

Общая постановка задачи и варианты заданий

Общая постановка.

Требуется разработать два «консольных» Linux-приложения, которые обмениваются сообщениями посредством FIFO-файлов (именованных конвейеров). При реализации обязательно использование функций, предназначенных для работы с FIFO-файлами.

Программы должны быть устойчивыми к некорректному пользовательскому вводу, а сообщения об ошибках — информативными.

Варианты заданий.

Вариант 1. Клиент отправляет серверу два беззнаковых целых числа L и U , введенные пользователем, каждое из которых состоит в точности из 6 знаков. L — это нижняя граница диапазона, U — верхняя граница диапазона.

Сервер принимает значения границ диапазона, осуществляет поиск «счастливых билетов» из этого диапазона, когда а) сумма цифр на 1-3 позиции равна сумме цифр, стоящих на 4-6 позиции; б) сумма цифр на нечетных позициях равна сумме цифр на четных позициях; в) арифметический квадратный корень из суммы шести цифр номера билета есть целое число. На стандартный вывод должны передаваться полученные значения, а также количество найденных «счастливых чисел».

Вариант 2. Клиент отправляет серверу введенный пользователем номер k обобщенного числа Фибоначчи (ОЧФ) и его порядок p , причем $p, k > 0$. Формула для вычисления ОЧФ заданного порядка с указанным номером:

$$F_k^p = \begin{cases} \sum_{i=k-1-p}^{k-1} F_i^p, & \text{если } k > p+1 \\ 1, & \text{если } 1 \leq k \leq p+1 \end{cases}$$

Сервер принимает номер и порядок числа, вычисляет по ним ОЧВ с номером k и порядком числа p и передает его на стандартный вывод.

Вариант 3. Клиент отправляет серверу введенную пользователем строку, хранящую целое число со знаком.

Сервер принимает строку, хранящую целое число со знаком, и передает на стандартный вывод эквивалент этого числа прописью. Например, ввод «-1211» должен приводить к выводу «минус одна тысяча двести одиннадцать», а ввод «100000101» приводит к выводу «сто миллионов сто один». Ограничения накладываются только представлениями максимально и минимально возможного знакового целого числа используемого языка программирования.

Вариант 4. Клиент отправляет серверу введенную пользователем строку, хранящую вещественное число в обычном или экспоненциальном форматах.

Сервер принимает строку, хранящую вещественное число, и передает на стандартный вывод эквивалент этого числа прописью. Например, ввод «-12.11» должен приводить к выводу «минус двенадцать целых одиннадцать сотых», а ввод «1.2E+2» приводит к выводу «120 целых». Ограничения: 1) для обычного формата после «точки» не более двух знаков; 2) для экспоненциального формата после «точки» не более трех знаков; 3) до точки не более 12 знаков для обычного формата; 4) знак «точки» - это «.».

Вариант 5. Клиент отправляет серверу две строки, введенные пользователем.

Сервер принимает две строки. Если обе строки хранят целые числа со знаком, то на стандартный вывод передается сумма чисел. Если были переданы вещественные числа в обычном или экспоненциальном форматах, то на стандартный вывод передается остаток от деления первого числа на второе. Во всех остальных случаях выводится результат конкатенации двух полученных строк.

Вариант 6. Клиент отправляет серверу две квадратные матрицы одинакового размера с элементами десятичного типа с плавающей запятой. Его, а также сами элементы матриц должен вводить пользователь.

Сервер принимает две квадратные матрицы, а затем передает на стандартный вывод сумму матриц и значения определителей и их произведения.

Вариант 7. Клиент отправляет серверу, два введенных пользователем десятичных целых числа A и B , где A — сдвигаемое число, B — величина логического, арифметического и циклического сдвига.

Сервер принимает два числа, после этого осуществляется логический, арифметический и циклический сдвиг числа A на величину B (отрицательное значение — сдвиг вправо, неотрицательное - влево), и результаты выводятся на экран.

Вариант 8. Клиент принимает от пользователя некое слово, после чего проверяет его на корректность (отсутствие знаков препинания, цифр, смещения латинских и кириллических символов) и отправляет его серверу.

