**Домашнее задание 1. 09.09.2013**

[Конспект пары](https://docs.google.com/document/d/1q67xg0fj14F_2btxkXGl1hXoJo5JIn4Pwc70gRyPPWM/edit)

1. Написать программу, считающую значение формулы x^4 + x^3 + x^2 + x за два умножения.
2. Реализовать алгоритм нахождения неполного частного от деления a на b (целые числа), используя только операции сложения, вычитания и умножения.
3. Дан массив целых чисел x[1]...x[m+n], рассматриваемый как соединение двух его отрезков: начала x[1]...x[m] длины m и конца x[m+1]...x[m+n] длины n. Не используя дополнительных массивов, переставить начало и конец.
4. Посчитать число "счастливых билетов" (билет считается "счастливым", если сумма первых трёх цифр его номера равна сумме трёх последних).
5. Написать программу проверки баланса скобок в исходной строке (т.е. число открывающих скобок равно числу закрывающих и выполняется правило вложенности скобок).
6. Заданы две строки: S и S1. Найти количество вхождений S1 в S как подстроки.
7. Написать программу, печатающую все простые числа, не превосходящие заданного числа.
8. Реализовать подсчет факториала (рекурсивно и итеративно).
9. Посчитать целую степень числа: a^n.
10. Реализовать программу, проверяющую, является ли строка палинромом.
11. Реализовать быструю сортировку (в рекурсивном варианте).

**Домашнее задание 2. 16.09.2013**

[Конспект пары](https://docs.google.com/document/d/10jKM5i0vaDHJZ0Rn7CaE9rBO1G1poPMUc8oxayq4E-Y/edit)

1. Реализовать рекурсивный и итеративный подсчет чисел Фибоначчи.
2. Реализовать возведение в целую степень (с логарифмической сложностью алгоритма).
3. Напечатать все представления натурального числа N суммой натуральных слагаемых. Перестановка слагаемых нового способа не дает.
4. Напечатать в порядке возрастания все простые несократимые дроби, заключенные между 0 и 1, знаменатели которых не превышают n.
5. Реализовать алгоритм пирамидальной сортировки.
6. *[штрафная задача]*  
   Дан массив с коэффициентами многочлена от x. Требуется написать программу, "красиво" выводящую этот многочлен на консоль в две строки.Например, многочлен, представляемый коэффициентами 1 0 5 -10, выводится так:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 3 |  |  |  |  |  |  |
| x |  | + | 5 | x | - | 1 | 0 |

1. *[дополнительно для желающих]*  
   Реализовать консольную игру ["Быки и коровы"](http://ru.wikipedia.org/wiki/%D0%91%D1%8B%D0%BA%D0%B8_%D0%B8_%D0%BA%D0%BE%D1%80%D0%BE%D0%B2%D1%8B).

**Домашнее задание 3. 23.09.2013**

[Конспект пары 23.09.13](https://docs.google.com/document/d/1Kz5U8PgO2Y4wI-GDjGREDLJeEvJmNgmqijK5vWuSD9k/edit)  
[Конспект пары 25.09.13](https://docs.google.com/document/d/1cfDiJ33hMBSNRBqSTFy1JnPNR9DfIpLaoIJ92eJXLeI/edit)

1. Даны две строки. Определить, можно ли, переставляя символы в первой строке, получить вторую строку. Хочется решение быстрее, чем за O(n^2).
2. Дан массив размерностью N x N, N — нечетное число. Вывести элементы массива при обходе его по спирали, начиная с центра.
3. Написать программу, которая переставляет цифры натурального числа таким образом, чтобы образовалось наименьшее число, записанное этими же цифрами.
4. Написать программу, которая в диалоговом режиме позволяет осуществлять следующие операции:  
   0 - exit  
   1 - add a value to sorted list  
   2 - remove a value from sorted list  
   3 - print list  
   Все операции должны сохранять сортированность. Начинаем с пустого списка. Список должен быть оформлен отдельным модулем.
5. "Считалочка" — отряд из 41-го сикария, защищавший галилейскую крепость Массада, не пожелал сдаваться в плен блокировавшим его превосходящим силам римлян. Сикарии стали в круг и договорились, что каждые два воина будут убивать третьего, пока не погибнут все. Самоубийство — тяжкий грех, но тот, кто в конце концов останется последним, должен будет его совершить. Иосиф Флавий, командовавший этим отрядом, якобы быстро рассчитал, где нужно стать ему и его другу, чтобы остаться последними, но не для того, чтобы убить друг друга, а чтобы сдать крепость римлянам. В нашем случае участвует n воинов и убивают каждого m-го. Требуется определить номер k начальной позиции воина, который должен будет остаться последним. Считать с помощью циклического списка, который должен быть оформлен отдельным модулем.
6. *[штрафная задача]*  
   Реализовать сортировку вставками на базе однонаправленных линейных списков.

