

**Приложение А. Общая постановка задачи.**

Требуется разработать программу в виде *Linux*-приложения, которая распараллеливает выполнение на несколько потоков. Функции дополнительных рабочих потоков не должны осуществлять ввод и/или вывод, если это не оговорено особо в конкретном задании. Ввод необходимых данных и вывод результатов выполнения на терминал/консоль осуществляется в основном (начальном) потоке. Для ожидания окончания работы потоков можно воспользоваться механизмом синхронизации по завершению.

**Вариант 1.** Программа принимает от пользователя два беззнаковых целых числа  $L$  и  $U$ , каждое из которых состоит в точности из 6 знаков.  $L$  – это нижняя граница диапазона,  $U$  – верхняя граница диапазона. Затем она 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».

Проверку того, что число является «счастливым» оформить как три разные функции потока. Значения границ диапазона вводятся пользователем, затем запускается цикл по проверке чисел в виде запускаемых потоков, а потом, после их завершения, на экран выводятся результаты.

**Вариант 2.** Программа принимает от пользователя номер  $k$  обобщенного числа Фибоначчи (ОЧФ) и его порядок  $p$ , причем  $p, k > 0$ , а затем вычисляет значение ОЧФ с заданными номером  $k$  и порядком числа  $p$ , по формуле

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

и выводят его на экран.

Таким образом шестое ОЧФ первого порядка равно 8, а восьмое ОЧФ третьего порядка равно 25.

Вычисление ОЧФ с заданными номером и порядком оформить как функцию потока. Сначала вводятся необходимые данные, затем запускается поток и только после этого должны выводиться результаты.

**Вариант 3.** Программа принимает от пользователя строку, хранящую десятичное целое число со знаком, и выводит на экран эквивалент этого числа прописью. Например, ввод «-1211» должен приводить к выводу «минус одна тысяча двести одиннадцать», а ввод «100000011» приводит к выводу «сто миллионов одиннадцать». Ограничения накладываются только представлениями максимально и минимально возможного знакового целого числа используемого языка программирования.

Получение строкового эквивалента числа оформить как функцию потока. Сначала вводится число, затем запускается поток и только после этого выводятся результаты.

**Вариант 4.** Программа принимает от пользователя 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) для экспоненциального формата после «точки» не более трех знаков; 3) до точки не более 12 знаков для обычного формата; 4) знак точки — это «.».

Получение строкового эквивалента числа оформить как функцию потока. Сначала вводится число, затем запускается поток и только после этого выводятся результаты.

**Вариант 5.** Программа принимает от пользователя две строки. Далее, 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.

Проверку строки на хранение целого либо вещественного числа оформить как две разные функции потока. Сначала вводятся две строки, затем запускаются два потока, и только по их завершению выводятся результаты.

**Вариант 6.** Программа принимает от пользователя две квадратные матрицы одинакового размера с элементами десятичного типа с плавающей запятой. Размер матрицы и значение элементов вводит пользователь. Затем программа выводит на экран сумму матриц и значения определителей и их произведения.

Вычисление определителя и суммы матриц оформить как две функции потока. Сначала вводятся размер и элементы матриц, затем запускаются потоки, и только после этого выводятся результаты.

**Вариант 7.** Программа принимает от пользователя два десятичных целых числа  $A$  и  $B$ , где  $A$  – сдвигаемое число,  $B$  — величина арифметического, логического или циклического сдвига. После этого осуществляется арифметический, циклический и логический сдвиги числа  $A$  на величину  $B$ , и результаты выводятся на экран. При  $B < 0$  осуществляется сдвиг вправо, в противном случае — влево.

Арифметический, логический и циклический сдвиги оформить как три функции потока. Сначала вводятся числа  $A$  и  $B$ , затем запускаются потоки, и только после этого выводятся результаты.

**Вариант 8.** Программа принимают от пользователя некое слово, после чего проверяют его на корректность (отсутствие знаков препинания, цифр, смещения латинских и кириллических символов). Затем программа проверяет, является ли введенное слово палиндромом, и информирует пользователя о результате проверки.

Проверку корректности заданного слова, а также того, что введенное слово — палиндром, оформить как две разные функции потока. Сначала осуществляется ввод слова, после этого запускаются необходимые потоки, а затем выводятся результаты.