Сервер принимает слово, проверяет, является ли введенное слово палиндромом и передает на стандартный вывод результат проверки.

Вариант 9. Клиент отправляет серверу элементы одномерного массива строк, введенные пользователем, а также один или несколько символов, означающих способ преобразования массива (например, символ A – упорядочение по возрастанию, D – упорядочение по убыванию, U – разупорядочение).

Сервер принимает массив и способ его преобразования, выполняет его и передает результат на стандартный вывод. Для упорядочения воспользоваться любым из так называемых «улучшенных алгоритмов» сортировки массивов (при этом исключено использование следующих алгоритмов: *пузырьковая* сортировка, *шейкерная* сортировка, сортировка *прямым выбором* и сортировка *прямыми включениями/вставками*)¹.

Вариант 10. Клиент отправляет серверу введенную пользователем квадратную матрицу с элементами десятичного типа с плавающей точкой.

Сервер принимает матрицу, вычисляет обратную ей матрицу, а затем передает обе матрицы на стандартный вывод. Обеспечить проверку возможности вычисления обратной матрицы как на стороне клиента, так и на стороне сервера.

Вариант 11. Клиент отправляет серверу элементы введенной пользователем квадратной матрицы, а также два введенных им числа – номер столбца N и номер строки R .

Сервер принимает матрицу и номер столбца N и номер строки R , передает ее на стандартный вывод, затем «вычеркивает» из матрицы столбец с номером N и строку с номером R , передает измененную матрицу на стандартный вывод, вычисляет и передает на стандартный вывод значение ее определителя.

Вариант 12. Клиент принимает от пользователя две даты – строки вида ЦЦ.ЦЦ.ЦЦЦЦ, где Ц – это любая цифра из диапазона [0-9] и отправляет серверу.

Сервер принимает обе даты, вычисляет полное количество дней, прошедших между ними, и передает его на стандартный вывод. И клиент, и сервер должны принимать во внимание некорректные даты, например, 37.06.2006 или 01.18.2006, а также такие особенности Григорианского календаря, как переход на него в России в 1918 (отсутствие дней с 1 по 13 февраля в этот год). В качестве базовой даты принять 1 января 1601.

¹ http://ru.wikipedia.org/wiki/Алгоритм_сортировки

Вариант 13. Клиент принимает от пользователя два значения времени суток – строки вида ЦЦ.ЦЦ.ЦЦ, где Ц – это любая цифра из диапазона [0-9] и отправляет их серверу.

Сервер принимает обе строки, вычисляет полное количество секунд, прошедших между двумя значениями времени двумя способами, и передает их на стандартный вывод. И сервер, и клиент должны принимать во внимание некорректные значения, например, 28.00.06, или 01.99.20, или 08.08.65.

Вариант 14. Клиент отправляет серверу две строки, введенные пользователем.

Сервер принимает две строки, осуществляет поиск вхождения второй строки в первую любым известным методом, кроме прямого (*линейного, грубой силы*), и передает на стандартный вывод значение индекса элемента первой строки, с которого началось совпадение, или строку «Нет вхождений» в противном случае.

Вариант 15. Клиент отправляет серверу две строки, введенные пользователем.

Сервер принимает две строки, осуществляет поиск количества вхождений второй строки в первую любым известным методом, кроме прямого (*линейного, грубой силы*) поиска, и передает на стандартный вывод полученное значение.

Вариант 16. Клиент принимает от пользователя дату – строку вида ЦЦ.ЦЦ.ЦЦЦЦ, где Ц – это любая цифра из диапазона [0-9] и отправляет серверу.

Сервер принимает дату и передает на стандартный вывод число, месяц и год прописью (например, ввод «29.02.2012» приводит к выводу «Двадцать девятое февраля две тысячи двенадцатого года»). И сервер, и клиент должны принимать во внимание некорректные даты, например, 37.06.2013 или 01.18.2016, а также такие особенности Григорианского календаря, как переход на него в России в 1918 (отсутствие дней с 1 по 13 февраля в этот год). В качестве базовой даты принять 1 января 1601.

Вариант 17. Клиент принимает от пользователя значение времени суток – строку вида ЦЦ.ЦЦ.ЦЦ,Ц или ЦЦ.ЦЦ.ЦЦ.ЦЦ, где Ц – это любая цифра из диапазона [0-9] и отправляет серверу.