**Домашнее задание 4. 30.09.2013**

[Конспект пары 02.10.13](https://docs.google.com/document/d/1leEESZUfuSeiGkkB_aJeERCtp5q-V84WTbag7QPvIyw)  
[Презентация с пары 02.10.13](https://docs.google.com/presentation/d/1RE-_bGaBgV6wGNpZS98dqO954EGsKoZXWZSTFx351oo)

1. Найдите максимальный элемент массива, встречающийся более одного раза (массив неупорядоченный). Задачу хочется решить быстрее, чем за O(n^2).
2. Написать программу преобразования инфиксной формы выражения в постфиксную. Известно, что каждый операнд занимает один символ. В выражении могут быть знаки {+, -, \*, /}, скобки и цифры. Пример: (1 + 1) \* 2 должно преобразовываться в 1 1 + 2 \*. Алгоритм перевода предлагается найти самостоятельно (алгоритм "сортировочной станции" Э.Дейкстры).
3. Реализовать программу, вычисляющую значение выражения в постфиксной записи с помощью стека.
4. Объединить предыдущие две задачи в одну программу — реализовать программу-калькулятор, вычисляющую значение арифметического выражения в инфиксной нотации. Выражение вводится с консоли, должны поддерживаться операции {+, -, \*, /} и скобки, операнды считать односимвольными (поддержка многосимвольных операндов по желанию).

**Домашнее задание 5. 07.10.2013**

[Конспект пары 09.10.13](https://docs.google.com/document/d/1LibDaZ7YWjoqIPXKLGAiS30l-DlGR_8iwnrtS1yiAqY)

1. Дан файл с текстом. Вывести все слова этого текста, предварительно оставив в каждом слове только первые вхождения каждой буквы.
2. По содержимому памяти вывести значение типа double в экспоненциальной форме: sm\*q^(Sp), где s — знак мантиссы, m — мантисса, q — основание системы счисления, S — знак порядка, p — порядок числа. Примеры допустимого вывода:  
   Enter a number: -2.5  
   Result: -1.25\*2^(1)  
   Enter a number: 12312.323  
   Result: +1.5029691162109375384\*2^(13)
3. Вывести на консоль все однострочные комментарии С++ (вида // комментарий) из входного файла (вместе с символами "//"). До комментария в строке может быть значимый текст, его выводить не надо, строки без комментариев не выводить. Конец строки представляется символом '\n', могут быть полезны функции fgetc и feof. Программа должна учитывать случаи, когда последовательность "//" находится внутри текстовой строки или многострочного комментария (/\* \*/). В таких случаях ничего выводить не нужно.
4. Написать программу-телефонный справочник. Она должна уметь хранить имена и номера телефонов, в интерактивном режиме осуществлять следующие операции:  
   0 - выйти  
   1 - добавить запись (имя и телефон)  
   2 - найти телефон по имени  
   3 - найти имя по телефону  
   4 - сохранить текущие данные в файл  
   При запуске программа должна читать данные из файла, если файла нет — начинать с пустой базы номеров. Формат представления данных в файле придумать самостоятельно.
5. *[штрафная задача]*  
   Дано натуральное число n, меньшее одного миллиона. Напечатать это число словами на английском ("thirty-five", "forty-four thousand one hundred and twenty" и т.п.). Скорее всего, будет полезна [вот эта статья](http://en.wikipedia.org/wiki/English_numerals).

