



МАТЕРИАЛЫ ЗАДАНИЙ

*олимпиады школьников
«ЛОМОНОСОВ»
по информатике*

2015/2016 учебный год

Задания отборочного этапа, 10-11 класс

Задача 1-1-1

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 273. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова.

Задача 1-1-2

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 165. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова.

Задача 1-1-3

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 230. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова.

Задача 1-1-4

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению

в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 238. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова.

Задача 1-2-1

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 2348346. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова. После ответа, опишите как он был получен.

Задача 1-2-2

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 1677390. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова. После ответа, опишите как он был получен.

Задача 1-2-3

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 3913910. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова. После ответа, опишите как он был получен.

Задача 1-2-4

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 1118260. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова. После ответа, опишите как он был получен.

Задача 1-3

Вася играет в кубики. Все кубики у Васи одинакового размера 1см3. Кубики у Васи двух видов: либо непрозрачные чёрные, поглощающие весь падающий на них свет, либо прозрачные бесцветные, пропускающие световые лучи. Вася собирает из маленьких кубиков большой куб размером $N \times N \times N$ см3 ($1 < N < 250$). При сборке кубики ставятся вплотную друг к другу, так, что не остаётся пустот. Собрав куб, Вася смотрит сквозь него на солнце, держа его в руке так, чтобы взгляд был направлен перпендикулярно грани куба, ближней к его лицу. Васе интересно, как свет проходит через куб, какая картинка складывается в проекции собранного им куба. Когда солнце зашло, Вася понял, что узнать какая картинка может получиться можно написав программу.

На вход подаётся целое число N и описание внутреннего строения куба. Это описание составлено так, что каждый непрозрачный (чёрный) кубик обозначается нулём, а прозрачный (бесцветный) - единицей. Первым кубиком в описании является тот, который находится в левой нижней ближней к Васе вершине. Эта вершина - начало координат. Система координат правая прямоугольная. Оси координат (абсцисса, ордината и аппликата) идут так, чтобы любое ребро было параллельно одной из осей. Куб лежит в первом октанте системы координат. Ось аппликат идёт в направлении от Васи к солнцу (вдоль неё Вася смотрит на куб). Итак, в описании первым является кубик с координатами $(0, 0, 0)$. За ним следуют остальные N^2-1 кубиков с нулевой аппlikатой, образующие нулевой (ближний к Васе) слой куба: $(1, 0, 0), (2, 0, 0), \dots, (N-1, 0, 0), (1, 1, 0), \dots, (N-1, 1, 0), \dots, (N-1, N-1, 0)$. Далее указываются значения кубиков с аппlikатой 1, образующие следующий слой: $(1, 0, 1), (2, 0, 1), \dots, (N-1, N-1, 1)$. Они упорядочены по тому же принципу, что и кубики нулевого слоя. За первым слоем следует 2-й, ... $N-1$ -й. В описании строения куба значения кубиков идут слитно без разделителей.

Программа должна вывести параллельную ортогональную проекцию куба на плоскость Oxy . При её построении считается, что лучи идут только параллельно оси Oz . Отражений и преломлений в стенках кубиков нет. Прозрачный кубик пропускает лучи, а непрозрачный поглощает без остатка. В выводе программы 0 должен обозначать непрозрачный (чёрный) квадратик проекции, 1 -- прозрачный (бесцветный) квадратик. Порядок квадратиков при выводе проекции соответствует следующему ряду координат: $(0, 0), (1, 0), \dots, (N-1, 0), (0, 1), \dots, (N-1, 1), \dots, (N-1, N-1)$.

Задача 1-4-1

Расшифруйте следующий текст:

HOA ILF YV FPJ LVGPOAFJX BOGQMOFJAC IH FPJ LVHOCPYIVOB RJ JVX IH FPJ MJCFJAV CKYAOR OAW IH
FPJ EOROZN RYJC O CWORR LVAJEOAXJX NJRRIM CLV. IABYFYVE FPYC OF O XYCFOVGJ IH AILEPRN
VYVJFN-FMI WYRRYIV WYRJY YC OV LFFJARN YVCYEVYHYGOVF RYFFRJ BR LJ EAJJV KROVJF MPICJ OKJ-
XJCGJVXJX RYHJ HIAWC OAJ CI OWODYVERN KAYWYFYSJ FPOF FPJN CFYRR FPYVQ XYEYFOR MOFGPJ
OAJ O KAJFFN VJOF YXJO. FPYC KROVJF ПОС - IA AOFPIA ПОХ - О КАИБРЈВ, МРЈГР МОС РРЈС: WICF IH
FPJ KJIKRJ IV YF MJA J LVPOKKN HIA KAJFFN WLGP IH FPJ FYWJ. WOVN CIRLFYIVC MJA J CLEEJCFJX HIA
FPYC КАИБРЈВ, BLF WICF IH FPJ CJ MJA J ROAEJRN GIVGJAVJX MYFP FPJ WISJWJVFC IH CWORR EAJJV
KYJGJC IH KOKJA, МРЈГР YC IXX BJGOLCJ IV FPJ MPIRJ YF MOCV'F FPJ CWORR EAJJV KYJGJC IH KOKJA
FPOF MJA J LVPOKKN.

В качестве ответа отправьте одну строку с расшифрованным текстом, в последующих строках опишите действия по решению задачи.

Задача 1-4-2

Расшифруйте следующий текст:

XB. NRF XBM. FTBMAJZ, LG RTXKJB GLTB, EBUPJI FBUPJ, WJBJ EBLTF IL MNZ IHNI IHJZ WJBJ EJBGSIAZ
RLBXNA, IHNRY ZLT PJBZ XTSH. IHJZ WJBJ IHJ ANMI EJLEAJ ZLT'F JQEJSI IL KJ URPLAPJF UR NRZIHURC
MIBNRCJ LB XZMIJBULTM, KJSNTMJ IHJZ DTMI FUFRI HIAF WUIH MTSH RLRMJRMJ. XB. FTBMAJZ WNM
IHJ FUBJSILB LG N GUBX SNAJF CBTRRURCM, WHUSH XNFJ FBUAAM. HJ WNM N KUC, KJGZ XNR WUIH
HNBFAZ NRZ RJSY, NAIHLTCH HJ FUF HNPJ N PJBZ ANBCJ XTMINSHJ. XBM. FTBMAJZ WNM IHUR NRF
KALRFJ NRF HNF RJNBAZ IWUSJ IHJ TMTNA NXLT RI LG RJSY, WHUSH SNXJ UR PJBZ TMJGTA NM MHJ
MEJRI ML XTSH LG HJB IUXJ SBNRURC LPJB CNBFJR GJRSJM, MEZURC LR IHJ RJUCHKLBM. IHJ FTBMAJZM
HNF N MXNAA MLR SNAJF FTAJZ NRF UR IHJUB LEURULR IHJBJ WNM RL GURJB KLZ NRZWHJBJ.

В качестве ответа отправьте одну строку с расшифрованным текстом, в последующих строках опишите действия по решению задачи.

Задача 1-4-3

Расшифруйте следующий текст:

XHXZO EVFI DU BGMM KLGFX EK FDHXZT RO G RDDCFGKX. XGFI RDDCFGKX IGK KEW KIXMHXK, ADEVA GMYDKQ QD QIX FXEMEVA. KDYX RDDCKIXMHXK GZX KQGFCXT QD QIX RZEY BEQI IGZTRGFC RDDCK: KFEXVFX, YGQIK, IEKQDZO, GVT XHXZOQIEVA XMKX. DQIXZ KIXMHXK IGHX QBD MGOXZK DU LGLXZRGFC KFEXVFX UEFQEDV, BEQI QIX RGFC MGOXZ DU RDDCK LZDLLXT PL DV DMT QEKKPX RDWXXK DZ MXVAQIK DU BDDT, KD QIGQ ODP FGV KXX QIX RGFC MGOXZ DU RDDCK GRDHX QIX RDDCK EV UZDVQ. GVT EQ KQEMM EKV'Q XVDPAI. RDDCK GZX DHXZUMDBEVA DVQD QIX QGRMXK GVT QIX KDUGK GVT YGCEVA MEQQMX IXGLK PVTXZ QIX BEVTDBK. QIEK EK QIX MEHEVA-ZDDY DU QIX IDPKX DFFPLEXT RO QIX XYEVXVQ LZDUXKKDZ YEFIGXM HXZZXK-XHGVK, GVT IEK BEUX, YZK. LXQPVG XHGVK-HXZZXK, GVT QIXEZ GTDLQXT KDV, IGZZO NGYXK LDQQXZ-XHGVK-HXZZXK.

В качестве ответа отправьте одну строку с расшифрованным текстом, в последующих строках опишите действия по решению задачи.

Задача 1-4-4

Расшифруйте следующий текст:

XHP'T THP PWH HYITPHOJH-KL-DKC ITTFH VTICH LKE V XVPHE NKXFBH, VOC ZFTP TPIUFXVPH PWVP IO TKBH AVR, THXL-EHUXIJVPIOD KEDVOITBT JVBH IOPK HYITPHOJH KO PWIT UXVOHP VOC IBBHCIVPHXR GHDVO PERIOD PK DHP EIC KL HVJW KPWEH, HIPWHE GR TUVBBIOD PWHIE HONIEKOBHOPT AIPW EKFDW JKUIHT KL PWHBTHXNHT, KE GR BKEH CIEHJP BHVOT AWIJW WVECXR OHHC PK GH GHXVGKEHC. BKTP KL PWHB LVIXHC, VOC PWHIE DHOHPIJ XHDVJR AVT HEVTHC LEKB PWH FOINHETH LKEHNHE, GFP V LHA LKFOC TKBH AVR PK TFENINH VOC PK UEKUVDVPH. VLPHE VGKFP PWEHH GIXXIKO RHVET KL PWIT TKBHPIBHT SVOR, LEHQFHOPXR PHCIKFT LFDFH KL JVEOVXIPR VOC JVEODH, DKCLEHR AVPHEWKFTH IN AVT GKEO, IO BFECK, TKFPW CVMKPV, PK GXVOJWH, PWH AILH KL V JKODEHDVPIKOVX UEHVJWHE OVBHC GFORVO AVPHEWKFTH. XIMH HNER KPWEH JEHVPFEH KO PWH LVJH KL PWH HVERW, DKCLEHR AVT, GR GIEPWEIDWP, V TPFUHOCKFT GVCVTT, VXGHIP IO PWH TKBHAWVP OVEEKA PHJWOIJVX THOTH PWVP WH JKFXC PEVJH WIT VOJHTPER GVJM FU V XKOD XIOH KL TXIDWPXR XHTT WIDWXR HNKXNHC TPFUHOCKFT GVCVTTHT PK PWVP LIETP THXL-EHUXIJVPIOD DISBK-AWIJW, DINHO PWH OFBGHE VOC NVEIHPR KL IPT CHTJHOCVOPT, BIDWP ZFTPILIVGXR GH CHTJEIGHC VT PWH BKTP TPFUHOCKFT GVCVTT KL VXX PIBH. HNERKOH VOC HNERPWIOD PWVP AVTO'P V TPFUHOCKFT GVCVTT AVT CHVC.

В качестве ответа отправьте одну строку с расшифрованным текстом, в последующих строках опишите действия по решению задачи.

Задача 1-5

В городе запланировано строительство небоскреба. Главный архитектор утвердил проект, а затем стало известно, что количество выделенных ресурсов N не хватает на строительство полноценного здания. Архитектор указал, сколько ресурсов необходимо для строительства каждого этажа. В здании можно убрать любое количество подряд идущих этажей, исключая первый. Помогите архитектору определить максимальную высоту здания, которое можно построить из выделенных ресурсов (высота каждого этажа равна единице).

Входные данные: Первая строка содержит два числа, записанных через пробел N и M (N – количество выделенных ресурсов, M – количество этажей, определенное проектировщиками). На второй строке через пробел записано M чисел p_i (количество ресурсов, требуемое для постройки i -го этажа). Гарантируется, что $p_1 \leq N$ (ресурсы для строительства 1 этажа). Все числа положительные и не превосходят 10^6 .

Выходные данные: одно число – максимальная высота здания.

Задача 1-6

Компания Barmaley's Computing учла пожелания пользователей и выпустила вторую версию компьютера - Barmaglot-2.

Поскольку у сотрудников компании несколько странные представления о технологиях, компьютер остался довольно непривычным.

Во-первых, исходные

данные он читает с ленты, где могут быть записаны целые числа и латинские буквы.

Чтобы показать, что ввод данных закончен, ленту нужно просто оторвать.

Во-вторых, результаты вычислений компьютер печатает на точно такой же

ленте. Центральный процессор компьютера оснащён одним регистром; вместо оперативной памяти компьютер оснащён очередью и стеком неограниченного размера. Регистр, а также каждая ячейка очереди и стека могут содержать либо число, либо латинскую букву или пробел, либо специальное значение [BARMALEY], которое используется для обозначения логической лжи, при том что любое другое обозначает истину.

Программа для Barmaglot'a представляет собой строку символов, каждый из которых задаёт машинную команду; программа, состоящая из символов-команд, выполняется последовательно слева направо, кроме двух команд, которые могут нарушить эту последовательность.

все эти команды используют значение из регистра в качестве левого операнда, значение с вершины стека в качестве правого (оно при этом из стека извлекается), результат заносится обратно в регистр.

Наконец, команда " прекращает выполнение программы, при этом выполнение считается успешным. Если программа кончилась, не встретив эту команду, она завершается аварийно.

Пробелы в программе игнорируются.

Пример программы, которая печатает традиционную строчку "HELLO WORLD":

В вашем распоряжении оказался эмулятор данного компьютера (<http://ejudge.cs.msu.ru/barmaglot2/>) и требуется написать для него программу, которая по заданной последовательности целых неотрицательных чисел выводит отсортированную по возрастанию последовательность

Задача 1-7-1

Назовем интересным фрагментом последовательность из 10 подряд идущих не пустых строк (пустой называется строка, состоящая из любого числа пробелов и знака табуляции или же из одного символа перевода строки). Дан архив с файлами. Необходимо найти 30 самых часто-встречаемых интересных фрагментов в данном архиве. В качестве решения необходимо прислать файл solve.txt, в котором будут выведены в порядке убывания частоты встречаемости 30 фрагментов. При равенстве частот первым идет лексикографически наименьший. В файле ответа фрагменты разделяйте строкой "====", после всех фрагментов опишите процесс получения результата.

