

Задания по программированию студентам группы 6113 ММФ НГУ

Александр Геннадьевич Фенстер
fenster@fenster.name http://6113.fenster.name

Для получения зачёта каждый студент должен выполнить 15 задач, список которых находится ниже, при этом последняя, пятнадцатая, задача является индивидуальной. Все программы должны быть написаны на языке C. На практических занятиях будет показано, как пользоваться средой программирования Microsoft Visual Studio 6.0, но программы можно писать и компилировать любым другим компилятором.

Задание 1. Простые программы с циклами

Задание состоит из трёх «тренировочных» задач на освоение основных инструкций языка C: циклов `for` и `while` и ветвлений. Предполагается, что все три задачи будут сданы в течение первых двух-трёх недель.

Задача 1. Числа Фибоначчи. Вычислить первые N членов ряда Фибоначчи:

$$f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-2}.$$

Задача 2. Уравнение. Пусть $f : \mathbb{R} \rightarrow \mathbb{R}$ и известно, что $f(x)$ возрастает на отрезке $[a, b]$. Найти приближённое решение уравнения $f(x) = 0$ с точностью до 0,001 на этом отрезке или сообщить, что решения нет. Для решения задачи определить в программе функцию `float f(float x)` и переменные `a` и `b`, например, так:

```
float a = 0;
float b = 2;

float f(float x)
{
    return (x * x - 1);
}
```

В этом примере уравнение имеет вид

$$x^2 - 1 = 0, \quad x \in [0, 2],$$

его решение: $x = 1$.

Задача 3. Интеграл. В программе задана непрерывная на отрезке $[a, b]$ функция $f : \mathbb{R} \rightarrow \mathbb{R}$. Вычислить приближённое значение интеграла

$$\int_a^b f(x) dx.$$

Задание 2. Работа с массивами

Задание содержит три задачи на работу с одномерными и двумерными массивами. Предполагается, что все эти задачи будут сданы до середины марта. Во второй и третьей задаче можно считать, что N не превышает, например, 100, ввести символ

```
#define NMAX 100
```

и определять массивы как `float A[NMAX][NMAX]` (и т. п.)

Задача 1. Циклический сдвиг. Ввести с клавиатуры массив из 10 элементов и число k . Выполнить циклический сдвиг элементов массива на k элементов вправо, не используя дополнительный массив.

Пример ввода:

```
1 3 5 7 9 2 4 6 8 10
4
```

Пример вывода:

```
4 6 8 10 1 3 5 7 9 2
```

Задача 2. Умножение матриц. Перемножить две матрицы размера $N \times N$. Матрицы A и B вводятся с клавиатуры (для желающих — из файла) по строкам. Вывести на экран элементы матрицы $C = A \cdot B$:

$$C_{ij} = \sum_{k=1}^n A_{ik} \cdot B_{kj}.$$

Задача 3. Бинарный поиск. Дан массив, про который известно, что его элементы упорядочены в порядке неубывания. Ввести с клавиатуры несколько чисел и, используя метод деления пополам, определить, встречаются ли эти числа в массиве.

Бинарный поиск является, пожалуй, лучшим способом поиска элемента в упорядоченном массиве. В наихудшем случае для поиска элемента (или определения того, что число в массиве отсутствует) необходимо сделать всего $\lceil \log_2 N \rceil + 1$ операций.

Задание 3. Работа с файлами

Две представленных задачи необходимо сдать до конца марта.

Задача 1. Количество символов. Подсчитать и вывести на экран количество строк, слов и символов в текстовом файле `input.txt`.

Аналогичная утилита в операционной системе UNIX называется `wc` (сокращение от word count).

Задача 2. Количество вхождений букв. Подсчитать количество вхождений букв латинского алфавита в текстовом файле `input.txt`. Результат в виде

```
a 5
b 4
...
z 0
```

выдать в файл `output.txt`.

