.743242861713e-11	+5.49102854543491858e+00	81	-5.743242861713e-11	-2.42535698652311815e+00
82	+4.306918789218e-11	+5.49102854547798813e+00	82	+4.306918789218e-11	-2.42535698648004905e+00
83	+3.229813635172e-11	+5.49102854551028585e+00	83	-3.229813635172e-11	-2.42535698651234721e+00
84	+2.422085540954e-11	+5.49102854553450648e+00	84	+2.422085540954e-11	-2.42535698648812614e+00
85	+1.816363133821e-11	+5.49102854555266973e+00	85	-1.816363133821e-11	-2.42535698650628984e+00
86	+1.362125193694e-11	+5.49102854556629083e+00	86	+1.362125193694e-11	-2.42535698649266873e+00
87	+1.021486139055e-11	+5.49102854557650577e+00	87	-1.021486139055e-11	-2.42535698650288367e+00
88	+7.660356769436e-12	+5.49102854558416631e+00	88	+7.660356769436e-12	-2.42535698649522313e+00

Результат:

n: 88

Sn: 5.49102854558416631e+00

Последний Un: 7.6603567694357588181321e-12

Результат:

n: 88

Sn: -2.42535698649522313e+00

Последний Un: 7.6603567694357588181321e-12

Вывод

На основе работы было изучено цикл `do {} while` закреплено ветвление алгоритма и применено на практике для разрешения задачи нахождения суммы конечного числа элементов от заданной точности. А также было проведено первичное ознакомление с библиотекой `fstream`.