

Уральский государственный университет



Association for Computing Machinery

*Четвертьфинал командного
студенческого чемпионата
мира по программированию*

XXXII Командный студенческий
чемпионат мира по
программированию

Сезон 2007/2008

1/4 финала

Северо-восточный
Европейский регион,
Восточный подрегион

Екатеринбург
2007

Четвертьфинал командного студенческого чемпионата мира по
программированию

Под редакцией М. О. Асанова

Учебно-методическое и информационное издание

Компьютерная верстка и дизайн А.Л. Гальперин

Отпечатано в типографии «СКБ Контур»

Издание содержит информационные материалы, связанные с соревнованиями командного студенческого первенства мира ACM. (Association for Computing Machinery).

Печатается по решению организационного комитета.

Организационный комитет соревнований

Директор соревнований

Асанов Магаз Оразкимович

к.ф.-м.н., декан математико-механического факультета УрГУ

Председатель жюри

Клепинин Александр Владимирович

к.ф.-м.н., инженер лаборатории математического обеспечения баз данных УрГУ, системный администратор – программист высшей категории ООО «Прикладные технологии»

Председатель программного комитета

Яковлев Владимир Александрович

Ведущий специалист отдела разработки департамента программных продуктов СКБ-Контур

Группа поддержки интернет–портала acm.timus.ru

Клепинин Александр Владимирович

Яковлев Владимир Александрович,

Программное обеспечение

Клепинин Александр Владимирович

Яковлев Владимир Александрович

Группа технического обеспечения:

Крохалев Евгений Михайлович

ведущий математик Центра Компьютерных Технологий УрГУ

Семигродских Александр Павлович

к.ф.-м.н., начальник отдела развития и администрирования сети Центра Компьютерных Технологий УрГУ

Исполнительный директор соревнований

Гальперин Александр Леонидович

ассистент кафедры математической экономики УрГУ

Программный комитет и жюри:

Васильев Станислав Николаевич

старший научный сотрудник ИММ УрО РАН, доцент кафедры математического анализа и теории функций УрГУ

Васильева Екатерина Владимировна

старший научный сотрудник ИММ УрО РАН

Демидов Андрей Константинович

старший преподаватель кафедры прикладной математики ЮУрГУ

Жевак Алексей Владимирович

аспирант УГАТУ

Крохалев Евгений Михайлович

Лахтин Алексей Станиславович

к.ф.-м.н., доцент кафедры информатики и процессов управления УрГУ

Мироненко Александр Васильевич

научный сотрудник Института математики и механики УрО РАН

Мусин Денис Анатольевич

ведущий специалист отдела разработки департамента программных продуктов

СКБ Контур

Пупырев Сергей Николаевич,

аспирант УрГУ

Программа четвертьфинала Командного студенческого чемпионата мира по программированию ACM-ICPC 2008

Дата	Время	Мероприятие
26.10.07	10 ⁰⁰ – 14 ⁰⁰	Регистрация участников (Тургенева, 4 — ауд. 640)
	15 ⁰⁰ – 15 ⁴⁵	Открытие (актовый зал)
	16 ⁰⁰ – 18 ⁰⁰	Пробный тур (компьютерные классы)
27.10.07	9 ⁰⁰ – 14 ³⁰	Основной тур соревнований (компьютерные классы)
	14 ³⁰ – 15 ³⁰	Обед
	16 ⁰⁰ – 18 ⁰⁰	Разбор задач, закрытие, награждение (актовый зал)
	18 ⁰⁰ – 19 ⁰⁰	Фуршет (столовая)
28.10.07	–	День отъезда

Рейтинг вузов по итогам соревнований по программированию в 2004/05-2006/07 годах

Магаз АСАНОВ

Впервые рейтинг-лист вузов, учитывающий достижения вузов в соревнованиях последних трех лет появился в буклет полуфинальных соревнований в 2001 году и с тех пор неизменно там присутствует. Рейтинг-лист неизменно вызывает интерес широкой публики и в прошлом году О.Б. Христенко автоматизировал процесс вычисления рейтинга по методике, предложенной мной ранее. В частности автоматизация позволила устранить ряд арифметических ошибок и неточностей. Прилагаемая ниже таблица взята с сайта snarknews.info.

За эти годы методика вычисления рейтинга неоднократно корректировалась, но в последние годы остается практически неизменной.

В расчет берутся результаты полуфинальных и финальных соревнований последних трех лет.

Напомню формулу вычисления рейтинга по итогам года:

$$R = k + a + 0,5(b + c) + d,$$

где R — рейтинг вуза за год, k — число задач, решенных сильнейшей командой вуза в полуфинале, a, b, c — премиальные очки за занятые места, полученные первой, второй и третьей командами вуза в полуфинале, d — дополнительные премиальные очки за выступление в финале чемпионата мира.

Премиальные очки за места в полуфинале начисляются следующим образом: 50-е место приносит команде 0.1 очка, 49-е — 0.2 и т.д. с шагом 0.1 до 11 места включительно, которое таким образом «стоит» 4 балла. Далее, 10 место приносит 4.2 балла; 9-е — 4.4; 8-е — 4.6; 7-е — 4.8; 6-е — 5.0; 5-е — 5.3; 4-е — 5.7; 3-е — 6.2; 2-е — 6.8 и, наконец, 1-е — 8.0 баллов.

