

Задачи для 1 курса, часть 2

1. Простейшие вычислительные задачи

Программы решения всех задач должны содержать: файл с функциями, реализующими предлагаемый алгоритм, файл с тестовой процедурой для проверки алгоритма, файл с функциями формирования исходных данных и вызова серии тестов.

Упр.1. Для заданного N вычислить суммы в прямом и обратном порядке. При каких N получается максимальное расхождение в этих двух значениях?

$$\sum_{k=1}^N \frac{1}{\sqrt{k}}, \quad \sum_{k=1}^N \frac{\ln k}{k}, \quad \sum_{k=1}^N \frac{1}{k}, \quad \sum_{k=1}^N \frac{1}{k^2}, \quad \sum_{k=1}^N \frac{1}{k!}$$

Что изменится, если слагаемые умножить еще на $(-1)^k$?

Упр.2. Написать программу, определяющую машинную точность, т.е. максимальное $a > 0$, такое что $1 + a = 1$. Усложненный вариант: найти два ближайших к единице представимых вещественных числа.

В следующих задачах требуется оформить решение в виде функции, получающей в качестве параметра указатель на функцию `double (*f)(double x)`

1. Вычислить корень уравнения $f(x) = 0$ на данном отрезке с заданной точностью ϵ

- 1.1. методом деления пополам;
- 1.2. методом Ньютона (касательных);
- 1.3. методом хорд (секущих).

Написать тест, в котором вывести количество итераций и сравнить результат с точным ответом.

2. Вычислить интеграл от данной функции $f(x)$ на данном отрезке $[a, b]$ с заданной точностью ϵ по составной формуле

- 2.1. прямоугольников;
- 2.2. трапеций;
- 2.3. Симпсона;
- 2.4. Гаусса (двухточечной).

Использовать алгоритм удвоения сетки и алгоритм автоматического выбора шага. Сравнить с точным ответом.

3. Вычислить минимальное значение данной функции $f(x)$ с заданной точностью ϵ методом построения параболы с поиском трех точек с помощью

- 3.1. измельчения геометрической прогрессии;
- 3.2. метода золотого сечения;
- 3.3. измельчением равномерной сетки.

4. Написать программу вычисления значения одной из элементарных функций (`sin`, `cos`, `exp`, `log`) в заданной точке x и с заданной точностью ϵ суммированием ряда Тейлора. Сравнить с точным ответом

5. Написать программу, которая по аргументам $x_1 \dots x_n$ и значениям функции $y_1 \dots y_n$ вычисляет приближенные значения данной функции на заданной сетке $\bar{x}_1 \dots \bar{x}_n$ с помощью

- 5.1. интерполяционного многочлена Лагранжа;
- 5.2. кусочно-линейной интерполяции;
- 5.3. кусочно-квадратичной интерполяции;
- 5.4. Эрмитовой интерполяции (по $f(x_i)$ и $f'(x_i)$);
- 5.5. наилучшего среднеквадратического приближения линейной функцией (здесь надо получить уравнение прямой).

Сравнить с точными значениями функции. Нарисовать исходные точки и “график” приближенной функции (Gnuplot и т.п.)

2. Задачи на работу с матрицами, метод Гаусса и итерационные алгоритмы

Формирование данных задачи (матрицы, правой части системы) выполняется: а) вводом из файла; б) вычислением (генерацией) по заданным формулам или случайно.

При выполнении задач из этого раздела нужно освоить два способа представления матрицы:

- а) как единого массива для всех элементов матрицы (функция получает аргумент `double *matr`)
- б) как набора массивов-строк с массивом указателей на каждую строку. (функция получает аргумент `double **matr`)

0. Реализовать функции умножения матрицы на вектор, умножения двух прямоугольных матриц, функции для создания и вывода матриц.

1. Определить ранг вещественной $N \times M$ матрицы.
2. Найти определитель вещественной $N \times N$ матрицы.
3. Для квадратной матрицы вычислить обратную.
4. Решить систему линейных уравнений методом исключения Гаусса.
5. Решить систему линейных уравнений методом простой итерации.
6. Решить систему линейных уравнений итерационным методом Зейделя.
7. Решить систему линейных уравнений итерационным методом скорейшего спуска.

3. Обработка текстового файла

В следующих задачах “словом” называется последовательность символов в пределах одной строки, не содержащая символов из заранее заданного набора (например, “.,;:!?()[]” и т.п.) При решении задач рекомендуется пользоваться стандартными функциями `strcmp`, `strcpy`, `strstr`, `strcat`, `strtok` и др. (см. `string.h`) или классом `string`. Результатом работы программы должен быть новый, преобразованный файл.

0. Реализовать стандартные функции работы со строками `strcmp`, `strcpy`, `strstr`, `strcat`, `strset` и др.

1. Реализовать программу перекодировки текстового файла из одной заданной кодировки в другую, например, `cp1251` в `utf-8`.

2. Заменить в файле каждую последовательность заданных одинаковых символов на один такой символ.

3. Заменить всюду в файле один заданный набор символов на другой (с учетом разницы в их длине).

4. Вывести номера строк исходного файла и номер позиции в строке, где встречается заданное слово.

5. Определить количество, максимальную, минимальную и среднюю длину слов из данного файла а также частоту (процент) появления каждого символа.

6. Вывести все слова из данного файла в алфавитном порядке без повторов.

7. Файл В получен из текстового файла А вставкой и/или удалением некоторого малого количества строк (например, не больше 10). Сравнить эти файлы и определить какие строки и где были добавлены или удалены в файле А.

8. Реализовать ввод табличных данных из файла, записанного в соответствии с заданным текстово-числовым форматом, где строки и столбцы таблицы размечены некоторыми ключевыми словами, а “индексы” чисел в таблице определяются их позиционированием в строках текста.

9. Разрезать “длинные” строки в файле по пробелам на более короткие (не более заданной длины).

10. Удалить из файла часть текста между двумя “скобками”, где скобка — это заданный набор символов. Например, убрать из файла комментарии в стиле C.

11. Реализовать инструкцию типа `#include`, т.е. вставить содержимое файла `filename` в то место файла, где встречается строка `#include filename`.

12. Реализовать инструкции типа `#define` и `#undef`, т.е. выполнить указанные подстановки в области их задания.

13. Реализовать инструкции типа `#ifdef` - `#else` - `#endif`, т.е. оставить в файле требуемый текст в зависимости от условия.

14. Отформатировать абзацы текста в заданных границах и с красной строкой (без переноса слов). Абзац — фрагмент текста между пустыми строками.

Продолжение следует ...