Задача 1-7-2

Назовем интересным фрагментом последовательность из 10 подряд идущих не пустых строк (пустой называется строка, состоящая из любого числа пробелов и знака табуляции или же из одного символа перевода строки). Дан архив с файлами. Необходимо найти 30 самых часто-встречаемых интересных фрагментов в данном архиве. В качестве решения необходимо прислать файл solve.txt, в котором будут выведены в порядке убывания частоты встречаемости 30

фрагментов. При равенстве частот первым идет лексикографически наименьший. В файле ответа фрагменты разделяйте строкой "=====", после всех фрагментов опишите процесс получения результата.

Задача 1-7-3

Назовем интересным фрагментом последовательность из 10 подряд идущих не пустых строк (пустой называется строка, состоящая из любого числа пробелов и знака табуляции или же из одного символа перевода строки). Дан архив с файлами. Необходимо найти 30 самых часто-встречаемых интересных фрагментов в данном архиве. В качестве решения необходимо прислать файл solve.txt, в котором будут выведены в порядке убывания частоты встречаемости 30 фрагментов. При равенстве частот первым идет лексикографически наименьший. В файле ответа фрагменты разделяйте строкой "=====", после всех фрагментов опишите процесс получения результата.

Задача 1-7-4

Назовем интересным фрагментом последовательность из 10 подряд идущих не пустых строк (пустой называется строка, состоящая из любого числа пробелов и знака табуляции или же из одного символа перевода строки). Дан архив с файлами. Необходимо найти 30 самых часто-встречаемых интересных фрагментов в данном архиве. В качестве решения необходимо прислать файл solve.txt, в котором будут выведены в порядке убывания частоты встречаемости 30 фрагментов. При равенстве частот первым идет лексикографически наименьший. В файле ответа фрагменты разделяйте строкой "=====", после всех фрагментов опишите процесс получения результата.

Задача 1-8

Составьте программу, принимающую на вход натуральное число $N \leq 32767$ и N целых чисел a_i ($i = 1, 2, \dots, N$). Любое число a_i не превосходит по модулю 32767. Известно, что входная последовательность обязательно обладает таким свойством, что в ней можно указать пару элементов a_m и a_k ($m \leq k$) таких, что, если их поменять местами, то последовательность a станет возрастающей.

Ваша программа должна найти, какие элементы следует переставить, чтобы последовательность стала возрастающей, и вывести их номера. Первым следует вывести меньший номер.

Указание: Последовательность не должна храниться в памяти целиком. Допускается не более чем один проход по последовательности. Объем используемой памяти должен быть мал и не должен зависеть от длины последовательности. Ваша программа должна экономно расходовать память и не производить лишних действий и вычислений.

Задача 2-1-1

Петя хочет зашифровать свой буквенный пароль, чтобы никто другой не смог его прочитать. Сначала он пробовал шифровать

перемножением, но потом понял, что этот способ не годится. В Сети ему попала статья про шифрованием полиномом.

Он прочитал её по диагонали и решил, что этот способ ему подходит. Закодированный Петей пароль s представляет собой число,

являющееся значением полинома $X(s) = s[1] \cdot 24^{n-1} + s[2] \cdot 24^{n-2} + \dots + s[n-1] \cdot 24 + s[n]$, где $s[i]$ - код буквы пароля s , стоящей на

i -ом месте. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется единицей, В – двойкой и т. д. Помогите Пете подобрать

пароль, если им записано заданное число 28850.

Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в

алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого пароля.

Задача 2-1-2

Петя хочет зашифровать свой буквенный пароль, чтобы никто другой не смог его прочитать. Сначала он пробовал шифровать

перемножением, но потом понял, что этот способ не годится. В Сети ему попала статья про шифрованием полиномом.

Он прочитал её по диагонали и решил, что этот способ ему подходит. Закодированный Петей пароль s представляет собой число,

являющееся значением полинома $X(s) = s[1] \cdot 24^{n-1} + s[2] \cdot 24^{n-2} + \dots + s[n-1] \cdot 24 + s[n]$, где $s[i]$ - код буквы пароля s , стоящей на

i -ом месте. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется единицей, В – двойкой и т. д. Помогите Пете подобрать

пароль, если им записано заданное число 56498.

Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в

алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого пароля.

Задача 2-1-3

Петя хочет зашифровать свой буквенный пароль, чтобы никто другой не смог его прочитать. Сначала он пробовал шифровать

перемножением, но потом понял, что этот способ не годится. В Сети ему попала статья про шифрованием полиномом.

Он прочитал её по диагонали и решил, что этот способ ему подходит. Закодированный Петей пароль s представляет собой число,

являющееся значением полинома $X(s) = s[1]*24^{n-1} + s[2]*24^{n-2} + \dots + s[n-1]*24 + s[n]$, где $s[i]$ - код буквы пароля s , стоящей на

i -ом месте. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется единицей, В – двойкой и т. д. Помогите Пете подобрать

пароль, если им записано заданное число 346226.

Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в

алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого пароля.

Задача 2-1-4

Петя хочет зашифровать свой буквенный пароль, чтобы никто другой не смог его прочитать. Сначала он пробовал шифровать

перемножением, но потом понял, что этот способ не годится. В Сети ему попала статья про шифрованием полиномом.

Он прочитал её по диагонали и решил, что этот способ ему подходит. Закодированный Петей пароль s представляет собой число,

являющееся значением полинома $X(s) = s[1]*24^{n-1} + s[2]*24^{n-2} + \dots + s[n-1]*24 + s[n]$, где $s[i]$ - код буквы пароля s , стоящей на

i -ом месте. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется единицей, В – двойкой и т. д. Помогите Пете подобрать

пароль, если им записано заданное число 111794.

Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в

алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого пароля.

Задача 2-2-1

Петя хочет зашифровать свой буквенный пароль, чтобы никто другой не смог его прочитать. Сначала он пробовал шифровать

перемножением, но потом понял, что этот способ не годится. В Сети ему попала статья про шифрованием полиномом.

Он прочитал её по диагонали и решил, что этот способ ему подходит. Закодированный Петей пароль s представляет собой число,

являющееся значением полинома $X(s) = s[1] \cdot 24^{n-1} + s[2] \cdot 24^{n-2} + \dots + s[n-1] \cdot 24 + s[n]$, где $s[i]$ - код буквы пароля s , стоящей на

i -ом месте. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется единицей, В – двойкой и т. д. Помогите Пете подобрать

пароль, если им записано заданное число 9571766450.

Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в

алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого пароля. После ответа, опишите как он был получен.

Задача 2-2-2

Петя хочет зашифровать свой буквенный пароль, чтобы никто другой не смог его прочитать. Сначала он пробовал шифровать

перемножением, но потом понял, что этот способ не годится. В Сети ему попала статья про шифрованием полиномом.

Он прочитал её по диагонали и решил, что этот способ ему подходит. Закодированный Петей пароль s представляет собой число,

являющееся значением полинома $X(s) = s[1] \cdot 24^{n-1} + s[2] \cdot 24^{n-2} + \dots + s[n-1] \cdot 24 + s[n]$, где $s[i]$ - код буквы пароля s , стоящей на

i -ом месте. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется единицей, В – двойкой и т. д. Помогите Пете подобрать

пароль, если им записано заданное число 18744709298.

Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в

алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого пароля. После ответа, опишите как он был получен.

Задача 2-2-3

Петя хочет зашифровать свой буквенный пароль, чтобы никто другой не смог его прочитать. Сначала он пробовал шифровать

перемножением, но потом понял, что этот способ не годится. В Сети ему попала статья про шифрованием полиномом.

Он прочитал её по диагонали и решил, что этот способ ему подходит. Закодированный Петей пароль s представляет собой число,

являющееся значением полинома $X(s) = s[1] \cdot 24^{n-1} + s[2] \cdot 24^{n-2} + \dots + s[n-1] \cdot 24 + s[n]$, где $s[i]$ - код буквы пароля s , стоящей на

i -ом месте. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется единицей, В – двойкой и т. д. Помогите Пете подобрать

пароль, если им записано заданное число 27917652146.

Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в

алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого пароля. После ответа, опишите как он был получен.

Задача 2-2-4

Петя хочет зашифровать свой буквенный пароль, чтобы никто другой не смог его прочитать. Сначала он пробовал шифровать

перемножением, но потом понял, что этот способ не годится. В Сети ему попала статья про шифрованием полиномом.

Он прочитал её по диагонали и решил, что этот способ ему подходит. Закодированный Петей пароль s представляет собой число,

являющееся значением полинома $X(s) = s[1] \cdot 24^{n-1} + s[2] \cdot 24^{n-2} + \dots + s[n-1] \cdot 24 + s[n]$, где $s[i]$ - код буквы пароля s , стоящей на

i -ом месте. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется единицей, В – двойкой и т. д. Помогите Пете подобрать

пароль, если им записано заданное число 37090594994.

Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в

алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого пароля. После ответа, опишите как он был получен.

Задача 2-3

Написать программу, которая

по заданному целому положительному числу до 1000000 получает минимальную по длине формулу дающую это число. Формула может состоять из символов '1', '2', '+', '*'.

Задача 2-4-1

На прямоугольном поле расположены лампочки, некоторые из которых включены (обозначаются единицей), некоторые выключены (обозначаются нулем).

За одну операцию можно выбрать лампочку и нажать на неё. После нажатия состояние лампочки и соседних с ней инвертируется. Ваша

Задача 2-по заданному в файле

состоянию получить все включенные лампочки.

В качестве ответа отправьте в первой строке число операций или -1, если включить все лампочки нельзя. В последующих строках сами операции - номер строки и номер столбца (нумерация начинается с 1) выбранной лампочки. После ответа опишите действия по решению задачи.

Например, для конфигурации

1100

1100

0000

0011

Ответом будут две операции 2 4 и 4 1

Число операции должно быть по возможности минимальным

Задача 2-4-2

На прямоугольном поле расположены лампочки, некоторые из которых включены (обозначаются единицей), некоторые выключены (обозначаются нулем).

За одну операцию можно выбрать лампочку и нажать на неё. После нажатия состояние лампочки и соседних с ней инвертируется. Ваша

Задача 2-по заданному в файле

состоянию получить все включенные лампочки.

В качестве ответа отправьте в первой строке число операций или -1, если включить все лампочки нельзя. В последующих строках сами операции - номер строки и номер столбца (нумерация начинается с 1) выбранной лампочки. После ответа опишите действия по решению задачи.

Например, для конфигурации

1100

1100

0000

0011

Ответом будут две операции 2 4 и 4 1

Число операции должно быть по возможности минимальным

Задача 2-4-3

На прямоугольном поле расположены лампочки, некоторые из которых включены (обозначаются единицей), некоторые выключены (обозначаются нулем).

За одну операцию можно выбрать лампочку и нажать на неё. После нажатия состояние лампочки и соседних с ней инвертируется. Ваша

Задача 2-по заданному в файле

состоянию получить все включенные лампочки.

В качестве ответа отправьте в первой строке число операций или -1, если включить все лампочки нельзя. В последующих строках сами операции - номер строки и номер столбца (нумерация начинается с 1) выбранной лампочки. После ответа опишите действия по решению задачи.

Например, для конфигурации

1100

1100

0000

0011

Ответом будут две операции 2 4 и 4 1

Число операции должно быть по возможности минимальным

Задача 2-4-4

На прямоугольном поле расположены лампочки, некоторые из которых включены (обозначаются единицей), некоторые выключены (обозначаются нулем).

За одну операцию можно выбрать лампочку и нажать на неё. После нажатия состояние лампочки и соседних с ней инвертируется. Ваша

Задача 2-по заданному в файле

состоянию получить все включенные лампочки.

В качестве ответа отправьте в первой строке число операций или -1, если включить все лампочки нельзя. В последующих строках сами операции - номер строки и номер столбца (нумерация начинается с 1) выбранной лампочки. После ответа опишите действия по решению задачи.

Например, для конфигурации

1100

1100

0000

0011

Ответом будут две операции 2 4 и 4 1

Число операции должно быть по возможности минимальным

Задача 2-5

Важный раздел информатики - это теория помехоустойчивого кодирования. Не секрет, что при передаче данных в любой среде передачи в данных могут возникать искажения или ошибки. Необходимо вовремя распознать и по возможности скорректировать ошибки, возникшие при передаче.

Мы будем считать, что данные передаются в бинарном формате. Данные передаются порциями по N бит, при этом гарантируется, что принято будет ровно столько бит, сколько передано. Однако, при передаче каждой порции из N бит в нее может быть внесена не более чем одна ошибка, то есть не более чем один бит может поменять свое значение либо с 1 на 0, либо с 0 на 1. Возможно, что передача N бит пройдет и без ошибок, но гарантируется, что более одной ошибки в каждой порции возникнуть не может.

Задача помехоустойчивого кодирования заключается в том, что входные бинарные данные преобразовываются в закодированные бинарные данные таким образом, чтобы в случае ошибки при передаче очередной порции данных можно было бы однозначно определить и исправить ошибку. Размер закодированных данных при этом неизбежно будет больше размера исходных данных. Эффективностью кодирования мы назовем отношение размера исходных данных к размеру закодированных данных в худшем возможном случае. Эффективность кодирования не может быть больше 1.

Разработайте алгоритм и напишите программу, которая реализовывает наиболее эффективное по вашему мнению помехоустойчивое кодирование и декодирование данных. Размер одной порции данных будем считать фиксированным и равным 8.

На вход программе подается либо поток данных для кодирования, либо поток данных для декодирования, а именно:

на стандартном потоке ввода (т. е., например, с клавиатуры) программе сначала задается число 0, если программа должна закодировать данные, либо число 1, если программа должна декодировать данные.

Далее программе на вход подается размер данных (то есть число бит в нем). Первые два числа каждое задаются на отдельной строке. Далее на вход подаются сами данные. Цифра '0' соответствует нулевому биту, цифра '1' соответствует единичному биту. Все прочие символы (пробелы, переводы строк) должны игнорироваться.

В случае кодирования данных программа должна вывести на стандартный поток вывода (экран) закодированные данные. Размер закодированных данных должен быть кратен размеру порции данных (при необходимости дополняется нулями). При желании вы можете разделять выводимые символы '0' или '1' символами пробела или перевода строки.

В случае декодирования данных программа должна вывести на стандартный поток вывода (экран) декодированные данные. Если в закодированных данных были обнаружены ошибки, они должны быть исправлены. При желании вы можете разделять выводимые символы '0' или '1' символами пробела или перевода строки.