Дополнительные премиальные очки вуз получает за успешное выступление в финале. Результаты финальных соревнований сложно учитывать, так как организаторы слишком часто меняют правила. Но в последние три года правила остаются неизменными, что и позволяет учитывать результаты финала более точно, чем ранее, когда учитывались только медали.

Золотая медаль — 4, серебряная — 3, бронзовая — 2, попадание в группу команд, решивших столько же задач, сколько и бронзовые призеры — 1,5; в следующую группу классифицированных команд — 1; и, наконец, еще в одну группу классифицированных — 0,5. Дополнительные очки присуждаются за завоевание титулов. Звание чемпионов мира — еще 2 очка, чемпионов Европы — еще 1 очко.

Итоговый рейтинг вуза считается по формуле:

$$R = 2 \cdot R_0 + 1,5 \cdot R_1 + R_2,$$

где R — итоговый рейтинг, R_0 — рейтинг последнего года, R_1 — рейтинг предыдущего года, R_2 — рейтинг года двухлетней давности.

В предлагаемой таблице, в первом столбце, в скобках, указано место, которое занимал вуз в предыдущем рейтинге.

Отметим, что в этом году произошли существенные изменения. На первое место вышел МГУ, прежний лидер — СПбГУ ИТМО, лидировавший в рейтинге последние три года, переместился на второе место. Впрочем, разрыв между лидерами совсем небольшой. Поменялись местами и бывшие третье и четвертое места. На третье, во второй раз (впервые в рейтинге 2003 года) вышел Саратовский университет, на четвертую переместился СПбГУ. Эта пара не сильно различаясь между собой заметно отстает от первых двух и сильно опережает группу из 5 университетов, расположившихся на 5–9 местах. Все эти вузы, исключая, быть может, Алтайский ГТУ, стабильно выступают практически на всех соревнованиях и их позиции в первой десятке достаточно прочны. После двух лет очень успешных выступлений немного отступил Уфимский ГАТУ и вполне может потерять место в десятке сильнейших. Перемещения внутри группы вузов, занимающих 11–20 места также в основном происходит внутри себя. Разве лишь, поднялся Вологодский ГПУ, вытеснивший Киргизско–Славянский университет. Из многочисленной общей группы можно отметить стремительное продвижение вверх Ставропольского и Удмуртского университетов, за два года поднявшихся из небытия на 27 и 33 места соответственно.

1.(2)	Moscow SU	28.500	24.500	23.950	113.150
2.(1)	St.Petersburg IFMO	28.650	22.150	23.650	109.175
3.(4)	Saratov SU	16.200	21.900	19.600	88.250
4.(3)	St.Petersburg SU	19.000	21.200	15.200	81.200
5.(6)	Ural SU	14.800	17.950	14.050	69.825
6.(9)	Petrozavodsk SU	12.200	13.450	16.400	65.175
7.(10)	Novosibirsk SU	16.300	10.300	16.300	64.350
8.(5)	Altai STU	12.550	20.200	9.200	61.250
9.(7)	Belarus SU	15.650	10.350	12.600	56.375
10.(8)	Ufa SATU	16.800	15.600	4.600	49.400
11.(15)	South USU	9.200	12.700	8.700	45.650
12.(18)	Orel STU	10.900	7.500	11.200	44.550
13.(11)	Perm SU	13.300	10.100	7.050	42.550
14.(12)	Izhevsk STU	14.800	8.000	7.750	42.300

15.(14)	Moscow IPT	12.200	10.000	7.050	41.300
16.(26)	Vologda SPU	8.700	7.900	10.000	40.550
17.(19)	Tbilisi SU	10.700	10.400	7.000	40.300
18.(16)	Latvia U	9.100	12.300	5.200	37.950
19.(20)	Tartu U	8.300	7.700	9.000	37.850
20.(17)	St.Petersburg ETU	9.350	8.400	6.900	35.750
21.(13)	Kirgizsko-Russian Slavic U	10.500	12.300	3.000	34.950
22.(21)	Yaroslavl SU	6.400	12.400	4.100	33.200
23.(31)	Kazakhstan NU	4.000	6.200	9.400	32.100
24.(25)	Ivanovo St Power U	8.500	9.800	4.300	31.800
25.(23)	Kazakh-British TU	4.000	13.800	3.000	30.700
26.(24)	Moscow EPhI	11.000	5.700	4.900	29.350
27.(60)	Stavropol SU	0.000	4.000	11.100	28.200
28.(60)	St.Petersburg TU	3.000	4.000	9.200	27.400
29.(32)	Vilnius U	6.700	3.000	6.800	24.800
30.(41)	Yerevan SU-	5.000	3.000	7.400	24.300
31.(33)	Moscow Energetical U	5.000	4.000	6.600	24.200
32.(42)	Rybinsk SAAT	5.000	2.000	7.200	22.400
33.(40)	Udmurt SU	0.000	7.800	5.300	22.300
34.(28)	Tomsk SU	8.600	3.000	4.500	22.100
35.(34)	Tashkent U of IT	5.000	4.000	4.000	19.000
36.(27)	Far Eastern SU	6.500	4.100	3.000	18.650
37.(39)	Kazakh NTU	3.000	5.800	3.000	17.700
38.(51)	Moscow Aviation U	3.000	3.000	4.000	15.500
38.(37)	Siberian SU of TI	5.000	3.000	3.000	15.500
38.(37)	Tyumen SOGU	5.000	3.000	3.000	15.500
41.(36)	Kaliningrad SU	0.000	6.000	3.000	15.000
42.(29)	Gomel SU	6.800	5.200	0.000	14.600
43.(42)	Baikal SU Economics & Law	5.000	2.000	3.000	14.000
43.(42)	Siberian St Indust. U	5.000	2.000	3.000	14.000
43.(47)	Almaty Suleyman Demirel U	5.000	2.000	3.000	14.000
46.(22)	N.Novgorod SU	13.600	0.000	0.000	13.600
47.(47)	Khakassia SU	3.000	3.000	3.000	13.500
48.(35)	Kazan SU	5.000	5.400	0.000	13.100
49.(54)	Karaganda STU	4.000	2.000	3.000	13.000
49.(51)	Samara Aerospace U	5.000	0.000	4.000	13.000
49.(46)	Tomsk USUR	4.000	2.000	3.000	13.000
52.(47)	Novosibirsk STU	5.000	1.000	3.000	12.500
52.(55)	Tomsk Polytechnical U	5.000	1.000	3.000	12.500