**Вариант 9.** Программа принимает от пользователя одномерный массив строк, перегруппировывает согласно пожеланиям пользователя (возрастание, убывание, без упорядочивания<sup>1</sup>), любым из так называемых «улучшенных алгоритмов» сортировки массивов (исключено использование следующих алгоритмов: *пузырьковая* сортировка, *шейкерная* сортировка, сортировка *прямым выбором* и сортировка *прямыми включениями/вставками*) и выводит отсортированный массив на экран.

Сортировку и «рассортировку» массива оформить как две функции потока. Сначала вводятся элементы массива, затем он сортируется или «рассортируется» и только после этого на экран должен выводиться результат.

**Вариант 10.** Программа принимает от пользователя квадратную матрицу с элементами десятичного типа с плавающей точкой, вычисляют обратную ей матрицу, и выводит элементы обеих матриц на экран. Обеспечить проверку возможности вычисления обратной матрицы.

Вычисление обратной матрицы оформить как функцию потока. Сначала вводятся элементы исходной матрицы, затем вычисляется обратная ей матрица и только после этого на экран должен выводиться результат.

---

<sup>1</sup> В этом случае элементы массива не должны быть отсортированы ни по возрастанию, ни по убыванию.

**Вариант 11.** Программа принимает от пользователя квадратную матрицу, а также два числа – номер столбца и номер строки, 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.

«Вычеркивание» строки и столбца из матрицы оформить как функцию потока. Сначала вводятся элементы исходной матрицы, номер строки и столбца, затем осуществляется «вычеркивание» элементов из матрицы и только после этого на экран должен выводиться результат.

**Вариант 12.** Программа принимает от пользователя две даты – строки вида ЦЦ.ЦЦ.ЦЦЦЦ, где Ц – это любая цифра из диапазона [0-9]. Далее она вычисляет полное количество дней, прошедших между двумя введенными датами, и выводит его на экран. Программа должна принимать во внимание некорректные даты, например, 37.06.2006 или 01.18.2010. Следует также учесть 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.

Вычисление разницы между датами оформить как функцию потока. Сначала вводятся даты, затем вычисляется разница между ними и только после этого на экран должен выводиться результат.

**Вариант 13.** Программа принимает от пользователя два значения времени суток – строки вида ЦЦ.ЦЦ.ЦЦ, где Ц – это любая цифра из диапазона [0-9]. Далее они вычисляют полное количество секунд, прошедших между двумя значениями времени, и выводят его на экран. Программа должна принимать во внимание некорректные значения, например, 28.00.06, или 01.99.20, или 08.08.65.

Вычисление разницы между значениями времени суток двумя разными способами оформить как две функции потока. Сначала вводятся эти значения, затем вычисляется разница между ними и только после этого на экран должен выводиться результат.

**Вариант 14.** Программа принимает от пользователя две строки, осуществляет поиск вхождения второй строки в первую любым известным методом, кроме прямого (линейного, грубой силы), и выводит на экран значение индекса элемента первой строки, с которого началось совпадение, или строку «Нет вхождений» в противном случае.

Поиск вхождения подстроки в строку оформить как функцию потока. Сначала вводятся две строки, затем осуществляется поиск и только после этого на экран должен выводиться результат.

**Вариант 15.** Программа принимает от пользователя две строки, осуществляют поиск количества вхождений второй строки в первую любым известным методом, кроме прямого (линейного, грубой силы), и выводит на экран полученное значение.

Поиск количества вхождений подстроки в строку оформить как функцию потока. Сначала вводятся две строки, затем осуществляется поиск и только после этого на экран должен выводиться результат.

**Вариант 16.** Программа принимает от пользователя дату – последовательность символов вида ЦЦ.ЦЦ.ЦЦЦЦ, где Ц – это любая цифра из диапазона [0-9], и выводят на экран число, месяц и год прописью, а затем – слово « года» (например, ввод «29.02.2008» приводит к выводу «Двадцать девятое февраля две тысячи восьмого года»). Программа должна принимать во внимание некорректные даты, например, 37.06.2006 или 01.18.2006. Следует учесть 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.