Если в качестве первого числа (флага кодирования/декодирования) ваша программа считала число -1, программа должна немедленно завершить работу с кодом завершения 0.

Программа в режиме декодирования данных должна раскодировать данные, закодированные той же самой вашей программой в режиме кодирования данных. Никаких других данных в других форматах для раскодирования подаваться не будет.

Ваша задача: придумать и реализовать такой алгоритм кодирования и декодирования, который имел бы наилучшую эффективность, то есть добавлял бы как можно меньше избыточности к данным.

Задача 2-6

Компания Barmaley's Computing учла пожелания пользователей и выпустила вторую версию компьютера - Barmaglot-2.

Поскольку у сотрудников компании несколько странные представления о технологиях, компьютер остался довольно непривычным.

Во-первых, исходные

данные он читает с ленты, где могут быть записаны целые числа и латинские буквы.

Чтобы показать, что ввод данных закончен, ленту нужно просто оторвать.

Во-вторых, результаты вычислений компьютер печатает на точно такой же

ленте. Центральный процессор компьютера оснащён одним регистром; вместо оперативной памяти компьютер оснащён очередью и стеком неограниченного размера. Регистр, а также каждая ячейка очереди и стека могут содержать либо число, либо латинскую букву или пробел, либо специальное значение [BARMALEY], которое используется для обозначения логической лжи, при том что любое другое обозначает истину.

Программа для Barmaglot'a представляет собой строку символов, каждый из которых задаёт машинную команду; программа, состоящая из символов-команд, выполняется последовательно слева направо, кроме двух команд, которые могут нарушить эту последовательность.

все эти команды используют значение из регистра в качестве левого операнда, значение с вершины стека в качестве правого (оно при этом из стека извлекается), результат заносится обратно в регистр.

Наконец, команда " прекращает выполнение программы, при этом выполнение считается успешным. Если программа кончилась, не встретив эту команду, она завершается аварийно.

Пробелы в программе игнорируются.

Пример программы, которая печатает традиционную строчку “HELLO WORLD”:

В вашем распоряжении оказался эмулятор данного компьютера (<http://ejudge.cs.msu.ru/barmaglot2/>) и требуется написать для него программу, которая по заданной последовательности целых неотрицательных чисел выводит отсортированную по убыванию последовательность

Задача 2-7

Петя увлекся ASCII графикой (<https://ru.wikipedia.org/wiki/ASCII-графика>) и нарисовал в этом стиле цифры от 0 до 9 (архив с исходными цифрами). Размер каждой цифрой получился 50x80 символов. Но на компьютер попал вирус, который внес в файлы некоторый шум. В каждой строке вирус выбирал 5 любых символов и заменял их на пробелы (при этом допустима замена пробела на пробел) и после этого выбирал 10 любых символов и заменял их на любые другие символы. Требуется написать программу, которая на вход принимает зашумленный файл и выдает единственное число – цифру, которая была в оригинальном файле.

Пример цифры после работы вируса, решение должно вывести одно число 0.

Задача 2-8

Составьте программу, принимающую на вход натуральное число $N \leq 32767$ и N целых чисел a_i ($i = 1, 2, \dots, N$). Любое число a_i не превосходит по модулю 32767. Известно, что входная последовательность обязательно обладает таким свойством, что в ней можно указать пару элементов a_m и a_k ($m \leq k$) таких, что, если их поменять местами, то последовательность a станет выпуклой.

Ваша программа должна найти, какие элементы следует переставить, чтобы последовательность стала выпуклой (https://ru.wikipedia.org/wiki/Выпуклая_функция), и вывести их номера. Первым следует вывести меньший номер.

Указание: Последовательность не должна храниться в памяти целиком. Допускается не более чем один проход по последовательности. Объём используемой памяти должен быть мал и не должен зависеть от длины последовательности. Ваша программа должна экономно расходовать память и не производить лишних действий и вычислений.

Задания отборочного этапа, 5-9 классы.

**Задача 1-1

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 273. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова.

**Задача 1-2

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 165. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова.

**Задача 1-3

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 230. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова.

**Задача 1-4

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 238. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова.

****Задача 2-1**

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 2348346. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова. После ответа, опишите как он был получен.

****Задача 2-2**

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 1677390. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова. После ответа, опишите как он был получен.

****Задача 2-3**

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 3913910. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке или пустой ответ, если по данному числу невозможно получить никакого слова. После ответа, опишите как он был получен.

****Задача 2-4**

Петя зашифровал свой буквенный пароль, чтобы никто другой не смог его прочитать, но потом понял, что даже сам не может этого сделать. Закодированный пароль представляет собой число, являющееся произведением кодов всех букв. Буквы кодируются подряд по своему расположению в английском алфавите, причем буква А кодируется двойкой, В – тройкой и т. д. Помогите Пете подобрать пароль, если им записано число 1118260. Отправьте текст, где выписаны всевозможные пароли в нижнем регистре через символ перевода строки в алфавитном порядке

или пустой ответ, если по данному числу невозможно получить никакого слова. После ответа, опишите как он был получен.

**Задача 3

Вася играет в кубики. Все кубики у Васи одинакового размера 1см^3 . Кубики у Васи двух видов: либо непрозрачные чёрные, поглощающие весь падающий на них свет, либо прозрачные бесцветные, пропускающие световые лучи. Вася собирает из маленьких кубиков большой куб размером $N \times N \times N \text{ см}^3$ ($1 < N < 250$). При сборке кубики ставятся вплотную друг к другу, так, что не остаётся пустот. Собрав куб, Вася глядит сквозь него на солнце, держа его в руке так, чтобы взгляд был направлен перпендикулярно грани куба, ближней к его лицу. Васе интересно, как свет проходит через куб, какая картинка складывается в проекции собранного им куба. Когда солнце зашло, Вася понял, что узнать какая картинка может получиться можно написав программу.

На вход подаётся целое число N и описание внутреннего строения куба. Это описание составлено так, что каждый непрозрачный (чёрный) кубик обозначается нулём, а прозрачный (бесцветный) - единицей. Первым кубиком в описании является тот, который находится в левой нижней ближней к Васе вершине. Эта вершина - начало координат. Система координат правая прямоугольная. Оси координат (абсцисса, ордината и аппликата) идут так, чтобы любое ребро было параллельно одной из осей. Куб лежит в первом октанте системы координат. Ось аппликат идёт в направлении от Васи к солнцу (вдоль неё Вася смотрит на куб). Итак, в описании первым является кубик с координатами $(0, 0, 0)$. За ним следуют остальные N^2-1 кубиков с нулевой аппlikатой, образующие нулевой (ближний к Васе) слой куба: $(1, 0, 0), (2, 0, 0), \dots, (N-1, 0, 0), (1, 1, 0), \dots, (N-1, 1, 0), \dots, (N-1, N-1, 0)$. Далее указываются значения кубиков с аппlikатой 1, образующие следующий слой: $(1, 0, 1), (2, 0, 1), \dots, (N-1, N-1, 1)$. Они упорядочены по тому же принципу, что и кубики нулевого слоя. За первым слоем следует 2-й, ... $N-1$ -й. В описании строения куба значения кубиков идут слитно без разделителей.

Программа должна вывести параллельную ортогональную проекцию куба на плоскость Oxy . При её построении считается, что лучи идут только параллельно оси Oz . Отражений и преломлений в стенках кубиков нет. Прозрачный кубик пропускает лучи, а непрозрачный поглощает без остатка. В выводе программы 0 должен обозначать непрозрачный (чёрный) квадратик проекции, 1 -- прозрачный (бесцветный) квадратик. Порядок квадратиков при выводе проекции соответствует следующему ряду координат: $(0, 0), (1, 0), \dots, (N-1, 0), (0, 1), \dots, (N-1, 1), \dots, (N-1, N-1)$.

**Задача 4-1

Расшифруйте следующий текст:

HOA ILF YV FPJ LVGPOAFJX BOGQMOFJAC IH FPJ LVHOCPIVOBRI JYX IH FPJ MJCFJAV CKYAOR OAW IH
FPJ EOROZN RYJC O CWORR LVAJEOAXJX NJRRIM CLV. IABYFYVE FPYC OF O XYCFOVGJ IH AILEPRN
VYVJFN-FMI WYRRYIV WYRJY YC OV LFFJARN YVCYEVYHYGOVF RYFFRJ BRJ EAJJV KROVJF MPICJ OKJ-
XJCGJVXJX RYHJ HIAWC OAJ CI OWODYVERN KAYWYFYJSJ PPOF FPJN CFYRR FPYVQ XYEYFOR MOFGPJX
OAJ O KAJFFN VJOF YXJO. FPYC KROVJF ПОС - IA AOFPIA ПОХ - O KAIBRJW, МРУГР МОС FPYC: WICF IH

FPJ KJIKRJ IV YF MJAJ LVPOKKN HIA KAJFFN WLGP IH FPJ FYWJ. WOVN CIRLFYIVC MJAJ CLEEJCFJX HIA
FPYC KAIBRJW, BLF WICF IH FPJCJ MJAJ ROAEJRN GIVGJAVJX MYFP FPJ WISJWJVFC IH CWORR EAJJV
KYJGJC IH KOKJA, MPYGP YC IXX BJGOLCJ IV FPJ MPIRJ YF MOCV'F FPJ CWORR EAJJV KYJGJC IH KOKJA
FPOF MJAJ LVPOKKN.

В качестве ответа отправьте одну строку с расшифрованным текстом, в последующих строках опишите действия по решению задачи.

****Задача 4-2**

Расшифруйте следующий текст:

XB. NRF XBM. FTBMAJZ, LG RTXKJB GLTB, EBUPJI FBUPJ, WJBJ EBLTF IL MNZ IHNI IHJZ WJBJ EJBGSIAZ
RLBXNA, IHNRY ZLT PJBZ XTSH. IHJZ WJBJ IHJ ANMI EJLEAJ ZLT'F JQEJSI IL KJ URPLAPJF UR NRZIHURC
MIBNRCJ LB XZMIJBULTM, KJSNTMJ IHJZ DTMI FUFRI HIAF WUIH MTSH RLRMJRMJ. XB. FTBMAJZ WNM
IHJ FUBJSILB LG N GUBX SNAAJF CBTRRURCM, WHUSH XNFJ FBUAAM. HJ WNM N KUC, KJGZ XNR WUIH
HNBFAZ NRZ RJSY, NAIHLTCH HJ FUF HNPJ N PJBZ ANBCJ XTMINSHJ. XBM. FTBMAJZ WNM IHUR NRF
KALRFJ NRF HNF RJNBAZ IWUSJ IHJ TMTNA NXLTRE LG RJSY, WHUSH SNXJ UR PJBZ TMJGTA NM MHJ
MEJRI ML XTSH LG HJB IUXJ SBNRURC LPJB CNBFJR GJRSJM, MEZURC LR IHJ RJUCHKLBM. IHJ FTBMAJZM
HNF N MXNAA MLR SNAAJF FTFJZ NRF UR IHJUB LEURULR IHJBJ WNM RL GURJB KLZ NRZWHJBJ.

В качестве ответа отправьте одну строку с расшифрованным текстом, в последующих строках опишите действия по решению задачи.

****Задача 4-3**

Расшифруйте следующий текст:

XHXZO EVFI DU BGMM KLGFX EK FDHXZT RO G RDDCFGKX. XGFI RDDCFGKX IGK KEW KIXMHXK, ADEVA
GMYDKQ QD QIX FXEMEVA. KDYX RDDCKIXMHXK GZX KQGFCXT QD QIX RZEY BEQI IGZTRGFC RDDCK:
KFEXVFX, YGQIK, IEKQDZO, GVT XHXZOQIEVA XMKX. DQIXZ KIXMHXK IGHX QBD MGOXZK DU
LGLXZRGFC KFEXVFX UEFQEDV, BEQI QIX RGFC MGOXZ DU RDDCK LZDLLXT PL DV DMT QEKKPX RDWXX
DZ MXVAQIK DU BDDT, KD QIGQ ODP FGV KXX QIX RGFC MGOXZ DU RDDCK GRDHX QIX RDDCK EV
UZDVQ. GVT EQ KQEMM EKV'Q XVDPAI. RDDCK GZX DHXZUMDBEVA DVQD QIX QGRMXK GVT QIX
KDUGK GVT YGCEVA MEQQMX IXGLK PVTXZ QIX BEVTDBK. QIEK EK QIX MEHEVA-ZDDY DU QIX IDPKX
DFFPLEXT RO QIX XEYVXVQ LZDUXKKDZ YEFIXM HXZZXK-XHGVK, GVT IEK BEUX, YZK. LXQPVEG XHGVK-
HXZZXK, GVT QIXEZ GTDLQXT KDV, IGZZO NGYXK LDQQXZ-XHGVK-HXZZXK.

В качестве ответа отправьте одну строку с расшифрованным текстом, в последующих строках опишите действия по решению задачи.

****Задача 4-4**

Расшифруйте следующий текст:

XHP'T THP PWH HYITPHOJH-KL-DKC ITTFH VTICH LKE V XVPHE NKXFBH, VOC ZFTP TPIUFXVPH PWVP IO
TKBH AVR, THXL-EHUXIJVPIOD KEDVOITBT JVBH IOPK HYITPHOJH KO PWIT UXVOHP VOC IBBHCIVPHXR
GHDVO PERIOD PK DHP EIC KL HVJW KPWHE, HIPWHE GR TUVBBIOD PWHIE HONIEKOBHOPT AIPW
EKFDW JKUIHT KL PWHBTHXNHT, KE GR BKEH CIEHJP BHVOT AWIJW WVECXR OHHC PK GH
GHXVGKEHC. BKTP KL PWHB LVIXHC, VOC PWHIE DHOHPJ XHDVJR AVT HEVTHC LEKB PWH FOINHETH
LKEHNHE, GFP V LHA LKFOC TKBH AVR PK TFENINH VOC PK UEKUVDVPH. VLPHE VGKFP PWEHN GIXXIKO
RHVET KL PWIT TKBHPIBHT SVOR, LEHQFHOPXR PHCIKFT LFDFH KL JVEOVXIPR VOC JVEOVDH, DKCLEHR
AVPHEWKFTH IN AVT GKEO, IO BFECK, TKFPW CVMKPV, PK GXVOJWH, PWH AILH KL V
JKODEHNVPIKOVX UEHVJWHE OVBHC GFORVO AVPHEWKFTH. XIMH HNNER KPWHE JEHVPFEH KO PWH
LVJH KL PWH HVEPW, DKCLEHR AVT, GR GIEPWEIDWP, V TPFUHOCKFT GVCVTT, VXGHP IO PWH
TKBHAWVP OVEEKA PHJWOIJVX THOTH PWVP WH JKFXC PEVJH WIT VOJHTPER GVMJ FU V XKOD XIOH
KL TXIDWPXR XHTT WIDWXR HNKXNHC TPFUHOCKFT GVCVTTHT PK PWVP LIETP THXL-EHUXIJVPIOD
DISBK-AWIJW, DINHO PWH OFBGHE VOC NVEIHPR KL IPT CHTJHOCVOPT, BIDWP ZFTPILIVGXR GH
CHTJEIGHC VT PWH BKTP TPFUHOCKFT GVCVTT KL VXX PIBH. HNNERKON VOC HNNERPWIOD PWVP
AVTO'P V TPFUHOCKFT GVCVTT AVT CHVC.

