

**Общая постановка задачи.**

Требуется: разработать две программы: первая реализует серверную часть, вторая — клиентскую часть. Обмен данными между ними организуется посредством механизма Internet-сокетов и протокола TCP либо UDP. Результат выполнения выводится на терминал/консоль. Должен использоваться интерфейс командной строки (CLI). При реализации обязательно использование изученных в лекционном курсе системных вызовов (OS Linux), предназначенных для работы с сокетами. Программный код, относящийся к пользовательскому интерфейсу, должен быть физически отделен от кода, реализующего межпроцессное взаимодействие, и оба они, в свою очередь, отделены от кода реализации основной логики, например, вычислений.

Допускается реализация одной программы в форме Windows-приложения, а второй — в форме Linux-приложения.

Далее оговаривается функционал клиентской и серверной частей. Обе части должны быть устойчивы к некорректному пользовательскому вводу.

В нечетных вариантах заданий используются потоковые сокеты, в четных — дейтаграммные сокеты.

**Варианты заданий**

**Вариант 1.** Клиент отправляет серверу два беззнаковых целых числа  $L$  и  $U$ , введенные пользователем, каждое из которых не может иметь более 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 экран должны выводиться 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  $p, k > 0$ .

Сервер принимает номер и порядок числа, вычисляет по ним ОЧФ номером  $k$  и порядком числа  $p$  и выводит его на экран. Формула для вычисления ОЧФ:

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

Шестое ОЧФ первого порядка равно 8, а восьмое ОЧФ третьего порядка равно 25.

**Вариант 3.** Разработать две программы – сервер и клиент.

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

Сервер принимает строку, хранящую целое число со знаком, и выводит на экран строковый эквивалент этого числа прописью. Например, ввод «-1211» должен приводить к выводу «минус одна тысяча двести одиннадцать», а ввод «100000001» приводит к выводу «сто миллионов 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.

Сервер принимает строку, хранящую вещественное число в обычном или экспоненциальном форматах, и выводит на экран строковый 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) до «точки» не более 15 знаков; 3) для экспоненциального формата после «точки» не более шести знаков; 4) знак «точки» - это «.»(точка) или «,» (запятая).

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

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

**Вариант 6.** Клиент отправляет серверу два введенных пользователем целых числа  $A$  и  $B$ , где  $A$  – сдвигаемое число,  $B$  — величина арифметического или циклического сдвига (отрицательное число означает сдвиг вправо, положительное — влево).

Сервер принимает числа  $A$  и  $B$ , после этого осуществляется арифметический, логический и циклический сдвиг числа  $A$  на величину  $B$ , и результаты выводятся на экран.

**Вариант 7.** Клиент принимает от пользователя некое слово, после чего проверяет его на корректность (отсутствие знаков препинания, цифр) и отправляет его серверу.

Сервер принимает слово, проверяет, является ли введенное слово палиндромом, и информирует пользователя о результате проверки. Учесть смещение и «похожесть» латинских и кириллических символов. Например, следует считать латинскую  $B$  эквивалентной кириллическому символу  $В$ .

**Вариант 8.** 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 даты, вычисляет полное количество дней, прошедших между двумя полученными датами, и выводит его на экран. В качестве базовой даты принять 1 января 1601.

Обе программы 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).

**Вариант 9.** 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. Программа должна принимать во внимание некорректные значения, например, 28.00.06, или 01.99.20, или 08.08.65.

Сервер принимает обе строки, вычисляет полное количество секунд, прошедших между двумя значениями времени двумя разными способами, и выводит результаты на экран.

**Вариант 10.** 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) поиска, и выводит на экран полученное значение.

**Вариант 11.** 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 выводит на экран число, месяц и год прописью (например, ввод «29.02.2012» приводит к выводу «Двадцать девятое февраля две тысячи двенадцатого года»).

Обе программы должны принимать во внимание некорректные даты, например, 37.06.2006 или 01.18.2006. Следует также учесть такие особенности Григорианского календаря<sup>1</sup>, как переход на него в России в 1918 году (отсутствие дней с 1 по 13 февраля). В качестве базовой даты принять 1 января 1601.

**Вариант 12.** 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. Программа должна принимать во внимание некорректные значения, например, 28.00.06,14; 01.99.20,15; 08.08.65,16.

Сервер принимает время и выводит на экран значение часов, минут, секунд и их долей прописью (например, ввод «12.01.20,15» приводит к выводу «двенадцать часов одна минута двадцать и пятнадцать сотых секунды»).

---

<sup>1</sup> Григорианский календарь - [http://ru.wikipedia.org/wiki/Григорианский\\_календарь](http://ru.wikipedia.org/wiki/Григорианский_календарь)

При реализации достаточно обеспечить обработку десятых или сотых долей секунды, более точные значения — по инициативе разработчика.

**Вариант 13.** Клиент принимает от пользователя беззнаковое целое число  $N$  — основание системы счисления (диапазон (1..20]) и последовательность цифр в соответствии с заданной системой счисления, отправляет серверу.