54.(45)	Krasnoyarsk STU	3.000	2.000	3.000	12.000
54.(51)	Omsk SU	3.000	2.000	3.000	12.000
56.(60)	East-Siberian U of Tech	4.000	1.000	3.000	11.500
57.(30)	Orenburg SUv	8.400	2.000	0.000	11.400
58.(68)	Astana Eurasian NU	3.000	1.000	3.000	10.500
58.(59).	Irkutsk STU	3.000	1.000	3.000	10.500
58.(64)	Rus-Armenian SU	3.000	1.000	3.000	10.500
58.(68)	U World Econ & Diplomacy	3.000	1.000	3.000	10.500
62.(73)	American U in Central Asia	4.000	0.000	3.000	10.000
62.(-)	Tyumen SU	0.000	0.000	5.000	10.000
64.(79)	Uzbekistan NU	2.000	1.000	3.000	9.500
65.(90)	Altai SU	0.000	2.000	3.000	9.000
65.(68)	Kemerovo SU	3.000	0.000	3.000	9.000
65.(90)	Pskov SPI	0.000	2.000	3.000	9.000
65.(81)	Siberian SAU	3.000	0.000	3.000	9.000
65.(81)	TSU Sokhumi branch	3.000	0.000	3.000	9.000
65.(90)	Volgograd SU	0.000	2.000	3.000	9.000
65.(81)	Westminster IU in Tashkent	3.000	0.000	3.000	9.000
72.(94)	Rubtsovsk II	2.000	0.000	3.000	8.000
72.(-)	Vyatka SHU	0.000	0.000	4.000	8.000
74.(95)	Amur SU	0.000	1.000	3.000	7.500
74.(90)	Krasnoyarsk SU	0.000	1.000	3.000	7.500
76.(57)	Sykt'yvkar SU	6.100	0.000	0.000	6.100
77.(-)	Adyghe SU	0.000	0.000	3.000	6.000
77.(-)	Almaty I of PET	0.000	0.000	3.000	6.000
77.(-)	Astana-Moscow SU	0.000	0.000	3.000	6.000
77.(60)	Astrakhan STU	0.000	4.000	0.000	6.000
77.(-)	Atyrau SU	0.000	0.000	3.000	6.000
77.(-)	Belarus St. U. of I&R	0.000	0.000	3.000	6.000
77.(-)	Belarus-Russia U	0.000	0.000	3.000	6.000
77.(-)	Chita SU	0.000	0.000	3.000	6.000
77.(-)	Fergana B of Tashkent UIT	0.000	0.000	3.000	6.000
77.(58)	Karelian SPU	6.000	0.000	0.000	6.000
77.(-)	Kutaisi SU	0.000	0.000	3.000	6.000
77.(-)	North Caucas STU	0.000	0.000	3.000	6.000
77.(-)	Novokuznetsk Dep of Kem.SU	0.000	0.000	3.000	6.000
77.(-)	Siberian SU of Transport	0.000	0.000	3.000	6.000
77.(-)	Tbilisi I Black Sea U	0.000	0.000	3.000	6.000
77.(-)	Volgograd STU	0.000	0.000	3.000	6.000