Перевод даты из числового формата в пропись оформить как функцию потока. Сначала вводится дата, затем осуществляется перевод даты и только после этого на экран должен выводиться результат.

**Вариант 17.** Программа 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 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»). Программа должна принимать во внимание некорректные значения, например, 28.00.06,14; 01.99.20,15; 08.08.65,16.

Перевод времени суток из числового формата в пропись оформить как функцию потока. Сначала вводится значение времени, затем осуществляется перевод значения времени в пропись и только после этого на экран должен выводиться результат.

**Вариант 18.** Программа принимает от пользователя беззнаковое целое десятичное число  $N$  – основание системы счисления ( $1 < N < 17$  или  $N = 1$ ) и последовательность цифр в соответствии с заданной системой счисления. Затем программа 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 число в десятичной системе и выводит полученное число на экран. Для случая  $N = 1$ , когда используется унарная система счисления, результатом является только перевод в десятичную систему и его вывод на экран.

Перевод числа в десятичную систему счисления и инвертирование числа оформить как две разных функции потока. Сначала вводятся необходимые значения, затем запускаются требуемые потоки и только после этого на экран должен выводиться результат.

**Вариант 19.** Программа принимает от пользователя беззнаковое целое десятичное число  $N$  – основание системы счисления ( $1 < N < 17$  или  $N = 1$ ) и последовательность цифр в соответствии с заданной системой счисления. Затем программа 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 число в десятичной системе и выводит его на экран. Для случая  $N = 1$ , когда используется унарная система счисления, результатом является только перевод в десятичную систему и его вывод на экран.

Перевод числа в десятичную систему счисления и реверс числа оформить как две разных функции потока. Сначала вводятся необходимые значения, затем запускаются требуемые потоки и только после этого на экран должен выводиться результат.

**Вариант 20.** Программа принимает от пользователя три строки (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 « $<$ », « $>$ », « $=$ », « $!=$ », « $>=$ », « $<=$ »), выполняет требуемую операцию над полученными операндами и выводит результат на экран. 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*».

Выполнение арифметической операции и операции сравнения над полученными операндами оформить как две разные функции потока. Сначала вводятся необходимые значения, затем запускаются требуемые потоки и только после этого на экран должен выводиться результат.

**Вариант 21.** Программа принимает от пользователя элементы целочисленного одномерного массива, а также значение элемента (ключа) для поиска, затем осуществляет поиск элемента по ключу любым известным методом, кроме прямого (линейного) поиска, и выводит результат на экран.

Поиск элемента (числа) в одномерном массиве оформить как функцию потока. Сначала вводятся необходимые значения, затем осуществляется поиск и только после этого на экран должен выводиться результат.

**Вариант 22.** Программа принимает от пользователя две символьные 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 выводятся на экран. Необходимо учесть тот факт, что некоторые латинские литеры могут не иметь однобуквенных кириллических эквивалентов, а некоторые символы кириллицы вообще не представимы латинскими буквами.

Замену символов реализовать как две разные функции потока. Сначала вводятся необходимые значения, затем запускаются требуемые потоки и только после этого на экран должен выводиться результат.

**Вариант 23.** Программа принимает от пользователя последовательность символьных строк, осуществляет смену регистра всех букв английского и русского алфавитов, а также замену символов табуляции четырьмя пробелами, удаляет лидирующие пробелы, выполняя эти операции везде, за исключением подстрок, заключенных в кавычки или апострофы, и выводит результат на экран.

Смену регистра символов и удаление пробельных символов реализовать как две разные функции потока. Сначала вводятся необходимые значения, затем запускаются требуемые потоки и только после этого на экран должен выводиться результат.

**Вариант 24.** Программа принимает от пользователя два целых числа. Далее, если первое число является степенью второго числа, то на экран выводится 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» приводит к выводу «81 не является степенью числа 4». Учесть области определения и значений степенной функции целочисленного переменного.

Проверку на то, что первое число является степенью второго числа, оформить как функцию потока. Сначала вводятся два числа, затем выполняется требуемая операция и только после этого на экран должен выводиться результат.

**Вариант 25.** Программа 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. Затем программа выводит на экран разложение биквадратного, кубического или квадратного уравнения на множители. Ограничение: для кубического и квадратного уравнения исключить комплексные корни.