Сервер принимает значение времени суток и передает на стандартный вывод часы, минуты, секунды и их долей прописью (например, ввод «12.01.20,15» приводит к выводу «двенадцать часов одна минута двадцать и пятнадцать сотых секунды», для «00.01.12,3» - «ноль часов одна минута двенадцать и три десятых секунды»). И сервер, и клиент должны принимать во внимание некорректные значения, например, 28.00.06,14; 01.99.20,15; 08.08.65,16.

Вариант 18. Клиент принимает от пользователя беззнаковое целое число N – основание системы счисления ($1 < N < 17$ или $N = 1$) и последовательность цифр в соответствии с заданной системой счисления, отправляет серверу.

Сервер принимает основание системы счисления и число в этой системе, передает его на стандартный вывод, переводит его в десятичную систему, передает на стандартный вывод, дополняет его (инвертирует) до максимальной цифры в заданной системе счисления, передает на стандартный вывод значение инвертированной последовательности, переводит ее в число в десятичной системе и передает его на стандартный вывод. Для случая $N = 1$, когда используется унарная система счисления, результатом является только перевод в десятичную систему с передачей его на стандартный вывод.

Вариант 19. Клиент принимает от пользователя беззнаковое целое число N – основание системы счисления ($1 < N < 17$ или $N = 1$) и последовательность цифр в соответствии с заданной системой счисления, и отправляет серверу.

Сервер принимает основание системы счисления и число в этой системе, передает его на стандартный вывод, переводит его в десятичную систему счисления, передает его на стандартный вывод, осуществляет его реверс (меняет порядок следования знаков на обратный), передает на стандартный вывод значение измененной последовательности, переводит ее в число в десятичной системе и передает его на стандартный вывод. Для случая $N = 1$, когда используется унарная система счисления, результатом является только перевод в десятичную систему с передачей его на стандартный вывод.

Вариант 20. Клиент отправляет серверу три строки, введенных пользователем (первая и третья строки – это правильные рациональные или десятичные дроби вида «1/3» или «0,5», вторая строка – это знак арифметической операции вида «+», «-», «*», «/» либо операции сравнения «<», «>», «=», «!=», «>=», «<=»).

Сервер принимает три строки, выполняет требуемую операцию над полученными операндами, и передает результат на стандартный вывод. Обеспечить также сокращение дроби, если это возможно. Если оба операнда арифметической операции являются рациональными дробями, результатом тоже должна быть рациональная дробь. Для операций сравнения достаточно результата «Истина» или «Ложь».

Вариант 21. Клиент принимает от пользователя элементы целочисленного одномерного массива, а также значение элемента (ключа) для поиска, и отправляет серверу.

Сервер принимает массив и значение элемента для поиска («ключ»), осуществляет поиск элемента по ключу любым известным методом, кроме прямого (*линейного, грубой силы*) поиска, и передает результат на стандартный вывод.

Вариант 22. Клиент принимает от пользователя две строки символов и отправляет серверу.

Сервер принимает обе строки, осуществляет в первой строке замену латинских букв на их аналоги из кириллицы, а во второй строке заменяет символы кириллицы латинскими буквами. Затем результирующие строки передаются на стандартный вывод. Учесть тот факт, что некоторые латинские литеры могут не иметь однобуквенных кириллических эквивалентов, а некоторые символы кириллицы вообще не представимы латинскими буквами.

Вариант 23. Клиент принимает от пользователя последовательность символьных строк и отправляет серверу.

Сервер принимает последовательность символьных строк, осуществляет смену регистра всех букв английского и русского алфавитов, а также замену символов табуляции четырьмя пробелами, удаляет хвостовые пробелы, выполняя эти операции везде, за исключением подстрок, заключенных в кавычки или апострофы, и передает результат на стандартный вывод.

Вариант 24. Клиент принимает от пользователя два целых числа и отправляет серверу.

Сервер принимает два числа. Если первое число является степенью второго числа, то на стандартный вывод передается показатель степени, и сообщение « не является степенью числа » - в противном случае (на месте прочерков выводятся конкретные числа). Например, при вводе «81 -3» должно выводиться «4», а ввод «81 4» – «81 не является степенью числа 4». Учесть областей определения и значений степенной функции целочисленного переменного.