77.(-)	Yerevan SEUA	0.000	0.000	3.000	6.000
94.(108)	Ferghana PI	1.000	0.000	2.000	5.000
94.(55)	Kazan STU	5.000	0.000	0.000	5.000
94.(64)	Samara SU	5.000	0.000	0.000	5.000
97.(68)	East Kazakhstan STU	3.000	1.000	0.000	4.500
97.(73)	Kaunas TU	0.000	3.000	0.000	4.500
97.(73)	Pomor SU	0.000	3.000	0.000	4.500
100.(73)	Cherepovets SU	4.000	0.000	0.000	4.000
100.(73)	Grodno SU	4.000	0.000	0.000	4.000
100.(94)	Karaganda SU	2.000	0.000	1.000	4.000
100.(-)	Namangan EPI	0.000	0.000	2.000	4.000
100.(-)	Osh STU	0.000	0.000	2.000	4.000
100.(-)	Smarkand B of Tashkent UIT	0.000	0.000	2.000	4.000
100.(-)	Tashkent I of AC	0.000	0.000	2.000	4.000
107.(95)	North Kazakhstan SU	0.000	1.000	1.000	3.500
107.(79)	SEU Armenia	2.000	1.000	0.000	3.500
109.(81)	Arkhangelsk STU	3.000	0.000	0.000	3.000
109.(81)	Kazakh I of MEP	3.000	0.000	0.000	3.000
109.(81)	Komsomolsk-on-Amur STU	3.000	0.000	0.000	3.000
109.(81)	Kuban STU	3.000	0.000	0.000	3.000
109(108)	Kyrgyz NU	1.000	0.000	1.000	3.000
109.(81)	Novosibirsk SPU	3.000	0.000	0.000	3.000
109.(81)	Sergiev Posad BrMSA IECS	3.000	0.000	0.000	3.000
109.(81)	Tashkent STU	3.000	0.000	0.000	3.000
109(108)	Urganch SU	1.000	0.000	1.000	3.000
109.(81)	Vologda STU	3.000	0.000	0.000	3.000
119.(-)	Batumi SMA	0.000	0.000	1.000	2.000
119.(94)	Bukhara SU	2.000	0.000	0.000	2.000
119.(-)	Karakalpak SU	0.000	0.000	1.000	2.000
119.(-)	Karshi SU	0.000	0.000	1.000	2.000
119.(94)	Kostanay SU	2.000	0.000	0.000	2.000
119.(-)	Rudny II	0.000	0.000	1.000	2.000
119.(-)	Tashkent SU of E	0.000	0.000	1.000	2.000
119.(-)	Urganch B of Tashkent UIT	0.000	0.000	1.000	2.000
119.(94)	Vladivostok SUE	2.000	0.000	0.000	2.000
128.(95)	Bijsk Technological I	0.000	1.000	0.000	1.500
128.(95)	Pacific Ocean SU	0.000	1.000	0.000	1.500
130(108)	Andijan EEI	1.000	0.000	0.000	1.000
130(108)	Atatyurk-Alatoo U Kyrgyzstan	1.000	0.000	0.000	1.000

130(108)	Bukhara TI of FLI	1.000	0.000	0.000	1.000
130(108)	Kutaisi STU	1.000	0.000	0.000	1.000
130(108)	Pavlodar SU	1.000	0.000	0.000	1.000
130(108)	Samarkand I of ES	1.000	0.000	0.000	1.000
130(108)	Samarkand SU	1.000	0.000	0.000	1.000
130(108)	Tashkent Islamic U	1.000	0.000	0.000	1.000

Команды Северо-Восточного региона в финалах чемпионатов мира

Мигаз АСАНОВ

Первыми Российскими университетами, команды которых добились права выступления в финале мирового первенства в 1996 году, стали СПбГУ ИТМО и МГУ. Успешное выступление первопроходцев привело к созданию в том же году Северо-Восточного европейского региона. Ниже в таблице приведены результаты выступления команд региона в финалах чемпионатов мира. Вузы упорядочены по количеству выступлений в финалах. В последующих столбцах указано сколько медалей того или иного достоинства получено командами университета. Напомним, что в финалах, начиная с 2000 года, присуждаются медали. Правила вручения медалей несколько раз менялись, в последние годы вручается по 4 золотых, серебряных и бронзовых медалей. Обращает на себя внимание поразительная стабильность команд СПбГУ ИТМО и МГУ, завоевавших по 6 медалей из 8 возможных. Правда, достижения СПбГУ ИТМО выглядят весомей, 5 медалей из 6 – золотые.

		Total	G	S	B
1	St Petersburg IFMO	12	5	1	-
2	Moscow SU	12	2	-	4
3	St Petersburg SU	10	2	1	-
4	Belarusian SU	7	1	-	-
5	Saratov SU	5	1	3	-
6	Novosibirsk SU	5	-	1	-
7	Ural SU	5	-	-	1
8	Petrozavodsk SU	4	-	-	1
9-10	N. Novgorod SU	3	-	-	-
9-10	South USU	3	-	-	-
11-12	Altai STU	2	1	-	-
11-12	Perm SU	2	1	-	-

		Total	G	S	B
13	Izhevsk STU	2	-	1	1
14	Ufa SATU	2	-	-	1
15-16	Orel STU	2	-	-	-
15-16	University of Tartu	2	-	-	-
17-23	Kyrgyz-Russian SU	1	-	-	-
17-23	Kazakh-British TU	1	-	-	-
17-23	Kazakh NU	1	-	-	-
17-23	Samara MNU	1	-	-	-
17-23	Stavropol SU	1	-	-	-
17-23	Ural STU	1	-	-	-
17-23	Yaroslavle SU	1	-	-	-

Университеты Урала в полуфиналах чемпионатов мира

Первые полуфинальные соревнования Северо-Восточного Европейского региона, объединяющего все постсоветское пространство, кроме Украины и Молдовы, состоялись в ноябре 1996 года. Проведено уже 11 полуфиналов. За эти годы в них участвовали команды 20 уральских вузов. Полуфинал чемпионата мира одновременно является заключительным туром Всероссийской командной студенческой олимпиады по программированию. Ниже в таблице приведены результаты выступления команд Урала в этих соревнованиях. В третьем столбце указано общее количество команд вуза, выступавших в соревнованиях, в последующих столбцах указано количество дипломов I, II и III степеней, полученных командами вуза.