В качестве ответа отправьте одну строку с расшифрованным текстом, в последующих строках опишите действия по решению задачи.

****Задача 5**

В городе запланировано строительство небоскреба. Главный архитектор утвердил проект, а затем стало известно, что количество выделенных ресурсов N не хватает на строительство полноценного здания. Архитектор указал, сколько ресурсов необходимо для строительства каждого этажа. В здании можно убрать любое количество подряд идущих этажей, исключая первый. Помогите архитектору определить максимальную высоту здания, которое можно построить из выделенных ресурсов (высота каждого этажа равна единице).

Входные данные: Первая строка содержит два числа, записанных через пробел N и M (N – количество выделенных ресурсов, M – количество этажей, определенное проектировщиками). На второй строке через пробел записано M чисел p_i (количество ресурсов, требуемое для постройки i -го этажа). Гарантируется, что $p_1 \leq N$ (ресурсы для строительства 1 этажа). Все числа положительные и не превосходят 10^6 .

Выходные данные: одно число – максимальная высота здания.

****Задача 6**

Компания Barmaley's Computing учла пожелания пользователей и выпустила вторую версию компьютера - Barmaglot-2.

Поскольку у сотрудников компании несколько странные представления о технологиях, компьютер остался довольно непривычным.

Во-первых, исходные

данные он читает с ленты, где могут быть записаны целые числа и латинские буквы.

Чтобы показать, что ввод данных закончен, ленту нужно просто оторвать.

Во-вторых, результаты вычислений компьютер печатает на точно такой же

ленте. Центральный процессор компьютера оснащён одним регистром; вместо оперативной памяти компьютер оснащён очередью и стеком неограниченного размера. Регистр, а также каждая ячейка очереди и стека могут содержать либо число, либо латинскую букву или пробел, либо специальное значение [BARMALEY], которое используется для обозначения логической лжи, при том что любое другое обозначает истину.

Программа для Barmaglot'a представляет собой строку символов, каждый из которых задаёт машинную команду; программа, состоящая из символов-команд, выполняется последовательно слева направо, кроме двух команд, которые могут нарушить эту последовательность.

все эти команды используют значение из регистра в качестве левого операнда, значение с вершины стека в качестве правого (оно при этом из стека извлекается), результат заносится обратно в регистр.

Наконец, команда " прекращает выполнение программы, при этом выполнение считается успешным. Если программа кончилась, не встретив эту команду, она завершается аварийно.

Пробелы в программе игнорируются.

Пример программы, которая печатает традиционную строчку "HELLO WORLD":

В вашем распоряжении оказался эмулятор данного компьютера (<http://ejudge.cs.msu.ru/barmaglot2/>) и требуется написать для него программу, которая по заданной последовательности трёх целых неотрицательных чисел выводит отсортированную последовательность

****Задача 7-1**

Назовем интересным фрагментом последовательность из 10 подряд идущих не пустых строк (пустой называется строка, состоящая из любого числа пробелов и знака табуляции или же из одного символа перевода строки). Дан архив с файлами. Необходимо найти 30 самых часто-встречаемых интересных фрагментов в данном архиве. В качестве решения необходимо прислать файл solve.txt, в котором будут выведены в порядке убывания частоты встречаемости 30 фрагментов. При равенстве частот первым идет лексикографически наименьший. В файле ответа фрагменты разделяйте строкой "====", после всех фрагментов опишите процесс получения результата.

****Задача 7-2**

Назовем интересным фрагментом последовательность из 10 подряд идущих не пустых строк (пустой называется строка, состоящая из любого числа пробелов и знака табуляции или же из одного символа перевода строки). Дан архив с файлами. Необходимо найти 30 самых часто-встречаемых интересных фрагментов в данном архиве. В качестве решения необходимо прислать файл solve.txt, в котором будут выведены в порядке убывания частоты встречаемости 30 фрагментов. При равенстве частот первым идет лексикографически наименьший. В файле ответа фрагменты разделяйте строкой "====", после всех фрагментов опишите процесс получения результата.

****Задача 7-3**

Назовем интересным фрагментом последовательность из 10 подряд идущих не пустых строк (пустой называется строка, состоящая из любого числа пробелов и знака табуляции или же из одного символа перевода строки). Дан архив с файлами. Необходимо найти 30 самых часто-встречаемых интересных фрагментов в данном архиве. В качестве решения необходимо прислать файл solve.txt, в котором будут выведены в порядке убывания частоты встречаемости 30 фрагментов. При равенстве частот первым идет лексикографически наименьший. В файле ответа фрагменты разделяйте строкой "====", после всех фрагментов опишите процесс получения результата.

****Задача 7-4**

Назовем интересным фрагментом последовательность из 10 подряд идущих не пустых строк (пустой называется строка, состоящая из любого числа пробелов и знака табуляции или же из одного символа перевода строки). Дан архив с файлами. Необходимо найти 30 самых часто-

встречаемых интересных фрагментов в данном архиве. В качестве решения необходимо прислать файл solve.txt, в котором будут выведены в порядке убывания частоты встречаемости 30 фрагментов. При равенстве частот первым идет лексикографически наименьший. В файле ответа фрагменты разделяйте строкой "====", после всех фрагментов опишите процесс получения результата.

****Задача 8**

Составьте программу, принимающую на вход натуральное число $N \leq 32767$ и N целых чисел a_i ($i = 1, 2, \dots, N$). Любое число a_i не превосходит по модулю 32767. Известно, что входная последовательность обязательно обладает таким свойством, что в ней можно указать пару элементов a_m и a_k ($m \leq k$) таких, что, если их поменять местами, то последовательность a станет возрастающей.

Ваша программа должна найти, какие элементы следует переставить, чтобы последовательность стала возрастающей, и вывести их номера. Первым следует вывести меньший номер. Указание: Последовательность не должна храниться в памяти целиком. Допускается не более чем один проход по последовательности. Объём используемой памяти должен быть мал и не должен зависеть от длины последовательности. Ваша программа должна экономно расходовать память и не производить лишних действий и вычислений.

Ответы на задания заочного этапа олимпиады школьников "Ломоносов" по информатике 2015/16 учебный год

*Решения отборочного тура 5-9 классов не отличаются от решений
первого отборочного тура 10-11 классов.*

Первый тур 10-11 классы

Задание 1

вариант 1.1) 273:

bfl
blf
fbl
flb
lbf
lfb
lt
tl

вариант 1.2) 165:

bdj
bjd
dbj
djb
jbd
jdb
jn
nj

вариант 1.3) 230:

adv
avd
dav
dva
iv
vad
vda

vi

вариант 1.4) 238:

afp
apf
far
fra
mp
raf
pfa
pm

Схема решения:

- 1) Следует найти все различные разложения заданного числа на множители, величина которых от 2 до 27. Каждое найденное разложение даёт набор букв, из которых можно составить пароль.
- 2) Для каждого найденного набора букв следует построить все перестановки. Каждая перестановка определяет порядок букв в пароле, т. е. одну из строк ответа. Если множитель дважды участвует в разложении, будут порождаться дублирующие строки в ответе.
- 3) Все строки ответа следует упорядочить по алфавиту. Дублирующие строки ответа, если таковые будут, можно отсеять на этом этапе.

Задание 2

Задача аналогично задаче 1, но требовалось отправить все варианты пароля и привести обоснование, например, программу генерации всех паролей.

2.1) 2348346:

$$2*3*7*11*13*17*23 = 2348346 \quad (7! + 5*6! + 2*5! = 8880 \text{ вариантов})$$

2.2) 1677390:

$$2*3*5*11*13*17*23 = 1677390 \quad (7! + 5*6! + 2*5! = 8880 \text{ вариантов})$$

2.3) 3913910:

$$2*5*7*11*13*17*23 = 3913910 \quad (7! + 4*6! = 7920 \text{ вариантов})$$

2.4) 1118260:

$$2*2*5*11*13*17*23 = 1118260 \quad (7!/2 + 4*6! + 4*5! = 5880 \text{ вариантов})$$

Решение:

Программу, обосновывающую решение, можно составить, опираясь на схему из решения задачи 1.

Задание 3

Пример решения на языке C++, в задаче нельзя было хранить весь куб в памяти:

```
#include <cstdio>
using namespace std;

int a[250][250];
int main(int argc, char *argv[])
{
    int n;
    scanf("%d\n", &n);

    for (int t = 0; t < n; ++t) {
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < n; ++j) {
                a[i][j] |= (getc(stdin) - '0') ^ 1;
            }
        }
    }

    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            printf("%d", (a[i][j] ^ 1));
        }
    }
    return 0;
}
```

Дополнение к решению: если вести подсчёт непрозрачных “пикселей” в проекции, то это позволит останавливаться досрочно, если по текущим вычислениям станет ясно, что вся проекция непрозрачна.

Задание 4

Во всех вариантах используется шифр замены
вариант 4.1:

Far out in the uncharted backwaters of the unfashionable end of the western spiral arm of the Galaxy lies a small unregarded yellow sun. Orbiting this at a distance of roughly ninety-two million miles is an utterly insignificant little blue green planet whose ape-descended life forms are so amazingly primitive that they still think digital watches are a pretty neat idea. This planet has - or rather had - a problem, which was this: most of the people on it were unhappy for pretty much of the time. Many solutions were

suggested for this problem, but most of these were largely concerned with the movements of small green pieces of paper, which is odd because on the whole it wasn't the small green pieces of paper that were unhappy.

вариант 4.2:

Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense. Mr. Dursley was the director of a firm called Grunnings, which made drills. He was a big, beefy man with hardly any neck, although he did have a very large mustache. Mrs. Dursley was thin and blonde and had nearly twice the usual amount of neck, which came in very useful as she spent so much of her time craning over garden fences, spying on the neighbors. The Dursleys had a small son called Dudley and in their opinion there was no finer boy anywhere.

вариант 4.3:

Every inch of wall space is covered by a bookcase. Each bookcase has six shelves, going almost to the ceiling. Some bookshelves are stacked to the brim with hardback books: science, maths, history, and everything else. Other shelves have two layers of paperback science fiction, with the back layer of books propped up on old tissue boxes or lengths of wood, so that you can see the back layer of books above the books in front. And it still isn't enough. Books are overflowing onto the tables and the sofas and making little heaps under the windows. This is the living-room of the house occupied by the eminent Professor Michael Verres-Evans, and his wife, Mrs. Petunia Evans-Verres, and their adopted son, Harry James Potter-Evans-Verres.

вариант 4.4:

Let's set the existence-of-God issue aside for a later volume, and just stipulate that in some way, self-replicating organisms came into existence on this planet and immediately began trying to get rid of each other, either by spamming their environments with rough copies of themselves, or by more direct means which hardly need to be belabored. Most of them failed, and their genetic legacy was erased from the universe forever, but a few found some way to survive and to propagate. After about three billion years of this sometimes zany, frequently tedious fugue of carnality and carnage, Godfrey Waterhouse IV was born, in Murdo, South Dakota, to Blanche, the wife of a Congregational preacher named Bunyan Waterhouse. Like every other creature on the face of the earth, Godfrey was, by birthright, a stupendous badass, albeit in the somewhat narrow technical sense that he could trace his ancestry back up a long line of slightly less highly evolved stupendous badasses to that first

self-replicating gizmo-which, given the number and variety of its descendants, might justifiably be described as the most stupendous badass of all time. Everyone and everything that wasn't a stupendous badass was dead.

Задание 5

Пример решения на C++:

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <algorithm>

using namespace std;

const int maxN = 1e6 + 1000;

typedef long long ll;

ll a[maxN];

int main(void) {
    ios_base::sync_with_stdio(0);
    ll n;
    int m;
    cin >> n >> m;
    ll sum = 0;
    for (int i = 1; i <= m; i++) {
        cin >> a[i];
        sum += a[i];
    }
    if (sum <= n) {
        cout << m << endl;
        return 0;
    }
    ll p = sum - n;
    int ans = 1;
    int r = 1;
    ll cur = 0;
    for (int i = 2; i <= m; i++) {
        if (r < i) {
            cur = 0;
            r = i;
        } else {
            cur -= a[r];
        }
    }
}
```

```

        while (cur < p && r <= m) {
            cur += a[r];
            r++;
        }
        if (cur < p) {
            break;
        }
        ans = max(ans, m - (r - i));
        r--;
        cur -= a[i];
    }
    cout << ans << endl;
    return 0;
}

```

Задание 6

Пример решения:

```

]?( {} )? ) } ! { { ( [ ] ] { ( ~ ] { > ( [ ] } ! ) ) [ ( ~ } } ! ) [ ] ] { } ) } [ . @ . { { } "

```

Задание 7

Для анализа архива с текстами программ требовалось написать программу, которая являлась обоснованием получения результата.

Задание 8

При чтении последовательности запоминаем индексы элементов, где нарушается порядок. Так получаем те элементы, которые требуется изменить.

Входная последовательность считывается поэлементно. Текущее считанное число следует сравнивать с предыдущим и запоминать номера пар, в которых числа не по порядку. Таких пар может быть одна или две. Если найдена лишь одна пара, и последовательность дочитана до конца, то ответ состоит из номера пары и номера следующего за ним числа. Если пар две, то ответ состоит из номера первой пары и увеличенного на 1 номера второй пары. Если встречено две пары, то ввод можно не дочитывать.