**Домашнее задание 6. 14.10.2013**

[Конспект пары 14.10.13](https://docs.google.com/document/d/10G7ZSF94RFb-_8cnYhnZ-y-wDYGlYPjv08Gi4BTg034)  
[Конспект пары 16.10.13](https://docs.google.com/document/d/1hNYn9yU-RRZSp8HSfJn0D_Kjw7eKgOhrSePhqgM_NPo)

1. Реализовать АТД "множество" на основе двоичного дерева поиска. Программа должна позволять в интерактивном режиме добавлять значения целого типа в множество, удалять значения, проверять, принадлежит ли значение множеству, печатать текущие элементы множества в возрастающем и убывающем порядках, а также в формате (a b c), где a — значение в узле, а b и c — аналогичные представления поддеревьев правого и левого потомка. Пример: "(5 (2 null null) (10 null (12 null null)))". Такой вывод бывает крайне полезен при отладке операций над деревом.
2. По дереву разбора арифметического выражения вычислить его значение. Дерево разбора хранится в файле в виде (<операция> <операнд1> <операнд2>), где <операнд1> и <операнд2> сами могут быть деревьями, либо числами. Например, выражение (1 + 1) \* 2 представляется в виде (\* (+ 1 1) 2). Должны поддерживаться операции +, -, \*, / и целые числа в качестве аргументов. Требуется построить дерево в явном виде, распечатать его (не обязательно так же, как в файле), и посчитать значение выражения обходом дерева. Может быть полезна функция ungetc (если не '(', возвращаем символ в поток и читаем число fscanf-ом). Можно считать, что входной файл корректен. Пример: по входному файлу (\* (+ 1 1) 2) может печататься ((1 + 1) \* 2) и выводиться 4.
3. Переделать множество из задачи 1 на основе АВЛ-дерева.

**Домашнее задание 7. 21.10.2013**

[Конспект пары](https://docs.google.com/document/d/1yTmL_376CgMh4NNoxJu-sgRczaeuUMuPwQRDhyoNx3I)

1. Получив домашнее задание по программированию, группа студентов приступила к решению задач. Три студента с номерами 1, 2 и 3 честно сделали все задание самостоятельно, другие решили списать с кого-нибудь, кто уже имеет готовое решение — либо решенное самостоятельно, либо уже списанное с другого. При проверке выяснилось, что некоторых студентов следует немедленно отчислить, т.к. они не только не написали решение сами, но и поленились списать. Задача: Определить, какой студент какое решение сдавал, и кого надо отчислить.  
   На входе: количество студентов и список пар чисел, где первое число — номер студента, второе — номер того, с кого было списано решение. Требуется вывести список пар чисел, где первое число — номер студента, второе — 1, 2 или 3 — сданный вариант.
2. Описать модуль для работы с АТД "Строка" со следующими операциями: создание, удаление, копирование (функцию clone(), возвращающую полную копию строки), конкатенация (дописывание строки-аргумента к текущей), сравнение на равенство, вычисление длины, проверка на пустоту, выделение подстроки, преобразование к char\*. Строка должна быть потенциально расширяемой в неограниченных пределах.
3. Реализовать алгоритмы для работы с хэш-таблицей (разрешение коллизий методом цепочек). По данному тексту (читается из файла, не ограничен по размеру) посчитать число использований каждого слова. Вывести load factor, среднюю длину цепочки, максимальную длину цепочки и значения, которые в нее попали, общее число добавленных слов, число пустых ячеек таблицы. Для работы со строками использовать модуль "Строка" из задачи 2.

**Домашнее задание 8. 28.10.2013**

[Конспект пары](https://docs.google.com/document/d/1e6GXRLHaYWunbzOXs4V_Zh1BWF-VUQoHOw1tAlxAkQg)

1. Пронумеровать вершины графа в произвольном порядке латинскими буквами. На входе имя файла с заданием графа и начальная вершина, от которой будет осуществляться нумерация. Предполагается, что в графе одна компонента связности.
2. В некоторой стране n городов, соединенных между собой дорогами различной длины. По каждой дороге можно проехать в обе стороны. 1 сентября 1939г силы вермахта подло вторглись в эту страну и захватили город с номером 1. Далее, каждый день они захватывали по одному городу, используя следующий алгоритм: среди всех еще не захваченных городов выбирается ближайший к городу 1 и захватывается.  
   Во входном файле заданы n (число городов) и m (число дорог). Далее следуют сами дороги в формате:  
   i j len  
   i и j — номера городов, len — длина дороги.  
   Задача: вывести номера городов в порядке захвата, а также расстояние от каждого захваченного города до города 1 и сами кратчайшие пути (списком посещенных городов).
3. Вывести компоненты связности заданного графа.