Во всех 11 полуфиналах выступали команды только двух уральских вузов - УрГУ и УГАТУ. Пермский ГУ не был представлен в полуфинале 2000 года, ЮУрГУ - 1999 г, Ижевский ГТУ - 2000 и 2001 гг..

			I ст.	II ст.	III ст.
1	Ural SU	28	2	10	7
2	Ufa SATU	15	1	2	3
3	Perm SU	14	0	3	2
4	South USU	13	1	1	3
5	Izhevsk STU	11	0	2	3
6	Ural STU	9	0	2	1
7	Orenburg SU	9	0	1	3
8	Tyumen SOGU	9	0	0	0

			I ст.	II ст.	III ст.
9	Perm STU	5	0	1	1
10	Chelyabinsk SU	5	0	0	1
11	Magnitogorsk STU	4	0	0	1
12	Bashkir SU	3	0	1	1
13	Udmurt SU	3	0	0	1
14-16	Syktvkar SU	2	0	0	0
	Tyumen SU	2	0	0	0
	Ufa Tech oOil U	2	0	0	0
17-20	Kurgan SU	1	0	0	0
	Magnitogorsk SU	1	0	0	0
	Orenburg SPU	1	0	0	0
	Perm SPU	1	0	0	0
	ВСЕГО		4	23	27

Университеты Урала в финалах чемпионатов мира

Первой уральской командой, получившей право выступления в финале, была команда УГТУ–УПИ, выступившая в финале 1998 года. С тех пор университеты Урала были представлены во всех финалах, кроме 2002 и 2003 годов. За эти годы команды 6 уральских вузов побывали в финалах. Наибольшее представительство было в 2005 году, когда в Шанхае выступили сразу 4 уральские команды. Ниже в таблице приведены результаты выступления команд Урала в финалах чемпионатов мира. Вузы упорядочены по количеству выступлений в финалах. Напомним, что в финалах, начиная с 2000 года, присуждаются медали. Правила вручения медалей несколько раз менялись, в последние годы вручается по 4 золотых, серебряных и бронзовых медалей.

	Вуз	Год, место проведения	Результат
1	Ural SU	5 1999, Эйндховен, Нидерланды	не классифицирована
		2001, Ванкувер, Канада	14 место, бронзовая медаль
		2005, Шанхай, Китай	17 место
		2006, Сан-Антонио, США	не классифицирована
		2007, Токио, Япония	26 место

	Вуз		Год, место проведения	Результат
2	South USU	3	2000, Орландо, США	15 место
			2001, Ванкувер, Канада	29 место
			2006, Сан-Антонио, США	19 место
3	Perm SU	2	2004, Прага, Чехия	4 место, золотая медаль
			2005, Шанхай, Китай	17 место
4	Izhevsk STU	2	2004, Прага, Чехия	8 место, серебряная медаль
			2005, Шанхай, Китай	9 место, бронзовая медаль
5	Ufa SATU	2	2005, Шанхай, Китай	13 место
			2006, Сан-Антонио, США	11 место, бронзовая медаль
6	Ural STU	1	1998, Атланта, США	17 место

Задачи XI чемпионата Урала по спортивному программированию

Задача А. Автодополнение

Автор: Ден Расковалов

Японцы бесконечно влюблены в технику, которая их окружает. Они внимательно следят за всеми техническими новинками и стараются пользоваться самыми современными и «умными» из них. У Дена и Сергея есть гениальный план: они хотят создать текстовый редактор, который покорит японцев. Важнейшей уберинтеллектуальной функцией редактора должна стать функция автодополнения. Если пользователь набрал несколько первых букв слова, редактор должен предложить ему самые правдоподобные окончания.

Ден и Сергей уже собрали огромное количество японских текстов. Для каждого слова японского языка они посчитали число раз, которое оно встречается в текстах. Если пользователь уже ввел несколько букв, то редактор должен показать не более десяти самых часто употребляемых слов, начинающихся со введенных пользователем букв, отсортированных по убыванию частоты упоминания.

Помогите Сергею с Деном перевернуть рынок текстовых редакторов.

Формат входного файла

В первой строке находится единственное число N ($1 \leq N \leq 100000$) — количество слов в найденных текстах. Каждая из следующих N строк содержит слово w_i (непустая последовательность строчных латинских букв длиной не более 15) и целое число n_i ($1 \leq n_i \leq 1000000$) — число раз, которое встречается это слово в текстах. Слово и число разделены единственным пробелом. Ни одно слово не повторяется более одного раза. В $(N + 2)$ -ой строке находится число M ($1 \leq M \leq 15000$). В следующих M строках содержатся слова u_i (непустая последовательность строчных латинских букв длиной не более 15) — начала слов, введенных пользователем.