```

program task8(input, output);
var  endFlag: boolean;
     aPrev, aCurr, i, n, num1, num2, a1st, a2nd: integer;
begin
    num1:= 0;
    num2:= 0;

```

```

endFlag:= false;
read(n, aCurr);
i:= 1;
repeat
    aPrev:= aCurr;
    read(aCurr);
    if (aPrev > aCurr) then begin {var 2: aPrev < aCurr}
        if (num1 = 0) then begin
            num1 := i;
            a1st:= aCurr;
            a2nd:= aPrev
        end
        else begin
            num2:= i+1;
            a1st:= aCurr;
            endFlag:= true;
        end;
    end;
    i:= i + 1;
until (i = n) or endFlag;
if endFlag then begin
    { writeln(num2, ' ', num1); var 2}
    writeln(a1st, ' ', a2nd) {var 1}
end
else begin
    { writeln(num1+1, ' ', num1); var 2}
    writeln(a1st, ' ', a2nd) {var 1}
end;
end.

```

Второй тур 10-11 классы

Задание 1

вариант 1.1

ayyz
ayzb
azaz
azbb
bayz
bazb
bbaz
bbbb

вариант 1.2

cyyz
cyzb
czaz
czbb
dayz
dazb
dbaz
dbbb

вариант 1.3

aaaaz
aaabb
yaaz
yabb
xyaz
xybb
xxyz
xxzb

вариант 1.4

gyyz
gyzb
gzaz
gzbb
hayz
hazb
hbaz
hbbb

Решение: Можно заметить, что некоторые числа могут быть представлены двумя или более разными полиномами. Так происходит потому, что $s[i] \cdot 24^{n-i} + s[i+1] \cdot 24^{n-i-1} = (s[i]-1) \cdot 24^{n-i} + (24+s[i+1]) \cdot 24^{n-i-1}$ а $s[i]$, $(s[i]-1)$, $s[i+1]$ и $(24+s[i+1])$ могут быть кодами букв при $1 \leq s[i] < 27$, $0 \leq s[i+1] < 3$. Если $i=1$, то следует рассматривать варианты из обеих частей тождества при $1 \leq s[1] < 27$, $0 \leq s[2] < 3$, а букву с 0-м номером считать "пустым местом" (см. ответы длиной 5 и длиной 4 в варианте 3).

Схема решения:

- 1) Если $X > 0$, то goto 2, иначе ответа нет.
- 2) Находим p – наибольшую степень 24, не превосходящую X .
- 3) Если $p > 1$, то goto 4, иначе ответ – строка из одной X -ой буквы.
- 4) Находим s – частное и r – остаток при делении нацело X на p .
- 5) Находим s_{next} – частное и r_{next} – остаток при делении r на $(p/24)$.
- 6) Если $1 \leq s < 27$ и $0 \leq s_{next} < 3$, то в ответ надо включить строки двух видов:

а) начинающиеся с s -ой буквы и продолжающиеся строками, полученными при расшифровке остатка r ;

б) начинающиеся с $(s-1)$ -ой буквы и $(s_{next}+24)$ -ой буквы за ней, и при $r_{next}>0$ продолжающиеся строками, полученными при расшифровке r_{next} , при $r_{next}=0$ и $p=24$ расшифровка состоит лишь из двух указанных букв.

7) Если $s=1$ и $0 \leq s_{next} < 3$ и мы определяем первую букву ответа для заданного числа (а не для его остатков), то также есть два вида строк: а) аналогичные указанному в б)а); и б) начинающиеся с $(s_{next}+24)$ -ой буквы, и продолжающиеся строками, полученными при расшифровке остатка r_{next} .

8) Во всех остальных случаях буква ответа определяется однозначно, это s -ая буква, и расшифровка продолжается для остатка r . По указанной схеме ищется множество слов, затем упорядочивается по алфавиту.

Задание 2

Кроме ответа требовалось привести то, как он был получен, например, программу.

вариант 2.1. $X = 9571766450$

Вывод (128 строк, для экономии места слова выписаны подряд через пробелы):

```
ayyyyyyz ayyyyyzb ayyyzaz ayyyzabb ayyyzayz ayyyzazb
ayyyzbaz ayyzbbb ayyzayyz ayyzayzb ayyzazaz ayyzazbb ayyzbayz
ayyzbazb ayyzbbaz ayyzbbbb ayzayyyz ayzayyzb ayzayzaz ayzayzbb
ayzazayz ayzazazb ayzazbaz ayzazbbb ayzbayyz ayzbayzb ayzbazaz
ayzbazbb ayzbbayz ayzbbazb ayzbbbaz ayzbbbbb azayyyyz azayyyzb
azayyzaz azayyzbb azayzayz azayzazb azayzbaz azayzbbb azazayyz
azazayzb azazazaz azazazbb azazbayz azazbazb azazbbaz azazbbbb
azbayyyz azbayyzb azbayzaz azbayzbb azbazayz azbazazb azbazbaz
azbazbbb azbbayyz azbbayzb azbbazaz azbbazbb azbbbayz azbbbazb
azbbbbbaz azbbbbbb bayyyyyz bayyyyzb bayyyzaz bayyyzbb bayyzayz
bayyzazb bayyzbaz bayyzbba bayzayyz bayzayzb bayzazaz bayzazbb
bayzbayz bayzbazb bayzbbaz bayzbbbb bazayyyz bazayyzb bazayzaz
bazayzbb bazazayz bazazazb bazazbaz bazazbbb bazbayyz bazbayzb
bazbazaz bazbazbb bazbbayz bazbbazb bazbbbaz bazbbbbbb bbayyyyz
bbayyyzb bbayyzaz bbayyzbb bbayzayz bbayzazb bbayzbaz bbayzbbb
bbazayyz bbazayzb bbazazaz bbazazbb bbazbayz bbazbazb bbazbbaz
bbazbbbb bbbayyyz bbbayyzb bbbayzaz bbbayzbb bbbazayz bbbazazb
bbbazbaz bbbazbbb bbbbayyz bbbbayzb bbbbazaz bbbbazbb bbbbbayz
bbbbbazb bbbbbbaz bbbbbbbb
```

вариант 2.2 $X = 18744709298$

Вывод (128 строк, для экономии места слова выписаны подряд через пробелы):

```
сyyyyyyz сyyyyyzb сyyyzaz сyyyzabb сyyyzayz сyyyzazb
сyyyzbaz сyyzbbb сyyzayyz сyyzayzb сyyzazaz сyyzazbb сyyzbayz
сyyzbazb сyyzbbaz сyyzbbbb сызayyyz сызayyzb сызayzaz сызayzbb
```

cyzazayz cyzazazb cyzazbaz cyzazbbb cyzbayyz cyzbayzb cyzbazaz
 cyzbazbb cyzbbayz cyzbbazb cyzbbbaz cyzbbbbb czayyyyz czayyyzb
 czayyzaz czayyzbb czayzayz czayzazb czayzbaz czayzbbb czazayyz
 czazayzb czazazaz czazazbb czazbayz czazbazb czazbbaz czazbbbb
 czbayyyyz czbayyzb czbayzaz czbayzbb czbazayz czbazazb czbazbaz
 czbazbbb czbbayyz czbbayzb czbbazaz czbbazbb czbbbayz czbbbazb
 czbbbbaz czbbbbbb dayyyyyz dayyyyzb dayyyzaz dayyyzbb dayyzayz
 dayyzazb dayyzbaz dayyzbbb dayzayyz dayzayzb dayzazaz dayzazbb
 dayzbayz dayzbazb dayzbbaz dayzbbbb dazayyyz dazayyzb dazayzaz
 dazayzbb dazazayz dazazazb dazazbaz dazazbbb dazbayyz dazbayzb
 dazbazaz dazbazbb dazbbayz dazbbazb dazbbbaz dazbbbbb dbayyyyz
 dbayyyzb dbayyzaz dbayyzbb dbayzayz dbayzazb dbayzbaz dbayzbbb
 dbazayyz dbazayzb dbazazaz dbazazbb dbazbayz dbazbazb dbazbbaz
 dbazbbbb dbbayyyz dbbayyzb dbbayzaz dbbayzbb dbbazayz dbbazazb
 dbbazbaz dbbazbbb dbbbayyz dbbbayzb dbbbazaz dbbbazbb dbbbbayz
 dbbbbazb dbbbbbaz dbbbbbbb

вариант 2.3 X = 27917652146

Вывод (128 строк, для экономии места слова выписаны подряд через пробелы):

euyuyuyz euyuyyzb euyuyzaz euyuyzbb euyyzayz euyyzazb
 euyyzbaz euyyzbbb euyyzayyz euyyzayzb euyyzazaz euyyzazbb euyzbayz
 euyzbazb euyzbbaz euyzbbbb eyzayyyz eyzayyzb eyzayzaz eyzayzbb
 eyzazayz eyzazazb eyzazbaz eyzazbbb eyzbayyz eyzbayzb eyzbazaz
 eyzbazbb eyzbbayz eyzbbazb eyzbbbaz eyzbbbbb ezayyyyz ezayyyzb
 ezayyzaz ezayyzbb ezayzayz ezayzazb ezayzbaz ezayzbbb ezazayyz
 ezazayzb ezazazaz ezazazbb ezazbayz ezazbazb ezazbbaz ezazbbbb
 ezbayyyyz ezbayyzb ezbayzaz ezbayzbb ezbazayz ezbazazb ezbazbaz
 ezbazbbb ezbbayyz ezbbayzb ezbbazaz ezbbazbb ezbbbayz ezbbbazb
 ezbbbbaz ezbbbbbb fayyyyyz fayyyyzb fayyyzaz fayyyzbb fayyzayz
 fayyzazb fayyzbaz fayyzbbb fayzayyz fayzayzb fayzazaz fayzazbb
 fayzbayz fayzbazb fayzbbaz fayzbbbb fazayyyz fazayyzb fazayzaz
 fazayzbb fazazayz fazazazb fazazbaz fazazbbb fazbayyz fazbayzb
 fazbazaz fazbazbb fazbbayz fazbbazb fazbbbaz fazbbbbb fbayyyyz
 fbayyyzb fbayyzaz fbayyzbb fbayzayz fbayzazb fbayzbaz fbayzbbb
 fbazayyz fbazayzb fbazazaz fbazazbb fbazbayz fbazbazb fbazbbaz
 fbazbbbb fbbayyyz fbbayyzb fbbayzaz fbbayzbb fbbazayz fbbazazb
 fbbazbaz fbbazbbb fbbbayyz fbbbayzb fbbbazaz fbbbazbb fbbbbayz
 fbbbbazb fbbbbaz fbbbbbb

вариант 2.4 X = 37090594994

Вывод (128 строк, для экономии места слова выписаны подряд через пробелы):

guyuyuyz guyuyyzb guyuyzaz guyuyzbb guyyzayz guyyzazb
 guyyzbaz guyyzbbb guyyzayyz guyyzayzb guyyzazaz guyyzazbb guyzbayz
 guyzbazb guyzbbaz guyzbbbb gyzayyyz gyzayyzb gyzayzaz gyzayzbb
 gyzazayz gyzazazb gyzazbaz gyzazbbb gyzbayyz gyzbayzb gyzbazaz
 gyzbazbb gyzbbayz gyzbbazb gyzbbbaz gyzbbbbb gzayyyyz gzayyyzb
 gzayyzaz gzayyzbb gzayzayz gzayzazb gzayzbaz gzayzbbb gzazayyz
 gzazayzb gzazazaz gzazazbb gzazbayz gzazbazb gzazbbaz gzazbbbb

gzbayyyz gzbayyyzb gzbayyzaz gzbayyzbb gzbazayz gzbazazb gzbazbaz
gzbazbbb gzbabayz gzbabayzb gzbazaz gzbazabb gzbabayz gzbabazb
gzbabbbaz gzbabbbb hayyyyyz hayyyyyzb hayyyzaz hayyyzbb hayyzayz
hayyzazb hayyzbaz hayyzbbb hayzayyz hayzayzb hayzazaz hayzazbb
hayzbayz hayzbazb hayzbbaz hayzbbbb hazayyyz hazayyzb hazayzaz
hazayzbb hazazayz hazazazb hazazbaz hazazbbb hazabayz hazbayzb
hazbazaz hazbazbb hazbbayz hazbbazb hazbbbaz hazbbbbb hbayyyyz
hbayyyzb hbayyzaz hbayyzbb hbayzayz hbayzazb hbayzbaz hbayzbbb
hbazayyz hbazayzb hbazazaz hbazazbb hbazbayz hbazbazb hbazbbaz
hbazbbbb hbbayyyz hbbayyzb hbbayzaz hbbayzbb hbbazayz hbbazazb
hbbazbaz hbbazbbb hbbbayyz hbbbayzb hbbbazaz hbbbazbb hbbbbayz
hbbbbazb hbbbbbaz hbbbbbbb

Программа, решающая задачу, может быть построена на основе схемы из решения задачи 1.

Задание 3

Решения проверялись на следующих тестах:

1 1
2 2
3 1+2
4 2*2
5 1+2*2
6 2+2*2
7 1+2+2*2
8 2*2*2
9 1+2*2*2
10 2+2*2*2
11 11
12 12
13 1+12
14 2+12
15 1+2+12
16 2+2+12
17 1+2+2+12
18 2+2+2+12
19 2*2*2+11
20 2*2*2+12
21 21
22 22
23 1+22
24 2*12
1000 112+2*2*222
1001 1+112+2*2*222
1002 2+112+2*2*222
1003 121+2*21*21

```

1004 122+2*21*21
1005 121+2*2*221
1006 122+2*2*221
1007 1+122+2*2*221
1008 2*2*12*21
1009 1+2*2*12*21
1010 2+2*2*12*21
1011 1+2+2*2*12*21
1012 2*22+2*22*22
1013 1+2*22+2*22*22
1014 11*12+2*21*21
100000 11112+2*2*22222
100001 1+11112+2*2*22222
100002 21*121+21*21*221
100003 11*211+2*221*221
100004 1+11*211+2*221*221
239 2*2*2+11*21
179 11+2*2*2*21
12345 1+2+11*1122
222222 222222

```

Для правильного решения требовалось найти оптимальное разложение числа на множители и числа на слагаемые.