Нахождение значений  $x_1, x_2, x_3, x_4$  должно оформляться как три потоковые функции (первая — для квадратного уравнения, вторая — для кубического уравнения, третья — для биквадратного уравнения), которая запускается после ввода коэффициентов уравнения. Результат должен выводиться только по завершению выполнения потока.

**Вариант 26.** Программа принимает от пользователя две или три строки. При вводе трех строк первая и третья — это комплексные числа<sup>1</sup> 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 программа выполняет требуемую операцию с полученными операндами, и выводит результат на экран.

Выполнение требуемых операций над комплексными числами оформить в виде функций потока, которые запускаются после осуществления необходимого ввода. Результаты должны выводиться только по завершению выполнения всех потоков.

**Вариант 27.** Программа принимает от пользователя целочисленную квадратную матрицу, после этого заменяет строку с минимальным элементом нулями, а затем **удаляет** путем перемещения соответствующих элементов матрицы столбец с максимальным элементом. На экран выводится результирующая матрица, номера модифицированной строки и удаленного столбца.

Поиск номера строки с минимальным элементом и номера строки с максимальным элементами, а также требуемая модификация матрицы должны быть оформлены как отдельные функции потока, которые запускаются после осуществления необходимого ввода. Результаты должны выводиться только по завершению выполнения всех потоков.

---

<sup>1</sup> Комплексное число - [https://ru.wikipedia.org/wiki/Комплексное\\_число](https://ru.wikipedia.org/wiki/Комплексное_число)



**Вариант 28.** Программа принимает от пользователя целочисленную матрицу и упорядочивает строки матрицы в порядке убывания суммы модулей их элементов. На экран выводится результирующая матрица и значения сумм модулей элементов каждой строки.

Упорядочивание строк оформляется как отдельная функция потока, которая запускается после ввода матрицы. Результаты должны выводиться только по завершению выполнения потока.

**Вариант 29.** Программа принимает от пользователя строку, содержащую дату в формате ДД.ММ.ГГГГ, и выводит на экран день недели, соответствующий введенной дате (для 18.12.2016 — «воскресенье»/«Sunday»). Необходимо учесть возможный ввод некорректных значений (например, 36.13.2006 или 04.02.1918). В качестве базовой даты принять 1 января 1901.

Вычисление номера дня недели, на которую приходится указанная дата, должно быть оформлено как отдельная функция потока, которая запускается после осуществления необходимого ввода. Результаты должны выводиться только по завершению выполнения всех потоков.

**Вариант 30.** Программа принимает от пользователя строку, содержащую дату в формате ДД.ММ.ГГГГ, и выводит на экран 6-недельный календарь<sup>1</sup> (колонки 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 (столбце) календаря при горизонтальном (вертикальном) расположении дней. Необходимо учесть ввод некорректных значений (например, 46.14.2008 или 05.02.1918). В качестве базовой даты принять 1 января 1601.

Вычисление позиции заданной даты в таблице — 6-недельном календаре должно быть оформлено как отдельная функция потока, которая запускается после осуществления необходимого ввода. Результаты должны выводиться только по завершению выполнения всех потоков.

**Вариант 31.** Программа принимает от пользователя три строки: 1) дата в формате ДД.ММ.ГГГГ; 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. Программа должна вывести на экран 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.

Расчет времени в новом часовом поясе оформить как функцию потока. Сначала вводятся исходные данные, затем производятся вычисления, и только после этого на экран должен выводиться результат.

---

<sup>1</sup> Пример календаря - <http://www.macmax.ru/calendar/>

**Вариант 32.** Программа принимает от пользователя две строки и выводит минимальное количество стираний, замен и добавлений символов, преобразующих одну строку в другую<sup>1</sup>.

Нахождение минимального количества стираний, замен и добавлений символов оформить как функцию потока. Сначала вводятся исходные данные, затем производятся вычисления, и только после этого на экран должен выводиться результат.

---

1 Расстояние Левенштейна - [http://ru.wikipedia.org/wiki/Расстояние\\_Левенштейна](http://ru.wikipedia.org/wiki/Расстояние_Левенштейна)