Формат выходного файла

Для каждой из M строк необходимо вывести наиболее часто употребляемые японские слова, начинающихся с u_i , в порядке убывания частоты. В случае совпадения частот слова необходимо сортировать по алфавиту. Если существует больше десяти возможных вариантов, то вывести нужно лишь первые десять из них. Варианты дополнения для каждого слова необходимо разделять переводами строк.

Пример

input.txt	output.txt
5	kanojo
kare 10	kare
kanojo 20	korosu
karetachi 1	karetachi
korosu 7	
sakura 3	kanojo
3	kare
k	karetachi
ka	
kar	kare
	karetachi

Задача В. Танцевальная революция

Автор: Денис Мусин

Dance Dance Revolution, or DDR (known as Dancing Stage in Europe) is a music video games series produced by Konami. As of 2005, over 90 official versions have been produced, including those for home video game consoles. The game is played on a dance pad with four arrow panels: left, down, up, and right. These panels are pressed using the player's feet, in response to arrows that appear on the screen in front of the player. The arrows are synchronized to the general rhythm or beat of a chosen song, and success is dependent on the player's ability to time and position his or her steps accordingly.

from Wikipedia, the Free Encyclopedia

Танцевальные автоматы DDR очень популярны в Японии. Постепенно они завоевали сердца и ноги многих поклонников со всего мира и появились даже на Урале. Но для суровых уральских условий потребовалось разработать новый вариант, который называли DDR Ural Mix (сокращенно DDRUMix), основанный на популярном в Японии и, к сожалению, распространенном только там автомате DDR 5th Mix.

Основная конструкция та же: игрок должен в такт выбранной музыке наступать ногами на определенные клавиши в заданной последовательности, таким образом, танцуя. Пока играет музыка, на экране снизу вверх прокручиваются строчки со стрелками; когда строчка доходит до верхней полосы, стрелка вспыхивает ярче, и в этот момент игрок должен наступить на соответствующую клавишу. В зависимости от нажатых клавиш и моментов нажатий действия игрока в каждом периоде, где от игрока требуется нажатие на одну из клавиш, оцениваются как «PERFECT», «GREAT», «GOOD», «BOO» или «MISS». В конце игрок видит общий счет, бонусные очки, и сколько разных шагов он сделал. Также игрок получает оценку за исполнение от «E» (плохо) до «AAA» (если все шаги были ранга «PERFECT»). В самом конце игрок получает общую оценку за все песни, которые он протанцевал.

Поскольку проект совместный, разработку ПО автомата DDRUMix японцы решили доверить ведущим уральским программистам, то есть вам. Первой вашей задачей будет создать систему подсчета очков, заработанных игроком во время одного танца на DDRUMix. Для этого вам необходимо познакомиться с алгоритмом подсчета очков.

Все песни разделены на три уровня сложности: Easy, Medium и Hard (в зависимости от их ритма: Easy — 1 такт в секунду, Medium для 2 тактов в секунду, а Hard для 4 тактов). Промежуток между двумя тактами называется *периодом*, например, период для сложности Medium составляет 500 миллисекунд. Периоды, в которые от игрока ожидается какое-то действие, называются *степами*. Каждая песня записана как последовательность тактов с указанием, какой шаг должен сделать игрок в соответствующий период (ступ «вверх», «вниз», «влево» или «вправо»). Такт может быть пустым, когда никуда наступать не надо. Действия игрока в течение «пустого» периода не оцениваются.

Правила оценки действий игрока в одном степе следующие:

1. Все не первые шаги в течение одного ступа не оцениваются.
2. Если игрок не нажал ни одной клавиши в течение всего ступа, он получает оценку «BOO».
3. Если первый шаг в ступе был неправильным, он получает «MISS».
4. Если нужная клавиша была нажата первым шагом в соответствующем ступе, и с начала периода прошло менее 40% периода, то игрок получает «PERFECT», менее 70% — «GREAT», и не менее 70% — «GOOD».

Общий счет игрока складывается из очков за танец, бонуса за комбо и бонуса за уровень исполнения.

Как считать очки за танец:

Пусть $\text{Diff} = 1$ для уровня Easy, 2 для Medium и 3 для Hard.

Пусть $B = 1000000 \cdot \text{Diff}$.

Пусть $p = \text{score multiplier}$ («PERFECT» = 10, «GREAT» = 5, остальное = 0).

Обозначим через N общее число стенов (тактов с действиями) и определим число $S = (N \cdot (N + 1))/2$. Пусть n — номер текущего стенов (от 1 до N), тогда количество очков за этот стенов равно:

$$\text{StepScore} = p \cdot n \cdot (B \text{ div } S)$$

(здесь $B \text{ div } S$ означает целую часть от деления B на S). Общий счет за танец равен сумме очков за все стенов.

Как считать бонус за комбо:

Комбо — это последовательность подряд идущих стенов, состоящая только из «PERFECT», «GREAT» и «GOOD» (конечно, внутри комбо могут также быть пустые такты, но они не учитываются и не увеличивают длины комбо).