```

1) строим множество с возможных чисел:
def fill_c(s):
    if (len(s) == 0 or int(s) < 1000000):
        if len(s) > 0:
            c.append(int(s))
            fill_c("1"+s)
            fill_c("2"+s)

2) строим ответ для каждого числа i в списке b[i]
def fill_b(n):
    if b[n] != "":
        return b[n]
    b[n] = "-1"
    if n in c:
        b[n] = str(n)
    for k in s_c:
        if n % k == 0 and k > 1 and n > k:
            b[n] = best_mul(b[n], fill_b(n // k), k)
    return b[n]

```

3) строим общий ответ для каждого числа, используя очередное слагаемое из b

```

def fill_a(n):

```

```

if a_f[n]:
    return a[n]
a[n] = b[n]
a_f[n] = True
if a[n] == "-1":
    a[n] = b_l[0][1] + "+" + fill_a(n - b_l[0][0])
for k in b_l:
    if k[0] >= n:
        break
    a[n] = best_sum(a[n], k[1], fill_a(n - k[0]))
return a[n]

```

Далее можно предподсчитать часть ответов и внести их в массив.

Задание 4

Задача "Лабиринт"

Во всех 4-х вариантах правильный ответ -1 (невозможно получить последовательность действий). Обоснованием может служить выделенная часть поля, где показывается что включить все лампочки невозможно. Общее решение следующее - для каждой лампочки вводим переменную включалась она или нет. Затем составляем систему линейных уравнений и получаем, что она не имеет решения.

Задание 5

Пример решения:

```

#include <cstdio>
#include <vector>
#include <cstdlib>
#include <cctype>

using namespace std;

const int frame_size = 8;

void encode(const vector<char> &in, vector<char> &out)
{
    for (int i = 0; i < int(in.size()); ++i) {
        out.push_back(in[i]);
        out.push_back(in[i]);
        out.push_back(in[i]);
        out.push_back(in[i]);
    }
    if (int(out.size()) % frame_size != 0) {

```

```

        out.push_back('0');
        out.push_back('0');
        out.push_back('0');
        out.push_back('0');
    }
}

void decode(const vector<char> &in, vector<char> &out)
{
    for (int pos = 0; pos < int(in.size()); pos += 4) {
        int sum = in[pos] + in[pos + 1] + in[pos + 2] + in[pos
+ 3] - 4 * '0';
        if (sum <= 1) sum = 0;
        else if (sum >= 3) sum = 1;
        out.push_back(char('0' + sum));
    }
}

int main()
{
    int cmd, size;
    vector<char> data;

    scanf("%d", &cmd);
    scanf("%d", &size);

    int c;
    while ((c = getchar()) != EOF) {
        if (isspace(c)) {
        } else if (c == '0' || c == '1') {
            data.push_back(c);
        }
    }

    vector<char> result;
    if (cmd == 0) {
        encode(data, result);
    } else if (cmd == 1) {
        decode(data, result);
    }

    for (int i = 0; i < int(result.size()); ++i) {
        if (i > 0 && i % 64 == 0) {
            putchar('\n');
        }
    }
}

```

```

        putchar(result[i]);
    }
    putchar('\n');
    return 0;
}

```

Задание 6

Пример решения

```

?}}?[]{(~[]]{<([~]!){} [({[]){([]){({}}[])?}{(.@.{})"

```

Задание 7

Один из вариантов решения – найти число одинаковых символов между каждым эталоном и запрашиваемой картинкой. Далее выбрать эталон с максимальным числом совпадающих символов.

Задание 8

Задача аналогична задаче 8 из первого тура – находим тройки, которые не являются возрастающими или убывающими. Требуется поменять местами элементы в этих тройках, если одна из них не содержит минимальный элемент.

Задача А. Упорядоченные кубики

У Пети есть набор кубиков. Строго говоря, не каждый кубик является кубом, но является прямоугольным параллелепипедом. Петья полагает, что кубики можно сравнивать между собой по их измерениям (длине, ширине и высоте). Так, кубик A считается меньшим чем кубик B , если можно их так расположить относительно друг друга, что их соответствующие ребра будут параллельны и при этом ширина A будет меньше ширины B , длина A — меньше длины B и высота A — меньше высоты B .

Рассмотрим пример. Измерения (длина x ширина x высота) кубика A : $3 \times 3 \times 2$. Измерения кубика B : $4 \times 3 \times 4$. Можно повернуть кубик A так, чтобы его измерения стали $3 \times 2 \times 3$ и убедиться, что он меньше кубика B . Если добавить кубик C с измерениями $5 \times 2 \times 1$, можно видеть, что он не больше и не меньше кубика A и кубика B .

Петья разложил свой набор кубиков в последовательность $C_1, C_2, \dots, C_N$. Найдите в этой последовательности самую длинную цепочку кубиков, упорядоченных по возрастанию, то есть набор $C_K, C_{K+1}, \dots, C_{K+M}$, где кубик C_K меньше C_{K+1} , C_{K+1} меньше C_{K+2} , и т. д., а K и M таковы, что M наибольшее из возможных. Цепочка может состоять из одного кубика. Если искомым цепочек несколько, следует указать ту, что находится ближе к концу последовательности.

Формат входных данных

На вход программа принимает натуральное число N ($1 \leq N \leq 100\,000$) — количество кубиков. В последующих N строках ввода программе заданы измерения кубиков — тройки натуральных чисел a_i, b_i, c_i , $0 < a_i, b_i, c_i \leq 10^4$.

Формат выходных данных

В первой строке необходимо вывести длину искомой цепочки кубиков. В последующих строках измерения каждого из кубиков цепочки. Измерения следует выводить в том виде, в котором они были даны во вводе.

Примеры

тест	ответ
5	3
2 1 1	3 1 3
3 2 2	4 4 2
3 1 3	5 3 5
4 4 2	
5 3 5	

Задача В. Бармаглот

Компания «Batmalay's Computing» утча пожелания пользователей и выпустила вторую версию компьютера — Batmalot-2. Поскольку у сотрудников компании несколько странные представления о технологиях, компьютер остался довольно непривычным.

Во-первых, исходные данные он читает с ленты, где могут быть записаны целые числа и латинские буквы. Чтобы показать, что ввод данных закончен, ленту нужно просто оторвать.

Во-вторых, результаты вычислений компьютер печатает на точно такой же ленте. Центральный процессор компьютера оснащён одним регистром; вместо оперативной памяти компьютер оснащён очередью и стеком неограниченного размера. Регистр, а также каждая ячейка очереди и стека могут содержать либо число, либо латинскую букву или пробел, либо специальное значение [BARMALU], которое используется для обозначения логического лжи, при том что любое другое обозначает истину.

Программа для Batmalot'a представляет собой строку символов, каждый из которых задаёт машинную команду; программа, состоящая из символов-команд, выполняется последовательно слева направо, кроме двух команд, которые могут нарушить эту последовательность.

- Команды 'a', 'b', 'c' и все остальные латинские буквы означают "занести данную букву в регистр".
- Команды '0', '1', ..., '9' означают "занести в регистр соответствующее число".
- Команда '@' означает занести в регистр пробел.
- Команды '+', '-', '*', '/', '%', означают соответствующие арифметические действия, операции целочисленные. % - операция взятия остатка
- Команды '<', '>', '=' означают сравнение двух чисел или двух символов, '&' и '!' означают логическое "и" и логическое "или", все эти команды используют значение из регистра в качестве левого операнда, значение с вершины стека в качестве правого (оно при этом из стека извлекается), результат заносится обратно в регистр.
- Команда '#' умножает содержимое регистра в десять раз,
- Команда ' _ ' делит содержимое регистра в десять раз.

- Команда '!' работает как логическое отрицание содержимого регистра: если там значение [BARMALU], то заносится значение 1, если любое другое - заносится значение [BARMALU].
- Команда ':' выдаёт текущее значение регистра на печать
- команда '?' вводит очередное число(цифром, а не по разрядам), а если на входе кончилась (оборвалась) лента, заносит в регистра значение [BARMALU].
- Команда ']' заносит значение из регистра в стек;
- Команда '[' извлекает значение с вершины стека и заносит его в регистра (если извлекать нечего, в регистр заносится [BARMALU]);
- Команда '~' меняет местами значения в регистра и на вершине стека.
- Команда '}' заносит значение из регистра в очередь,
- Команда '{' извлекает из очереди самое старое значение и помещает в аккумулятор (или помещает туда [BARMALU], если очередь пуста).
- Команды '(' и ')' предназначены для организации ветвлений и циклов и всегда должны в программе стоять парами, то есть в программе должен обязательно соблюдаться баланс круглых скобок. Выполняются они так. Команда (, если в регистре [BARMALU], идёт по программе вперёд, пока не найдёт паруную скобку, и после этого выполнение продолжится со следующей за этой закрывающей скобкой позиции; если в регистре было что-то другое, команда вообще ничего не делает, то есть выполнение продолжается прямо с командой, следующей за ней. Команда), наоборот, если в регистре [BARMALU], не делает ничего, тогда как если там что-то другое, просматривает программу назад до парной круглой скобки, после чего продолжает выполнение с командой, стоящей после такой скобки справа.

Наконец, команда '~' прекращает выполнение программы, при этом выполнение считается успешным. Если программа кончилась, не встретив эту команду, она завершается аварийно.

В нашем распоряжении оказался эмулятор данного компьютера (<http://ejudge.cs.msu.ru/battlejot2/>) и требуется написать для него программу, решающую следующую задачу:

В одной компьютерной игре игрок выставляет в линию шариков разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарики при этом сдвигаются друг к другу, и ситуация может повториться.

Напишите программу, которая по данной ситуации определит, сколько шариков будет сейчас "уничтожено". Естественно, непрерывный цепочек из трех и более одноцветных шаров в начальный момент может быть не более одной.

Формат входных данных

Сначала вводится количество шариков в цепочке (не более 1000) и цвета шариков (от 0 до 9, каждому цвету соответствует свое целое число).

Формат выходных данных

Требуется вывести количество шариков, которое будет "уничтожено".

Примеры

тест	ответ
5 1 3 3 3 2	3

Задача С. Корпорация добра

В одной очень известной фирме существует четкая иерархическая структура. У каждого из N работников есть свой непосредственный начальник. Директором фирмы является сотрудник под номером 0, у него нет начальника.

Руководство фирмы приняло решение об увольнении некоторых служащих и пометило их в специальном списке. К счастью, не все еще решено окончательно. Если разозлить любого сотрудника, то он вычеркивает из списка на увольнение всех своих подчиненных, которые были в этом списке и подписывает в него всех своих подчиненных, которых там не было. (Когда сотрудник элиты, он становится настолько злым, что считает своими подчиненными себя, своих непосредственных подчиненных, непосредственных подчиненных своих непосредственных подчиненных и так далее)

Борис, как невероятно добрый человек решил спасти ситуацию. Он решил разозлить некоторых сотрудников так, чтобы никого в итоге не пришлось увольнять. Помогите Борису узнать минимальное количество сотрудников, которое ему придется элиты.

Формат входных данных

В первой строке вводится число N ($1 \leq N \leq 10000$) — количество сотрудников в очень известной фирме. В следующих $N - 1$ строках в i -й строке вводится непосредственный начальник i -го сотрудника.

Далее вводится число M ($0 \leq M \leq 10000$) — количество сотрудников в начальном списке на увольнение. Далее вводится M различных целых чисел A_i — список номеров сотрудников в начальном списке на увольнение.

Формат выходных данных

Выведите одно число - минимальное количество сотрудников, которое придется злить Борису.

Примеры

тест	ответ
3	1
0	
1	
2	
1 2	

Задача Д. Нарисованный граф

Петя очень любит рисовать графы в тетрадке в клетку. В углах клеток могут быть нарисованы вершины графа, некоторые соседние вершины могут быть соединены ребрами, при этом ребра могут быть только длины 1 (сторона одной клетки). Нарисовав граф, Петя нумерует вершины и выписывает список ребер.

Однажды Петя потерял листочек с нарисованным графом, но у него остался список ребер. Помогите ему восстановить рисунок.

Формат входных данных

В этой задаче у вас есть несколько входных файлов в тестирующей системе, для каждого из которых требуется отправить на проверку файл с графом. В первой строке исходного файла находится число вершин N и число ребер M . В следующих строках пары чисел a, b , означающие наличие ребра между вершиной с номером a и b .

Гарантируется что граф во входном файле всегда задает какой-то из нарисованных графов.

Формат выходных данных

В файле с ответом граф имеет следующий формат.

- Вершина обозначается символом 'o'
- Ребра обозначаются символами '-', ' ' или '|'
- Ребро имеет всегда длину 1 и состоит из одного символа
- Пустые места обозначаются символом ' '.

Примеры

тест	ответ
4 4	0-0
1 2	.
2 3	0-0
3 4	
4 1	
тест	ответ
6 6	0-0-0-0-0-0
1 2	
2 3	..0-0-0-0
3 4	
4 1	
1 5	
3 6	

Задача Е. Распознавание языка

В тестирующей системе вам дан архив с исходными текстами программ и файлом (files.txt), который содержит перечисление исходных кодов с указанием языка программирования (c++, Pascal, python и т.д.)

Вам требуется написать программу, которой на стандартный поток передается список из какого-то подмножества файлов в архиве. Для каждого файла из списка требуется определить язык программирования.

Например, если в исходном архиве две программы в файлах file1

```
int quest(int x) {
    while (x < 1024) {
        for(int y = 0; y < x; y = y + 1) {
            if (y * y == x) {
                return y;
            }
        }
        x = x + 1;
    }
    return -1;
}
```

и file2

```
def f():
    for i in range(10):
        print(i)
```

то в файле files.txt будут следующие строки:

```
2
file1 c++
file2 python
```

Вашей программе при проверке будет передан на стандартный поток ввода список файлов, имена которых могут отличаться, но содержимое каждого совпадает с содержимым какого-то из файлов в архиве. Ваша задача по заданному файлу определить язык программирования, который соответствует языку данного файла из архива. Например, если содержимое somefile совпадает с содержимым file2, то это язык python.

Формат входных данных

На стандартном потоке задается число файлов N ($1 \leq N \leq 100$), для которых мы хотим определить язык, затем в N строках имена файлов. Гарантируется что, вашей программе доступны эти файлы для открытия на чтение.

Формат выходных данных

Для каждого файла выведите язык программирования.

Примеры

тест	ответ
1 somefile	python

Задача F. Оптимизация программы

Как известно, программы на таких языках, как Си++, Паскаль и др., как правило, необходимо перед запуском скомпилировать. Результатом компиляции является исполняемый файл, содержащий команды процессора, которые будут исполняться при запуске программы. Исполняемый файл является специфичным для операционной системы и архитектуры процессора. Например, исполняемый файл для Windows и архитектуры x64 не может быть напрямую запущен на операционной системе iOS и процессоре ARM.

Когда необходимо работать с командами процессора обычно используется бинарное представление, хранимое в исполняемом файле, а язык ассемблера. Язык ассемблера позволяет записывать команды процессора в человеко-читаемой форме. Мы рассмотрим некоторое упрощенный язык ассемблера, достаточное для решения задачи.

Процессор исполняет команды последовательно начиная от первой команды в файле и до конца файла. Каждая команда в файле занимает одну строку текста и имеет следующий вид:

```
[LABELS] opcode [arg]
```

LABELS — это обязательная последовательность *меток*, каждая метка - это цепочка заглавных и строчных латинских букв, цифр и знаков `.` (точка), `-`, `$`. Метка завершается знаком «двоеточие». Перед метками, между метками и знаком двоеточия, после двоеточия могут располагаться пробельные символы (то есть пробел или табуляция). Примеры:

```
L1: L2:
.L4$:
L3:
```

OPCODE — это операция процессора. Определены следующие операции:

- **C** — операция сравнения. Операция сравнения устанавливает *флаги результата* в зависимости от результата операции над своими аргументами.
- **A** — операция вычисления. Операция не модифицирует флаги результата, которые, возможно, остались после предыдущей операции сравнения.
- **Z** — операция вычисления, в отличие от **A** эта операция модифицирует флаги результата unpredictable образом.
- **E** — последняя операция в программе. Эта операция всегда является текстуально последней. Она всегда единственная в файле.
- **B** — операция передачи управления.

Операции **C**, **A**, **Z** имеют единственный аргумент в виде **#ID**, где **ID** — это положительное целое 32-битное число. По сути — это уникальный идентификатор каждой команды в программе. Ни один **ID** не используется в программе дважды. Операция **E** не имеет аргументов. Операция **B** имеет один аргумент — метку.

Рассмотрим программу:

A	#1
A	#2
B	L1
L2:	C
	#3
	#4
	L3
L1:	Z
	#5
	L2
L3:	A
	#6
E	

Операция B L1 передает управление на метку L1, то есть после этой инструкции следующей будет выполняться инструкция B L2. С учетом таких передач управления рассмотренный фрагмент эквивалентен следующему:

A	#1
A	#2
Z	#5
C	#3
Z	#4
A	#6
E	

В исходной программе было 10 команд процессора, а в преобразованной стало 7 команд.

Семантикой программы назовем последовательность идентификаторов команд в том порядке, в котором они выполнялись процессором. В семантику программы не входят аргументы инструкции B, хотя они учитываются при построении семантики.

Задача оптимизации программы заключается в нахождении программы минимальной длины, семантика которой совпадает с семантикой исходной программы. Операции, рассмотренные выше, не позволяют реализовать условные операторы или циклы. Условное выполнение реализуется с помощью суффиксов условного выполнения. Например, команда B всегда выполняет переход в другую точку программы (говорят «выполняет безусловный переход»), а команда BEQ выполняет переход только если предыдущая инструкция сравнения C установила флаг равенства аргументов.

Команда сравнения C устанавливает флаги «равно» и «больше», что позволяет проверить аргументы на равенство, неравенство, больше, меньше, больше или равно, меньше или равно. Соответствующие суффиксы условного выполнения записываются как EQ (equal), NE (not equal), GT (greater than), LT (less than), GE (greater

or equal), LE (less or equal). Суффиксы условного выполнения могут записываться после любой инструкции, кроме E. Однако команда Z устанавливает флаги некоторым непредсказуемым образом, и поэтому после команды Z использовать условное выполнение нельзя.

Например,

C	#1
A	#2
BEQ	L1
BGT	L2

; переход, если сравнение #1 установило флаг равенства
; переход, если сравнение #1 установило флаг <<больше>>

Условие выполняться может любая инструкция. Например, фрагмент программы:

C	#1
BLT	L1
A	#2
B	L2
L1:	A
	#3
L2:	

этот фрагмент программы использует условные переходы, но он может быть оптимизирован следующим образом:

C	#1
AGE	#2
ALT	#3

Вам нужно скачать архив программ. Архив имеет формат .zip, и файлы с программами называются in01.asp, in02.asp, ..., in10.asp. Вам необходимо оптимизировать выданные вам программы, то есть найти семантически-эквивалентные программы минимальной длины. В преобразованной команде должны быть сохранены все команды 'C', 'A', 'Z', которые присутствовали в исходной программе. Их порядок может быть текстуально другим, но при условии, что при выполнении программы при любом возможном исходе выполнения условной операции порядок выполнения операций 'C', 'A', 'Z' сохраняется неизменным.

В качестве ответа вам нужно будет сдать 10 оптимизированных программ.

Оптимизированная программа не засчитывается, если в нем нарушен синтаксис программы, или получившаяся программа не будет эквивалентна исходной программе.

Задача А. Упорядоченные кубики

У Пети есть набор кубиков. Строго говоря, не каждый кубик является кубом, но является прямоугольным параллелепипедом. Пета полагает, что кубики можно сравнивать между собой по их измерениям (длине, ширине и высоте). Так, кубик A считается меньшим чем кубик B , если можно их так расположить относительно друг друга, что их соответствующие ребра будут параллельны и при этом ширина A будет меньше ширины B , длина A — меньше длины B и высота A — меньше высоты B .

Рассмотрим пример. Измерения (длина x ширина x высота) кубика A : $3 \times 3 \times 2$. Измерения кубика B : $4 \times 3 \times 4$. Можно повернуть кубик A так, чтобы его измерения стали $3 \times 2 \times 3$ и убедиться, что он меньше кубика B . Если добавить кубик C с измерениями $5 \times 2 \times 1$, можно видеть, что он не больше и не меньше кубика A и кубика B .

Пета разложил свой набор кубиков в последовательность $C_1, C_2, \dots, C_N$. Найдите в этой последовательности самую длинную цепочку кубиков, упорядоченных по убыванию, то есть набор $C_K, C_{K+1}, \dots, C_{K+M}$, где кубик C_K больше C_{K+1} , C_{K+1} больше C_{K+2} , и т. д., а K и M таковы, что M наибольшее из возможных. Цепочка может состоять из одного кубика. Если искомым цепочек несколько, следует указать ту, что находится ближе к концу последовательности.

Формат входных данных

На вход программа принимает натуральное число N ($1 \leq N \leq 100\,000$) — количество кубиков. В последующих N строках ввода программе заданы измерения кубиков — тройки натуральных чисел a_i, b_i, c_i , $0 < a_i, b_i, c_i \leq 10^4$.

Формат выходных данных

В первой строке необходимо вывести длину искомой цепочки кубиков. В последующих строках измерения каждого из кубиков цепочки. Измерения следует выводить в том виде, в котором они были даны во вводе.

Примеры

тест	ответ
5	3
2 1 1	5 3 5
3 2 2	4 4 2
5 3 5	3 1 3
4 4 2	
3 1 3	

Задача В. Бармаглот

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Компания «Barmaley's Computing» утѣла пожелания пользователям и выпустила вторую версию компьютера — Barmaqlot-2. Поскольку у сотрудников компании несколько странные представления о технологиях, компьютер остался довольно непризывным.

Во-первых, исходные данные он читает с ленты, где могут быть записаны целые числа и латинские буквы. Чтобы показать, что ввод данных закончен, ленту нужно просто оторвать.

Во-вторых, результаты вычислений компьютер печатает на точно такой же ленте. Центральный процессор компьютера оснащён одним регистром; вместо оперативной памяти компьютер оснащён очередью и стеком неограниченного размера. Регистр, а также каждая ячейка очереди и стека могут содержать либо число, либо латинскую букву или пробел, либо специальное значение [BARMALEY], которое используется для обозначения логической лжи, при том что любое другое обозначает истину.

Программа для Barmaqlot'a представляет собой строку символов, каждый из которых задаѣт машинную команду; программа, состоящая из символов-команд, выполняется последовательно слева направо, кроме двух команд, которые могут нарушить эту последовательность.

- Команды 'a', 'b', 'c' и все остальные латинские буквы означают "занести данную букву в регистр".
- Команды '0', '1', ..., '9' означают "занести в регистр соответствующее число".
- Команда '@' означает занести в регистр пробел.
- Команды '+', '-', '*', '/', '%' означают соответствующие арифметические действия, операции целочисленные. % - операция взятия остатка
- Команды '<', '>', '=' означают сравнение двух чисел или двух символов, '&' и '!' означают логическое "и" и логическое "или", все эти команды используют значение из регистра в качестве левого операнда, значение с вершины стека в качестве правого (оно при этом из стека извлекается), результат заносится обратно в регистр.
- Команда '#' умножает содержимое регистра в десять раз,
- Команда ' _ ' делит содержимое регистра в десять раз.

- Команда '!' работает как логическое отрицание содержимого регистра: если там значение [BARMALU], то заносится значение 1, если любое другое - заносится значение [BARMALU].
- Команда ':' выдаёт текущее значение регистра на печать
- команда '?' вводит очередное число(цифром, а не по разрядам), а если на входе кончилась (оборвалась) лента, заносит в регистра значение [BARMALU].
- Команда ']' заносит значение из регистра в стек;
- Команда '[' извлекает значение с вершины стека и заносит его в регистра (если извлекать нечего, в регистр заносится [BARMALU]);
- Команда '~' меняет местами значения в регистра и на вершине стека.
- Команда '}' заносит значение из регистра в очередь,
- Команда '{' извлекает из очереди самое старое значение и помещает в аккумулятор (или помещает туда [BARMALU], если очередь пуста).
- Команды '(' и ')' предназначены для организации ветвлений и циклов и всегда должны в программе стоять парами, то есть в программе должен обязательно соблюдаться баланс круглых скобок. Выполняются они так. Команда (, если в регистре [BARMALU], идёт по программе вперёд, пока не найдёт паруную скобку, и после этого выполнение продолжится со следующей за этой закрывающей скобкой позиции; если в регистре было что-то другое, команда вообще ничего не делает, то есть выполнение продолжается прямо с командой, следующей за ней. Команда), наоборот, если в регистре [BARMALU], не делает ничего, тогда как если там что-то другое, просматривает программу назад до парной круглой скобки, после чего продолжает выполнение с командой, стоящей после такой скобки справа.

Наконец, команда '~' прекращает выполнение программы, при этом выполнение считается успешным. Если программа кончилась, не встретив эту команду, она завершается аварийно.

В нашем распоряжении оказался эмулятор данного компьютера (<http://ejudge.cs.msu.ru/barmalot2/>) и требуется написать для него программу, решающую следующую задачу:

В одной компьютерной игре игрок выставляет в линию шариков разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарики при этом сдвигаются друг к другу, и ситуация может повториться.

Напишите программу, которая по данной ситуации определяет, сколько шариков останется. Естественно, непрерывный цепочек из трех и более одноцветных шаров в начальный момент может быть не более одной.

Формат входных данных

Сначала вводится количество шариков в цепочке (не более 1000) и цвета шариков (от 0 до 9, каждому цвету соответствует свое целое число).

Формат выходных данных

Требуется вывести количество шариков, которые останутся.

Примеры

тест	ответ
5 1 3 3 3 2	2

Задача С. Корпорация добра

Ограничение по времени:

2 секунды

Ограничение по памяти:

256 мегабайт

В одной очень известной фирме существует четкая иерархическая структура. У каждого из N работников есть свой непосредственный начальник. Директором фирмы является сотрудник под номером 0, у него нет начальника.

Руководство фирмы приняло решение об увольнении некоторых служащих и пометило их в специальном списке. К счастью, не все еще решено окончательно. Если разозлить любого сотрудника, то он вычеркивает из списка на увольнение всех своих подчиненных, которые были в этом списке и дописывает в него всех своих подчиненных, которых там не было. (Когда сотрудник элиты, он становится настолько элитой, что считает своими подчиненными себя, своих непосредственных подчиненных, непосредственных подчиненных своих непосредственных подчиненных и так далее)

Борис — сотрудник компании, соперничающей с очень известной компанией. Он решил разозлить некоторых сотрудников так, чтобы в итоге все сотрудники были уволены. Помогите Борису узнать минимальное количество сотрудников, которое ему придется элитить.

Формат входных данных

В первой строке вводится число N ($1 \leq N \leq 10000$) — количество сотрудников в очень известной фирме. В следующих $N - 1$ строках в i -й строке вводится непосредственный начальник i -го сотрудника.

Далее вводится число M ($0 \leq M \leq 10000$) — количество сотрудников в изначальном списке на увольнение. Далее вводится M различных целых чисел A_i — список номеров сотрудников в изначальном списке на увольнение.

Формат выходных данных

Выведите одно число - минимальное количество сотрудников, которое придется элить Борису.

Примеры

тест	ответ
3	2
0	
1	
2	
1 2	

Задача D. Нарисованный граф

Петя очень любит рисовать графы в тетрадке в клетку. В углах клеток могут быть нарисованы вершины графа, некоторые соседние вершины могут быть соединены ребрами, при этом ребра могут быть только длины 1 (сторона одной клетки). Нарисовав граф, Петя нумерует вершины и выписывает список ребер.

Однажды Петя потерял листочек с нарисованным графом, но у него остался список ребер. Помогите ему восстановить рисунок.

Формат входных данных

В этой задаче у вас есть несколько входных файла в тестирующей системе, для каждого из которых требуется отправить на проверку файл с графом. В первой строке исходного файла находится число вершин N и число ребер M . В следующих строках пары числе a, b , означающие наличие ребра между вершиной с номером a и b .

Гарантируется что граф во входном файле всегда задает какой-то из нарисованных графов.

Формат выходных данных

В файле с ответом граф имеет следующий формат.

- Вершина обозначается символом 'o'
- Ребра обозначаются символами ',-' или ',|'

- Ребро имеет всегда длину 1 и состоит из одного символа.
- Пустые места обозначаются символом ', '

Примеры

тест	ответ
4 4	o-o
1 2	.
2 3	o-o
3 4	
4 1	
тест	ответ
6 6	o-o-o-.-
1 2	
2 3	..o-o-o
3 4	
4 1	
1 5	
3 6	

Задача E. Распознавание языка

В тестирующей системе вам дан архив с исходными текстами программ и файлом (files.txt), который содержит перечисление исходных кодов с указанием языка программирования (c++, Pascal, python и т.д.)

Вам требуется написать программу, которой на стандартный поток передается список из какого-то подмножества файлов в архиве. Для каждого файла из списка требуется определить язык программирования.

Например, если в исходном архиве две программы в файлах file1

