

Задача 3.1 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \exp((1/61)/x^3 + 4/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"  \");
print("Backslash: \\  ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.2 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \sinh((2/61)/x^3 + 4/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"    \");
print("Backslash: \\    ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.3 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \cos((3/61)/x^3 + 4/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"  \");
print("Backslash: \\  ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.4 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \cosh((4/61)/x^3 + 4/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"  \");
print("Backslash: \\  ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.5 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \exp((5/61)/x^3 + 4/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"  \");
print("Backslash: \\  ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.6 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \exp((1/61)/x^3 - 5/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"    \");
print("Backslash: \\    ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.7 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \sinh((2/61)/x^3 - 5/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writetfile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"  \");
print("Backslash: \\  ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.8 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \cos((3/61)/x^3 - 5/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"  \");
print("Backslash: \\  ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.9 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \cosh((4/61)/x^3 - 5/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"    \");
print("Backslash: \\    ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.10 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \exp((5/61)/x^3 - 5/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"    \");
print("Backslash: \\    ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.11 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \exp((1/61)/x^3 + 5/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"  \");
print("Backslash: \\  ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).

Задача 3.12 Выдача в формате программы на plain “C”

Написать программу, которая создает файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять две функции:

`double f(double x)` — исходная функция;

`double s(double x)` — ряд;

и сравнивать результат разложения функции

$$f(x) = \sinh((2/61)/x^3 + 5/x^2 + 2/x)$$

в ряд по степеням $1/x$ до 20 порядка включительно:

$$s(x) = \sum_{n=0}^{20} c_n (1/x)^n,$$

с самой функцией, т.е. печатать таблицу:

x_n	$f(x_n)$	$s(x_n)$	$f(x_n) - s(x_n)$
-------	----------	----------	-------------------

где $x_n = 0.9, 1.0, 1.1, 1.2, \dots, 4.9, 5.0$.

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writefile` почему-то делает `append`, а не `overwrite` существующего файла. Регулярный и не зависящий от версии способ выводить в файл (кстати, вывод будет идти без излишней в данном случае служебной информации) — это функция `with_stdout` в сочетании с флагом `file_output_append`:

```
file_output_append:false;
with_stdout("myfile.out", print("line 1"));
file_output_append:true;
with_stdout("myfile.out", print("line 2"));
with_stdout("myfile.out", print("line 3"));
with_stdout("myfile.out", print("line 4"));
```

(здесь первые 2 строки гарантированно затирают предыдущее содержимое файла и выводят в файл "line 1", а следующие строки позволяют дописать "line2", "line3", и т.д.)

Печать двойной кавычки и обратного слеша производится обычным образом:

```
print("Double quote: \"    \");
print("Backslash: \\    ");
```

Не мучайте компилятор gcc излишне длинными выражениями (“x”, умноженный на себя 20 раз — это ужасно, функция `pow(x, 20)` — это безграмотно, и сумма ряда, записанная в явном виде, а не в виде цикла — это ужасно, присвоения значений коэффициентов ряда при каждом вызове функции — это безграмотно, отдельное вычисление x^{11} , притом что шаг назад был сосчитано x^{10} — это ужасно, и т.д.).