Пусть $q = \text{step multiplier}$ («PERFECT» = 55, «GREAT» = 33, остальное = 0). Каждый стенов в комбо оценивается в $q \cdot c$ очков, где c — номер стенов в текущем комбо. Общий бонус за комбо равен сумме очков за все стенов во всех комбо.

Как считать бонус за уровень исполнения:

Каждый стенов добавляет или вычитает баллы «Dance Points», которые начисляются следующим образом: «PERFECT» добавляет 2 «Dance Points», «GREAT» добавляет 1 «Dance Point», «GOOD» не добавляет ничего, «BOO» вычитает 4 «Dance Points», а «MISS» вычитает 5 «Dance Points».

Бонус за уровень исполнения начисляется так:

AAA — 10 000 000 (если все стенов были сыграны «PERFECT»)

AA — 1 000 000 (все стенов «PERFECT» или «GREAT»)

A — 100 000 (не менее 80% от максимально возможного количества «Dance Points»)

B — 10 000 (не менее 64% от максимума)

C — 1 000 (не менее 50% от максимума)

D — 100 (менее, чем 50% от максимума)

E — 0 (если в течении игры «Dance Points» опускались ниже нуля).

Зная всю информацию про одну игру (длительность песни, ее сложность, последовательность тактов и нажатий клавиш), вы должны вычислить общий счет и вывести статистику по игре.

Формат входного файла

Первая строка содержит длительность песни в формате `m:ss` (длительность каждой песни не менее 10 секунд и менее 10 минут) и сложность (Easy, Medium или Hard). Вторая строка содержит последовательность тактов. Такт может иметь знак вверх, вниз, влево или вправо (обозначаются соответственно буквами U, D, L и R) или быть пустым (обозначается буквой N). Песня не может состоять из одних пустых тактов. Оставшиеся строки содержат лог действий игрока, где каждому нажатию соответствует одна строка в следующем формате:

`<time> <action>`

где `<time>` — время в миллисекундах от начала песни, а `<action>` обозначается одним из четырех символов U, D, L или R. Заметьте, что человеческая гибкость не позволяет нажимать клавиши чаще, чем раз в 50 миллисекунд.

Формат выходного файла

Вывести статистику игрока. В первой строке выведите количество оценок «PERFECT», зарабо-

танных игроком, во второй — «GREAT», в третьей — «GOOD», в четвертой — «BOO», в пятой — «MISS». Шестая строка — разделитель из 20 дефисов («-»). В седьмой строке выведете очки за танец, в восьмой — бонус за комбо, в девятой — бонус за уровень исполнения и в десятой — суммарное количество очков, заработанных игроком. Для пояснения формата вывода смотрите пример.

Пример

input.txt	output.txt
0:30 Easy	Perfect: 7
NNNNNLNNRNUNNUNNNDDNNUNNNNRNU	Great: 2
5201 L	Good: 1
8475 R	Boo: 0
8735 L	Miss: 0
10000 U	-----
13200 U	Stage Score: 7181495
17284 D	Combo Bonus: 2233
18355 D	Dance Level Bonus: 100000
19124 D	Total Score: 7283728
22222 U	
27543 R	
29923 U	

Задача С. Одноклассники-3

Автор: Ден Расковалов
Подготовка: Сергей Пупырев
Условие: Станислав Васильев

Таня выросла, и теперь она работает учительницей информатики в школе. Недавно в её классе установили новое японское программное обеспечение. Теперь каждый компьютер может общаться с соседними компьютерами класса по одному из двух протоколов — японскому или европейскому — и может переключаться с одного протокола на другой. Получив команду на смену протокола, компьютер автоматически пересылает эту команду всем соседним компьютерам и тут же переключается сам. К сожалению, протоколы несовместимы. Настолько несовместимы, что даже простую команду на смену протокола смогут принять только те из соседей, которые работали в том же самом протоколе, что и компьютер, пославший команду. Заметьте, что каждый из соседних компьютеров отправит сигнал о смене протокола и пославшему сигнал компьютеру, но тот его уже не получит — ведь он уже переключился на новый протокол.

В начале занятия Таня обнаружила, что при установке нового программного обеспечения каждому компьютеру был случайным образом назначен один из двух доступных протоколов. Для проведения занятия Тане надо срочно перевести все компьютеры в один протокол.

Таня может попросить кого-то из своих учеников сменить протокол, например, с японского на европейский. Тогда его компьютер и все связанные с ним напрямую или через другие (но только работающие по японскому протоколу) компьютеры перейдут на европейский протокол. Остальные компьютеры, например, уже работавшие в европейском протоколе или просто не связанные с первым (напрямую или через другие компьютеры с японским протоколом), так и будут работать по европейскому протоколу. Аналогичная картина будет, если на каком-то компьютере сделать смену с европейского на японский протоколы. Помогите Тане переключить все компьютеры в какой-нибудь один протокол за минимальное количество обращений к ученикам.

