

Лабораторная работа 4

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Разработайте пользовательскую программу для решения указанной задачи.

Входные данные должны передаваться в программу через аргументы командной строки. Имя программы совпадает с именем файла исходного кода. Выполните сборку пользовательских программ. Поместите созданную вами программу на виртуальный диск и протестируйте ее в ОС pintos. Разработайте не менее 10 тестовых наборов входных данных. Приведите в отчете входные данные, результаты запуска программы и ваши пояснения к ним.

Варианты заданий:

1. Программа должна находить НОД и НОК введенных целых чисел с помощью алгоритма Евклида. Количество чисел, для которых вычисляется НОД – от 2 до 6.

Аргументы: числа для вычисления НОД и НОК.

Файл реализации: examples/nodnok.c

2. Программа должна генерировать N псевдослучайных чисел в заданном числовом диапазоне. Использовать ЛКГ. В качестве начального значения ЛКГ использовать S.

Аргументы: нижняя граница, верхняя граница, N, S.

Файл реализации: examples/rndgen.c

3. Программа должна строить маршрут обхода конем шахматного поля (конь должен пройти через все поля доски по одному разу) – находить любое решение.

Аргументы: размер поля, начальное положение коня.

Файл реализации: examples/knight.c

4. Программа должна строить последовательность ходов для решения задачи о ханойских башнях.

Аргументы: число дисков.

Файл реализации: examples/hanoi.c

5. Программа должна реализовывать перевод числа из одной системы счисления в другую.

Поддерживаемые системы счисления: двоичная, десятичная, восьмеричная, шестнадцатеричная.

Аргументы: число, исходная система счисления, требуемая система счисления.

Файл реализации: examples/snconv.c

6. Программа должна сортировать заданные целые числа в порядке возрастания с использованием одного из алгоритмов: quicksort, bubblesort.

Аргументы: алгоритм сортировки (quick или bubble), целые числа — каждое число — отдельный аргумент командной строки.

Пример запуска: run "sortdig quick 5 6 3 10 -34 17 90 904 14 123 754 -34"

Файл реализации: examples/sortdig.c

7. Программа должна определять, можно ли из заданной последовательности целых чисел X получить заданную последовательность Y путем вычеркивания элементов из X.

Аргументы: последовательность X (целые числа, каждое число — отдельный аргумент командной строки), разделитель: "@", последовательность Y (числа, каждое число — отдельный аргумент командной строки).

Пример запуска: run "checkseq 5 4 3 6 8 @ 5 8 6"

Файл реализации: examples/checkseq.c

8. Программа должна генерировать все возможные k -элементные подмножества множества A из N чисел, $A = \{1, 2, \dots, N\}$.

Аргументы: k, N.

Пример запуска: run "subsets 2 4"

Ответ: {1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}.

Файл реализации: examples/subsets.c

9. Программа должна представлять натуральное число $N > 1$ в виде суммы натуральных чисел так, чтобы произведение этих слагаемых было максимально. Задачу необходимо решать рекурсивным перебором.

Аргументы: N.

Файл реализации: examples/nsum.c

10. Программа должна определять минимальное число ходов, необходимых шахматному коню для перехода с одного поля на другое. Программа должна напечатать результат и найденные координаты клеток.

Аргументы: начальное поле, конечное поле.

Пример запуска: run "kjump e2 e4"

Файл реализации: examples/kjump.c

11. Программа должна расставлять N ферзей на шахматной доске $K \times K$ так, чтобы они не били друг друга — вывести любое решение.

Аргументы: N, K.

Файл реализации: examples/queen.c

12. Программа должна найти все простые числа в заданном числовом диапазоне (включительно) и вывести их. Обратите внимание на оптимальность и эффективность решения.

Аргументы: нижняя граница, верхняя граница.

Файл реализации: examples/primes.c