Сервер принимает основание системы счисления и число в этой системе, выводит число на экран, переводит его в десятичную систему, выводит на экран, дополняет его (инвертирует) до максимальной цифры в заданной системе счисления, выводит на экран значение инвертированной последовательности, переводит ее в десятичную систему и выводит полученное число на экран.

**Вариант 14.** Клиент принимает от пользователя беззнаковое целое число  $N$  — основание системы счисления (диапазон (1..20]) и последовательность цифр в соответствии с заданной системой счисления, отправляет серверу.

Сервер принимает основание системы счисления и число в этой системе, выводит число на экран, переводит его в десятичную систему, выводит на экран, осуществляет его реверс (меняет порядок следования знаков на обратный), выводит на экран значение измененной последовательности, переводит ее в десятичную систему и выводит его на экран.

**Вариант 15.** Клиент отправляет серверу три строки, введенных пользователем (первая и третья строки — это правильные дроби вида «1/6» или «0,75», вторая строка — это знак арифметической операции вида «+», «-», «\*», «/» или операции сравнения - «<», «>», «=», «!=», «>=», «<=»).

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

**Вариант 16.** Клиент отправляет серверу элементы двух квадратных матриц одинакового размера. Его, а также сами элементы матриц должен вводить пользователь.

Сервер принимает две квадратные матрицы, а затем выводит на экран сумму матриц, а также определитель суммарной матрицы.

**Вариант 17.** Клиент отправляет серверу элементы введенной пользователем квадратной матрицы, а также два введенных им числа — номер столбца и номер строки.

Сервер принимает матрицу и номер столбца  $N$  и номер строки  $M$ , выводит ее на экран, затем «вычеркивает» из матрицы столбец с номером  $N$  и строку с

номером  $M$ , выводит измененную матрицу на экран, вычисляет и выводит на экран значение ее определителя.

**Вариант 18.** Клиент принимает от пользователя элементы целочисленного одномерного массива, а также целочисленное значение для поиска, реализует проверку массива на упорядоченность элементов по неубыванию и отправляет серверу.

Сервер принимает массив и ключ поиска, затем осуществляет поиск элемента в массиве двумя любыми известными методами, кроме прямого (линейного, грубой силы) поиска, и выводит результат на экран.

**Вариант 19.** Клиент принимает от пользователя последовательность строк символов и отправляет серверу.

Сервер принимает последовательность строк, осуществляет смену регистра всех букв английского алфавита, а также замену символов табуляции двумя пробелами, удаляет хвостовые пробелы, выполняя эти операции везде, за исключением подстрок, заключенных в кавычки или апострофы, и выводит результат на экран.

**Вариант 20.** Клиент принимает от пользователя коэффициенты квадратного уравнения ( $ax^2+bx+c=0$ ) или кубического уравнения ( $ax^3+bx^2+cx+d=0$ ), каждый из которых не равен нулю или единице, и отправляет их серверу.

Сервер принимает коэффициенты и выводит на экран разложение кубического или квадратного уравнения на множители.

**Вариант 21.** Клиент принимает от пользователя две либо три строки и отправляет их пользователю. 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).

Сервер принимает строки, выполняет запрошенную операцию над полученными операндами, и выводит результат на экран.

**Вариант 22.** 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 элементов столбец с максимальным элементом. На экран выводится результирующая матрица, номера модифицированной строки и удаленного столбца.

**Вариант 23.** 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 выводит на экран результирующую матрицу и значения сумм модулей элементов каждой строки.

**Вариант 24.** Клиент принимает от пользователя строку, содержащую дату в формате ДД.ММ.ГГГГ и отправляет ее серверу. Необходимо учесть ввод некорректных значений (например, 36.14.2106 или 04.02.1918). В качестве базовой даты принять 1 января 1601.

Сервер получает строку с датой и выводит на экран день недели, соответствующий введенной дате (для 08.04.2013 — «понедельник» или «monday»).

**Вариант 25.** Клиент принимает от пользователя три строки: 1) дата в формате ДД.ММ.ГГ; 2) местное время (Красноярск, UTC +8) в формате ЧЧ.ММ; 3) Часовой пояс, в котором необходимо узнать время и дату в формате  $N$ , где  $N$  — число от -12 до +14; и отправляет их серверу. Необходимо учесть ввод некорректных значений (например, 36.13.2006, 25.15). В качестве базовой даты принять 1 января 1901.

Сервер принимает три строки, производит требуемые операции и выводит на экран дату и время в выбранном часовом поясе. Необходимо учесть переход в следующие 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).

**Вариант 26.** Клиент принимает от пользователя две строки и отправляет серверу.

Сервер принимает три строки и выводит минимальное количество стираний, замен и добавлений символов, преобразующих одну строку в другую.

**Вариант 27.** Клиент принимает от пользователя строку, содержащую дату в формате ДД.ММ.ГГГГ и отправляет серверу. Необходимо учесть ввод некорректных значений (например, 46.14.2008 или 05.02.1918). В качестве базовой даты принять 1 января 1901.

Сервер принимает дату и выводит на экран 6-недельный календарь (столбцы либо строки — 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 при горизонтальном расположении дней (для столбцов — при вертикальном расположении).