```
int quest(int x) {
    while (x < 1024) {
        for(int y = 0; y < x; y = y + 1) {
            if (y * y == x) {
                return y;
            }
        }
        x = x + 1;
    }
}
```

```
    return -1;
}

и file2

def f():
    for i in range(10):
        print(i)

то в файле files.txt будут следующие строки:

2
file1 c++
file2 python
```

Вашей программе при проверке будет передан на стандартный поток ввода список файлов, имена которых могут оглядаться, но содержимое каждого совпадает с содержимым какого-то из файлов в архиве. Ваша задача по заданному файлу определить язык программирования, который соответствует языку данного файла из архива. Например, если содержимое somefile совпадает с содержимым file2, то это язык python.

Формат входных данных

На стандартном потоке задается число файлов N ($1 \leq N \leq 100$), для которых мы хотим определить язык, затем в N строках имена файлов. Гарантируется что, вашей программе доступны эти файлы для открытия на чтение.

Формат выходных данных

Для каждого файла выведите язык программирования.

Примеры

	тест	ответ
1		python
somefile		

Задача F. Оптимизация программы

Как известно, программы на таких языках, как Си++, Паскаль и др., как правило, необходимо перед запуском скомпилировать. Результатом компиляции является исполняемый файл, содержащий команды процессора, которые будут исполняться при запуске программы. Исполняемый файл является специфичным для операционной системы и архитектуры процессора. Например, исполняемый файл

для Windows и архитектуры x64 не может быть напрямую запущен на операционной системе iOS и процессоре ARM.

Когда необходимо работать с командами процессора обычно используется бинарное представление, хранящееся в исполняемом файле, а язык ассемблера. Язык ассемблера позволяет записывать команды процессора в человеко-читаемой форме. Мы рассмотрим некоторое упрощение языка ассемблера, достаточное для решения задачи.

Процессор исполняет команды последовательно начиная от первой команды в файле и до конца файла. Каждая команда в файле занимает одну строку текста и имеет следующий вид:

[LABELS] OP CODE [ARG]

LABELS — это обязательная последовательность *меток*, каждая метка - это цепочка заглавных и строчных латинских букв, цифр и знаков . (точка), -, \$. Метка завершается знаком «двоеточие». Перед метками, между метками и знаком двоеточия, после двоеточия могут располагаться пробельные символы (то есть пробел или табуляция). Примеры:

L1: L2:
L1\$:
L3 :

OP CODE — это операция процессора. Определены следующие операции:

- C — операция сравнения. Операция сравнения устанавливает *флаги результата* в зависимости от результата операции над своими аргументами.
- A — операция вычисления. Операция не модифицирует флаги результата, которые, возможно, остались после предыдущей операции сравнения.
- Z — операция вычисления, в отличие от A эта операция модифицирует флаги результата независимо от результата операции.
- E — последняя операция в программе. Эта операция всегда является текстовой последней. Она всегда единственная в файле.
- B — операция передачи управления.

Операции C, A, Z имеют единственный аргумент в виде #ID, где ID — это положительное целое 32-битное число. По сути — это уникальный идентификатор каждой команды в программе. Ни один ID не используется в программе дважды. Операция E не имеет аргументов. Операция B имеет один аргумент — метку. Рассмотрим программу:

A	#1
A	#2
B	L1
L2:	C #3
	Z #4
	B L3
L1:	Z #5
	B L2
L3:	A #6
	E

Операция B L1 передает управление на метку L1, то есть после этой инструкции следующей будет выполняться инструкция B L2. С учетом таких передач управления рассмотренный фрагмент эквивалентен следующему:

A	#1
A	#2
Z	#5
C	#3
Z	#4
A	#6
E	

В исходной программе было 10 команд процессора, а в преобразованной стало 7 команд.

Семантикой программы назовем последовательность идентификаторов команд в том порядке, в котором они выполнялись процессором. В семантику программы не входят аргументы инструкции B, хотя они учитываются при построении семантики.

Задача оптимизации программы заключается в нахождении программы минимальной длины, семантика которой совпадает с семантикой исходной программы. Операции, рассмотренные выше, не позволяют реализовать условные операторы или циклы. Условное выполнение реализуется с помощью суффиксов условного выполнения. Например, команда B всегда выполняет переход в другую точку программы (говорят «выполняет безусловный переход»), а команда BEQ выполняет переход только если предыдущая инструкция сравнения C установила флаг равенства аргументов.

Команда сравнения C устанавливает флаги «равно» и «больше», что позволяет проверить аргументы на равенство, неравенство, больше, меньше, больше или равно, меньше или равно. Соответствующие суффиксы условного выполнения записываются как EQ (equal), NE (not equal), GT (greater than), LT (less than), GE (greater

or equal), LE (less or equal). Суффиксы условного выполнения могут записываться после любой инструкции, кроме E. Однако команда Z устанавливает флаги некоторым непредсказуемым образом, и поэтому после команды Z использовать условное выполнение нельзя.

Например,

C	#1
A	#2
BEQ	L1
BGT	L2

; переход, если сравнение #1 установило флаг равенства
; переход, если сравнение #1 установило флаг <<больше>>

Условие выполняться может любая инструкция. Например, фрагмент программы:

C	#1
BLT	L1
A	#2
B	L2
L1:	A #3
L2:	

этот фрагмент программы использует условные переходы, но он может быть оптимизирован следующим образом:

C	#1
AGE	#2
ALT	#3

Вам нужно скачать архив программ. Архив имеет формат .zip, и файлы с программами называются in01.asp, in02.asp, ..., in10.asp. Вам необходимо оптимизировать выданные вам программы, то есть найти семантически-эквивалентные программы минимальной длины. В преобразованной команде должны быть сохранены все команды 'C', 'A', 'Z', которые присутствовали в исходной программе. Их порядок может быть текстуально другим, но при условии, что при выполнении программы при любом возможном исходе выполнения условной операции порядок выполнения операций 'C', 'A', 'Z' сохраняется неизменным.

В качестве ответа вам нужно будет сдать 10 оптимизированных программ.

Оптимизированная программа не засчитывается, если в нем нарушен синтаксис программы, или получившаяся программа не будет эквивалентна исходной программе.

Задача А. Упорядоченные плитки

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У Пети есть коллекция шоколадных плиток. Каждая плитка имеет форму прямоугольника. Пети полагает, что плитки можно сравнивать между собой по их измерениям (ширине и длине). Так, плитка A считается меньшей чем плитка B , если можно их так расположить относительно друг друга, что их соответствующие стороны будут параллельны и при этом ширина A будет меньше ширины B и длина A — меньше длины B .

Рассмотрим пример. Измерения (длина $\times$ ширина) плитки A : 2×3 . Измерения плитки B : 4×3 . Можно повернуть плитку A так, чтобы его измерения стали 3×2 и убедиться, что она меньше плитки B . Если добавить плитку C с измерениями 5×1 , можно видеть, что она не больше и не меньше плиток A и B .

Пети разложил свою коллекцию плиток в последовательность $C_1, C_2, \dots, C_N$. Найдите в этой последовательности самую длинную цепочку плиток, упорядоченных по возрастанию, то есть набор $C_K, C_{K+1}, \dots, C_{K+M}$, где плитка C_K меньше C_{K+1} , C_{K+1} меньше C_{K+2} , и т. д., а K и M таковы, что M наибольшее из возможных. Цепочка может состоять из одной плитки. Если искомым цепочек несколько, следует указать ту, что находится ближе к концу последовательности.

Формат входных данных

На вход программа принимает натуральное число N ($1 \leq N \leq 100\,000$) — количество плиток. В последующих N строках ввода программе заданы измерения плиток — двойки натуральных чисел a_i, b_i , $0 < a_i, b_i \leq 10\,000$.

Формат выходных данных

В первой строке необходимо вывести длину искомой цепочки плиток. В последующих строках измерения каждой из плиток цепочки. Измерения следует выводить в том виде, в котором они были даны во вводе.

Примеры

тест	ответ
5	3
1 1	1 1
3 2	3 2
3 4	3 4
2 4	
1 3	

Задача В. Шахматы

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В варианте шахмат, предложенном Хосе Раулем Капабланкой, используется фигура канцлер. Эта фигура сощает в себе возможности коня и ладьи. Канцлер может бить фигуру, находящуюся на одной горизонтали с ним, либо на одной вертикали, либо на расстоянии одного хода «кочергой» — на одно поле по горизонтали и на два по вертикали, либо на два поля по горизонтали и одно по вертикали. При ходе «кочергой» канцлер может перепрыгивать через другие фигуры, но не при ходе по прямой. Например, канцлер не может бить другую фигуру, стоящую на одной горизонтали с ним, если между ними находится ещё одна фигура.

Вам требуется написать программу, которая проверяет, есть ли на доске хотя бы одна пара канцлеров, бьющих друг друга.

Формат входных данных

На стандартном потоке вводится длина стороны квадратной шахматной доски — натуральное число N , такое что $4 \leq N \leq 100$; количество канцлеров, стоящих на доске — натуральное число M , такое что $1 < M \leq N^2$. В последующих M строках ввода программе заданы позиции канцлеров на доске — пары натуральных чисел a_i, b_i , где $i = 1..M$, $0 < a_i, b_i \leq N$, a_i — номер горизонтали, b_i — номер вертикали. Позиции разных канцлеров не совпадают.

Формат выходных данных

Программа проверяет, есть ли на доске хотя бы одна пара канцлеров, бьющих друг друга. Если таких пар нет программа выводит 0 CAPTURES. В остальных случаях выводится любая из обнаруженных пар. Вывод пары осуществляется указанием номеров канцлеров во вводе. Первым выводится меньший номер.

Примеры

тест	ответ
4 3	0 CAPTURES
1 1	
2 2	
4 4	
4 3	1 2
1 1	
1 2	
4 4	

Задача С. Слова на слух

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Петя и Катя играют в игру. Катя называет Пете слово, а Петя пытается по услышанному слову определить, как оно пишется. По правилам игры слова представляются собой строку из одной или нескольких латинских букв. Никакие другие символы в них не используются. Услышанное слово Петя записывает кодом, который представляет собой строку начинающуюся с той же буквы, что и слово, а далее идёт последовательность цифр, составленная по остальным буквам слова. Цифра 0 соответствует одной любой букве из набора A, E, I, O, U, Y. Любая другая цифра с 1 по 9 соответствует одной или двум подряд идущим согласным буквам из одной соответствующей группы похоже звучащих букв: 1 – B, P; 2 – F, V, W; 3 – C, H, K, S; 4 – G, J; 5 – Q, X, Z; 6 – D, T; 7 – L; 8 – M, N; 9 – R. Помогите Пете обработать все варианты написания слова по известной длине слова N и строке кода C . Составьте программу, которая считывает натуральное N и строку C и выводит всевозможные слова в верхнем регистре через символ пробела в алфавитном порядке или -1, если по данному числу N и строке C невозможно составить никакое слово.

Формат входных данных

В первой строке вводится длина слова N ($1 \leq N \leq 5$). Во второй строке код слова C , длина которого не больше N .

Формат выходных данных

Если по заданным длине и строке кода нет ни одного слова, то выведите -1. Иначе необходимо вывести всевозможные слова в верхнем регистре в алфавитном порядке по одному в строке.

Примеры

тест	ответ
3 C6	CDD CDT STD STT
5 C05	-1

Задача D. Нарисованный граф

Петя очень любит рисовать графы в тетрадке в клетку. В углах клеток могут быть нарисованы вершины графа, некоторые соседние вершины могут быть соединены ребрами, при этом ребра могут быть только длины 1 (сторона одной клетки). Нарисовав граф, Петя нумерует вершины и выписывает список ребер.

Однажды Петя потерял листочек с нарисованным графом, но у него остался список ребер. Помогите ему восстановить рисунок.

Формат входных данных

В этой задаче у вас есть несколько входных файлов в тестирующей системе, для каждого из которых требуется отправить на проверку файл с графом. В первой строке исходного файла находится число вершин N и число ребер M . В следующих строках пары чисел a, b , означающие наличие ребра между вершиной с номером a и b .

Гарантируется что граф во входном файле всегда задает какой-то из нарисованных графов.

Формат выходных данных

В файле с ответом граф имеет следующий формат.

- Вершина обозначается символом 'o'
- Ребра обозначаются символами '-', ' ' или ' | '
- Ребро имеет всегда длину 1 и состоит из одного символа
- Пустые места обозначаются символом ' '.

Примеры

тест	ответ
4 4 1 2 2 3 3 4 4 1	o--o . o--o

Решения задач олимпиады «Ломоносов-2015/16» по информатике

Задания для 10-11 классов

Задача А

Требуется за один проход найти самую длинную последовательность. Решение может быть достигнуто следующим образом: храним для префикса найденный максимум и текущий максимум; обновляем максимум если текущий максимум больше/больше или равен найденному.

Задача В

Задача решается с помощью организации стека. При добавлении очередного элемента E_i :

- если на вершине стека расположен такой же, то добавляем E_i в стек,
- иначе, если в стеке больше двух одинаковых элементов, то все одинаковые элементы выталкиваем из стека.

Задача С

После построения дерева выполняем его обход, раскрывая сначала вершины, лежащие ниже рассматриваемой (выполняем поиск в-глубину). Если во время обхода мы считаем, что рассматриваемая вершина должна быть перекрашена, то перекрашиваем ее, и при обходе поддеревьев учитываем - была перекраска или нет.

Задача D

С помощью перебора выполняем разложение данного графа на сетке. Так как в рассматриваемой задаче файлы имеют небольшое размер, ответ может быть получен за разумное время. При этом могут использоваться различные эвристики для отсечения перебора

Задача Е

По заданному набору файлов можно построить несколько правил для определения языка по ключевым словам. Для всех файлов, которые были определены неправильно (их должно быть меньшинство) можно завести отображения хеш-таблицы файла в язык.

Задания для 5-9 классов

Задача А

Аналогично задаче А для 10-11 классов.

Задача В

Требуется только правильно заполнить таблицу, с любой сложностью.

Задача С

При решении задачи нужно построить последовательность. Например, сделать рекурсивное построение, где на каждом шаге производить перебор возможных сочетаний символов из очередного символа кода. Глубину перебора ограничивать по длине пароля.

Задача D

Аналогично задаче D для 10-11 классов

Задача E

Аналогично задаче E для 10-11 классов