

Обязательные задания

1. Простые программы с циклами

Задача 1. Вычислить первые N членов ряда Фибоначчи:

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

Задача 2. В программе определена функция `double f(double x)`, вычисляющая некоторую функцию, неубывающую на отрезке $[a, b]$. Найти приближённое решение уравнения $f(x) = 0$ на отрезке $[a, b]$.

Пример функции:

```
int a = 0;
int b = 2;

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

т.е. уравнение имеет вид

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

его решение: $x = 1$.

2. Работа с массивами

Задача 1. Ввести с клавиатуры массив из 10 элементов и число k . Выполнить циклический сдвиг элементов массива на k элементов вправо. 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. Перемножить две матрицы размера $N \times N$.

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.

3. Работа с файлами

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

Задача 2. 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`.

4. Работа со строками

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

Пример строки, являющейся палиндромом:

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

Задача 2. 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.

5. Алгоритмы сортировки

Задача 1. 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

Задача 1. 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. 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

Все задачи в данном задании взяты из книги В.Н.Касьянова «Курс программирования на Паскале в задачах и упражнениях» (Новосибирск, НГУ, 2001).

1. Абылхожин Руслан

Проверить, содержит ли данный связный неориентированный граф хотя бы одну *точку сочленения*, т.е. вершину, при удалении которой из графа в графе увеличивается количество компонент связности.

2. Баскаков Иван

Даны неориентированный граф и число k . Найти все вершины графа, расстояние от которых до первой вершины графа равно k .

3. Востриков Вячеслав

Для данного связного неориентированного графа найти *диаметр графа*, т.е.

$$\max_{(i,j) \in E} D(i, j)$$

($D(i, j)$ — длина кратчайшего пути между вершинами i и j).

4. Давлетбаев Максим

В данном связном неориентированном графе найти длины кратчайших путей между всеми парами вершин.

5. Иванченко Вадим

Даны неориентированный граф и число k . Выяснить, есть ли в данном графе вершина, расстояние от которой до каждой вершины не превышает k .

6. Карпенко Анна

Дан связный неориентированный *взвешенный граф* (т.е. каждое ребро графа имеет неотрицательный *вес*, который можно условно считать расстоянием между соответствующими вершинами). Найти вершину графа, сумма длин кратчайших путей от которой до всех остальных вершин графа минимальна.

7. Лупинос Ольга

Найти все вершины данного неориентированного графа, недостижимые из первой вершины.

8. Морозова Мария

Подсчитать количество компонент связности в данном неориентированном графе.

9. Мочалов Родион

Дан связный неориентированный граф. Найти *медиану графа*, т.е. такую его вершину, что сумма расстояний от неё до остальных вершин минимальна.

10. Пичуева Анна

Дан связный неориентированный *взвешенный граф* (т.е. каждое ребро графа имеет неотрицательный *вес*, который можно условно считать расстоянием между соответствующими вершинами). Найти кратчайшие пути от первой вершины до всех остальных вершин графа.

11. Сапрыкина Софья

Правильной раскраской вершин графа в k цветов называется такое отображение $c : V \rightarrow \{1 \dots k\}$, что

$$c(i) \neq c(j) \text{ для любого ребра } (i, j) \in E.$$

Выяснить, можно ли раскрасить данный неориентированный граф в 2 цвета, и вывести эту раскраску, если можно.

12. Соловьёв Владимир

Определить, существует ли в данном связном неориентированном графе гамильтонов цикл, т.е. цикл, проходящий через каждую вершину графа ровно один раз.

13. Толстик Виктор

Дан связный неориентированный граф. Определить, содержит ли он хотя бы один *мост*, т.е. ребро, при удалении которого в графе увеличивается количество компонент связности.

Дополнительные задания

Данные задания являются дополнительным, выполнение их не обязательно, но принесёт некие абстрактные бонусы.

1. Функции с переменным числом параметров

Задача 1. Реализовать свой упрощённый вариант функции `printf` (назовём её, например, `myprintf`) с использованием заголовочного файла `stdarg.h`. Строка формата может содержать директивы `%d`, `%c`, `%s`. Функция должна иметь следующий заголовок:

```
int myprintf(char *fmt, ...);
```

и возвращать количество символов, которые были выведены на печать. Для вывода использовать только функцию `int putchar(int c)`, выводящую один символ `c` в поток `stdout`.

Задача 2. Реализовать свой упрощённый вариант функции `scanf` (назовём её, например, `myscanf`) с использованием заголовочного файла `stdarg.h`. Строка формата может содержать директивы `%d`, `%c`, `%s`. Функция должна иметь следующий заголовок:

```
int myscanf(char *fmt, ...);
```

и возвращать количество реально прочитанных аргументов либо `EOF` в случае достижения конца входного потока. Для ввода использовать только функцию `int getchar()`, читающую один символ из потока `stdin`.

2. Хэш-таблицы

Задача 1. Количество слов. Подсчитать количество вхождений слов в текст. Слова состоят из латинских букв и разделяются любыми символами, не являющимися латинскими буквами. Для ускорения работы программы использовать хэш-таблицу для хранения словаря.

Хэш-функцией называется функция

$$f : K \rightarrow \{0, \dots, N - 1\}, \quad N \in \mathbb{N},$$

сопоставляющая каждому элементу *множества ключей* K натуральное число от 0 до $N - 1$. В данном примере множеством K является множество всех слов из букв латинского алфавита.

При помощи хэш-функции каждому слову из текста сопоставляется хэш-код, по которому слово ищется в массиве из N элементов. Ситуация, при которой два разных слова имеют один и тот же код, называется *коллизией*.

Хэш-таблицей называется массив из N элементов, в i -м элементе которого хранится указатель на начало списка слов, хэш-код которых равен i .