**Домашнее задание 9. 06.11.2013**

[Конспект пары](https://docs.google.com/document/d/1kb0heLa3-DZiye3e8AFC7UDXimicdnfbGC_tI7_Ea4A)  
[Презентация с пары](https://docs.google.com/presentation/d/1TFQIJSI_kq26v8asda8aEGU509TXjb272SdPAWOh4jQ)

1. Даны 2 текстовых файла. Записать в третий файл только те строки, которые встречаются и в первом, и во втором файлах.
2. Реализовать алгоритм Рабина-Карпа поиска подстроки в строке.
3. По данному неориентированному графу построить минимальное остовное дерево одним из рассмотренных алгоритмов. В файле задаётся матрица смежности, программа должна вывести на консоль минимальное остовное дерево в каком-либо представлении.

**Домашнее задание 10. 14.11.2013**

1. Реализовать кодирование текста с помощью [алгоритма Хаффмана](http://habrahabr.ru/post/144200/). На входе программы файл с текстом из латинских букв, пробелов и знаков препинания. На выходе текстовый файл, состоящих из двух строк. Первая строка содержит в себе представление дерева, которое строится в ходе работы алгоритма (например, в отладочном варианте из задачи 6.1), вторая строка — последовательность символов нулей и единиц, в которую кодируется входной текст.
2. Реализовать раскодирование с помощью алгоритма Хаффмана. На входе файл, генерируемый программой из задачи 1, на выходе — исходный закодированный текст.
3. *[дополнительно для желающих]*  
   Переделать упаковщик и распаковщик для работы с бинарным форматом промежуточного файла (упаковщик выдает на выходе бинарный файл, распаковщик его раскодирует обратно в исходный текстовый). Вариант представления дерева в бинарном виде выбрать самостоятельно.

**Домашнее задание 11. 18.11.2013**

[Конспект пары](https://docs.google.com/document/d/17nLYPWR-H1-8LkT4nn2ZpkTiWoCGkguzNG4uGCG_zuw)  
[Презентация с пары](https://docs.google.com/presentation/d/1Cnlox807bUDrApmSrzEN2bTfdLlYAiT1lJtOpcqUmjA)

1. Многочлен с целыми коэффициентами представлен в виде связного списка: в каждом элементе хранится степень x и ненулевой коэффициент. Описать тип данных, реализовать следующие функции:  
   - equals(p,q) -- проверка равенства многочленов p и q;  
   - value(p,x) -- вычисление значения многочлена в точке x  
   - add(p,q,r) -- вычисление суммы многочленов r = p + q;  
   Реализовать интерфейс пользователя, позволяющий выполнять указанные операции над многочленами и выводить полученные результаты.
2. Реализовать конечный автомат по разбору чисел с плавающей точкой, подходящих под следующее регулярное выражение: (+ | -)? digit+ (. digit+)? (E(+ | -)? digit+)?

**Домашнее задание 12. 27.11.2012**

[Конспект пары](https://docs.google.com/document/d/1HS-VeSMz0FCTPB8e-kgKXuvqH1Y9j28mnwtJXRiVge4)   
[Презентация с пары](https://docs.google.com/presentation/d/1Qv2RPoKtsXG-41WMZz5ODrvJcBTZ8LvbpYWJvkf5dRI)

1. Реализовать синтаксический анализатор, разбирающий арифметические выражения методом рекурсивного спуска. Входной поток составляют числа, разбираемые лексическим анализатором из прошлой домашней работы, круглые скобки и знаки операций {+, -, /, \*}

**Домашнее задание 13. 02.12.2013**

[Конспект пары](https://docs.google.com/document/d/13NNbcrF13TPRUmRrpsaF27BLgYB53kBkYpbAHY1NMgw/edit)  
[Презентация](https://docs.google.com/presentation/d/1TwMIvyj5tnJoLJxJeI1TB6rpzrGbxbz7JvBTbn2B1nw/edit)

1. Зарегистрироваться на github.com, прислать мне на почту имя логина. Создать себе репозиторий для домашних задач с адекватным именем.
2. Придумать подходящую структуру папок и выложить в гит все имеющиеся исходники
3. решений задач.