Задание 4. Работа со строками

Умение работать со строками является весьма важным при программировании на C. В трёх следующих задачах необходимо ввести строку с клавиатуры (можно считать, что максимальная длина строки не превышает 100 символов) и выполнить с ней некоторые действия. Срок сдачи задания — конец марта, начало апреля.

Задача 1. Палиндром. Проверить, что введённая с клавиатуры строка является палиндромом, т.е. читается одинаково слева направо и справа налево (без учёта пробелов и знаков препинания). Не использовать дополнительную строку.

Пример строки, являющейся палиндромом:

Он дивен, палиндром, и ни морд, ни лап не видно...

Задача 2. Сумма чисел. В строке записана последовательность натуральных чисел, разделённых нецифровыми символами: буквами, знаками препинания, пробелами и т.д. Найти сумму чисел последовательности.

Пример текста:

В 12 часов 35 минут термометр показывал 23 градуса ниже нуля.

Ответ: 70.

Задача 3. Алфавит. Необходимо проверить, содержит ли данная строка все символы русского алфавита. Такие строки используются, например, программой `fontview` для того, чтобы показать, как будут в данном шрифте выглядеть все русские буквы.

Пример строк:

Съешь ещё этих вкусных французских булок, да выпей же чаю!

Ответ: да.

Съешь ещё этих вкусных французских булок, да выпей чаю!

Ответ: не хватает буквы «ж».

Задание 5. Алгоритмы сортировки

В течение апреля необходимо будет разобраться с одним из улучшенных алгоритмов сортировки, который в состоянии отсортировать массив длины N за время $O(N \log_2 N)$, а не за $O(N^2)$, как «простые» алгоритмы.

Задача 1. Сортировка. Реализовать быструю или пирамидальную сортировку (на ваш выбор). В обоих случаях, программа должна брать входные данные из файла `input.txt`. В первой строке этого файла записано число N — количество чисел, которые необходимо отсортировать (известно, что $1 \leq N \leq 1000$). Далее в файле записаны N чисел типа `int`. В файл `output.txt` необходимо выдать эти N чисел в неубывающем порядке.

Теория по этим алгоритмам сортировки представлена в отдельном файле.

Пример файла `input.txt`:

```
5
3 1 5 4 2
```

Пример файла `output.txt`:

```
1 2 3 4 5
```

Задание 6. Работа со списками

Динамические списки применяются повсеместно для хранения данных, размер которых заранее неизвестен. В данном задании необходимо будет научиться работать с односвязными списками (в которых каждый элемент хранит указатель на следующий элемент списка). Срок сдачи задания — первая половина мая.

Задача 1. Сортировка вставкой в список. Отсортировать последовательность чисел из файла путём вставки их в упорядоченный список (список изначально пуст). Длина последовательности заранее неизвестна.

Пример чтения чисел из файла до конца файла:

```
FILE *f = fopen("input.txt", "r");
int a;
while (fscanf(f, "%d", &a) == 1) {
    /* do something */
}
fclose(f);
```

Функция `fscanf()` в нормальном случае возвращает количество прочитанных аргументов. Если она вернула не 1, делается вывод о том, что либо достигнут конец файла, либо произошла ошибка (например, в файле записано не число), и цикл завершается.

Пример описания списка:

```
struct item {
    int data;
    struct item *next;
};
struct item *head = 0;
```

Задача 2. Количество вхождений. Подсчитать количество вхождений каждого слова в текст. Текст читается из файла `input.txt`, для простоты можно считать, что словом является любая последовательность латинских букв, а всё, что не является латинской буквой, считается разделителем. Максимальная длина слова равна 10 символам.

Можно определить следующую структуру:

```
struct item {
    char word[11];
    int count;
    struct item *next;
};
```

Таким образом, будет храниться список слов, причём для каждого слова хранится количество вхождений его в текст.

Задание 7. Индивидуальное задание на графы

Индивидуальные задания будут распределены в начале апреля. Каждый студент получит одну задачу на графы. Срок сдачи задания — начало зачётной недели.