Вариант 25. Клиент принимает от пользователя коэффициенты квадратного уравнения ($ax^2+bx+c=0$), кубического уравнения ($ax^3+bx^2+cx+d=0$) или биквадратного уравнения ($ax^4+bx^3+cx^2+dx+e=0$), каждый из которых не равен нулю или единице, и отправляет их серверу.

Сервер принимает коэффициенты квадратного, кубического или биквадратного уравнения и передает на стандартный вывод разложение биквадратного, кубического или квадратного уравнения на множители. Ограничение: для кубического и квадратного уравнения исключить комплексные корни.

Вариант 26. Клиент принимает от пользователя две или три строки. При вводе трех строк первая и третья – это комплексные числа вида « $\pm 1.2.\pm i5.4$ » (моделируется программно двумя вещественными и/или целыми числами), причем первый знак необязателен, вторая строка – это знак операции вида «+», «-», «*», «/». При вводе двух строк первая — комплексное число, вторая — операция, обозначенная литерами S (комплексно-сопряженное число), R (вещественная часть), I (мнимая часть). Затем строки отправляются серверу.

Сервер принимает комплексные числа и знак операции, выполняет требуемую операцию с полученными операндами, и передает результат на стандартный вывод.

Вариант 27. Клиент принимает от пользователя целочисленную квадратную матрицу и отправляет ее серверу.

Сервер принимает целочисленную квадратную матрицу, после этого **заменяет** строку с минимальным элементом нулями, а затем **удаляет** путем перемещения соответствующих элементов матрицы столбец с максимальным элементом. На стандартный вывод передается результирующая матрица, номера модифицированной строки и удаленного столбца.

Вариант 28. Клиент принимает от пользователя целочисленную матрицу и отправляет ее серверу.

Сервер принимает матрицу, упорядочивает ее строки в порядке убывания суммы модулей их элементов, а затем передает на стандартный вывод результирующую матрицу и значения сумм модулей элементов каждой строки.

Вариант 29. Клиент принимает от пользователя строку, содержащую дату в формате ДД.ММ.ГГГГ и отправляет ее серверу

Сервер принимает дату и передает на стандартный вывод название дня недели, соответствующего введенной дате (для 18.12.2016 — «воскресенье»/«Sunday»). И сервер, и клиент должны учитывать ввод некорректных значений (например, 36.13.2006 или 04.02.1918). В качестве базовой даты принять 1 января 1901.

Вариант 30. Клиент принимает от пользователя строку, содержащую дату в формате ДД.ММ.ГГГГ и отправляет ее серверу.

Сервер принимает дату и передает на стандартный вывод 6-недельный календарь (колонки — дни недели), содержащий введенную дату. Эта дата должна быть особым образом выделена среди окружающих ее дат и не должна располагаться на заголовочной, первой и последней строке календаря. И сервер, и клиент должны учитывать ввод некорректных значений. В качестве базовой даты принять 1 января 1601.

Вариант 31. Клиент принимает от пользователя три строки: 1) дата в формате ДД.ММ.ГГ; 2) местное время (Красноярск, UTC+7) в формате ЧЧ.ММ; 3) Часовой пояс, в котором необходимо узнать время и дату в формате N , где N — число от -12 до 14. Затем три строки отправляются серверу.

Сервер принимает строки и передает на стандартный вывод дату и время в выбранном часовом поясе. Необходимо учесть переход в следующие и предыдущие сутки (например, когда в Красноярске 01.03.2010, 08.15 в Мехико 28.02.2010, 19.15), а также ввод некорректных значений (например, 36.13.2006, 25.15). В качестве базовой даты принять 1 января 1901.

Вариант 32. Клиент принимает от пользователя две строки и отправляет их серверу.

Сервер принимает две строки и передает на стандартный вывод, помимо содержимого самих строк, минимальное количество стираний, замен и добавлений символов, преобразующих одну строку в другую¹.

¹ Расстояние Левенштейна - http://ru.wikipedia.org/wiki/Расстояние_Левенштейна