

Задача 3.1 Выдача в формате программы на plain “C”

Написать программу, которая находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \exp\left(2x + 4x^2 + \frac{x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writelfile` 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 находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \sinh\left(2x + 4x^2 + \frac{2x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

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 находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \cos\left(2x + 4x^2 + \frac{3x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

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 находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \sin\left(2x + 4x^2 + \frac{4x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

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 находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \cosh\left(2x + 4x^2 + \frac{5x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

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 находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \exp\left(2x - 4x^2 + \frac{x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

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 находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \sinh\left(2x - 4x^2 + \frac{2x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

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 3.8 Выдача в формате программы на plain “C”

Написать программу, которая находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \cos\left(2x - 4x^2 + \frac{3x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

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 находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \sin\left(2x - 4x^2 + \frac{4x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

Функции вывода в файл в разных версиях МАХИМ’ы работают по-разному. Например, в текущей версии функция `writelfile` 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 находит явный вид отрезка ряда Тейлора длиной в 21 член ($0 \dots 20$) для функции $f(x) = \cosh\left(2x - 4x^2 + \frac{5x^3}{61}\right)$.

После этого она должна создать файл, который является программой на языке “plain C”, работающей под gcc. Ручная правка полученного файла не допускается. Программа должна определять 2 функции:

```
double f(double x) — исходная функция;
double t(double x) — ряд Тейлора.
```

Она должна сравнивать точное значение функции и ответ для ряда Тейлора на наборе точек $x_i = -1.0 + 0.05 * (i)$ при $i = 0 \dots 40$. Выходной файл, который создает программа на “C”, должен содержать таблицу

x_i	$f(x_i)$	$t(x_i)$	$f(x_i) - t(x_i)$
-------	----------	----------	-------------------

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.).