Формат входного файла

В первой строке записаны числа N и M – число компьютеров в сети и число сетевых соединений между ними ($1 \leq N \leq 50$). В следующей строке через пробел записано N букв «Е» или «J». Если на i -м компьютере включен европейский протокол, то i -я буква будет «Е», если японский, – то «J». Далее идут M строк, в каждой из которых находятся два различных числа a_i и b_i ($1 \leq a_i, b_i \leq N$) – номера компьютеров, соединенных сетью непосредственно. Известно, что с каждого компьютера можно через существующие сетевые соединения передать сообщение на любой.

Формат выходного файла

В первой строке должно находиться число K – наименьшее число просьб о смене протокола, с которыми Таня должна обратиться к ученикам, чтобы сменить протокол всех компьютеров на какой-то один. Далее должны следовать K строк с описанием просьб. Просьба сменить протокол i -го компьютера на европейский записывается в виде « i Е», на японский – « i J». Если существует несколько оптимальных решений, выведите любое из них.

Пример

input.txt	output.txt
5 5	1
Е Е Е J J	1 J
1 2	
1 3	
1 4	
4 2	
5 2	

Задача D. Иероглифы

Авторы: Евгений Крохалев и Владимир Яковлев

Подготовка: Евгений Крохалев

Условие: Евгений Крохалев

Вова очень нравится аниме. Он настолько увлекся этим творчеством, что даже научился сносно общаться со своими японскими друзьями на их родном языке. Но вот при электронной переписке Вове до сих пор приходится пользоваться латиницей. Он хочет сразу печатать иероглифы со своей клавиатуры. Сергей, его товарищ по команде, написал маленькое приложение, позволяющее писать иероглифы, набирая при этом на клавиатуре латинские символы. Для этого, так как иероглифов значительно больше, чем букв латинского алфавита, Сергей создал специальный справочник, в котором каждый иероглиф представляется в виде последовательности из двух строчных букв латинского алфавита. Как только программка осознаёт, что набрана последовательность латинских букв, соответствующую некоторому иероглифу, то эта последовательность заменяется соответствующим иероглифом.

Когда Вова стал работать с программкой Сергея, ему быстро надоело заглядывать в справочник соответствия комбинаций букв и иероглифов за подсказкой. Помогите Сергею доработать свою программку, чтобы при вводе некоторой латинской буквы на экране автоматически отображалась подсказка, помогающая продолжить ее до корректного иероглифа.

Формат входного файла

В первой строке находится количество иероглифов в справочнике Сергея N ($1 \leq N \leq 1000$). В следующих N строках содержатся последовательности латинских символов из справочника, со-

ответствующие японским иероглифам, по одной последовательности в строке. Каждая такая последовательность содержит ровно две буквы латинского алфавита. В следующей строке содержится введенная Вовой буква.

Формат выходного файла

Выведите список иероглифов из справочника, которые начинаются на введенный Вовой символ, по одному в строке в любом порядке. Если таких иероглифов нет, то ничего выводить не надо.

Пример

input.txt	output.txt
6	ka
na	ki
no	ku
ni	
ki	
ka	
ku	
k	

Задача Е. Сортировка по-японски

Автор: *Владимир Яковлев*
Подготовка: *Владимир Яковлев*
Условие: *Станислав Васильев*

Японцы любят фотографировать. Но фотографии, сделанные цифровым фотоаппаратом, не всегда корректно сортируются на компьютере: например, файл `photo12.jpg` почему-то находится между файлами `photo1.jpg` и `photo2.jpg` — и фотографии при просмотре идут не в том порядке. Некоторые японцы используют систему, которая сортирует строки по другим, японским, правилам, и у них числа сортируются естественным образом:

лексикографический порядок	японский порядок
photo2	photo2
photox1	photox1
photox10	photox2
photox2	photox10

Полностью эти правила излагаться в условии не будут. Они достаточно просты и логичны. Попробуйте проникнуться духом Японии и сами определить их.

Формат входного файла

На вход подается несколько строк. Строки содержат только строчные латинские буквы и цифры. Общий объем входных данных не превосходит 100 КБ.

Формат выходного файла

Выведите те же строки, но отсортированные в соответствии с японскими правилами.

Пример

input.txt	output.txt
photox10 photox1 photo2 photox2	photo2 photox1 photox2 photox10

Задача F. Перебор паролей

Автор: *Владимир Яковлев*

Подготовка: *Владимир Яковлев*

Условие: *Станислав Васильев и Владимир Яковлев*

После поездки в Японию Вова забыл пароль к своему архиву. К счастью, студенты УрГУ имеют доступ к мощному суперкомпьютеру MBC-1000, и для решения сложных математических задач Вове могут выделить M сверхбыстрых процессоров. Вова хочет использовать суперкомпьютер для простого перебора паролей. Он помнит, что длина пароля не превосходит N , и пароль состоит из строчных латинских букв. Сначала он хочет перебрать все пароли длины 1 в лексикографическом порядке (то есть $a, b, \dots, z$), потом — все длины 2 в том же порядке (то есть $aa, ab, \dots, zz$) и т.д.

Чтобы использовать суперкомпьютер максимально эффективно, пароли надо распределить на все процессоры примерно поровну: первую часть паролей перебирает первый процессор, вторую часть второй и т.д. Если поровну разделить нельзя, то пусть первые несколько процессоров перебирают на 1 пароль больше, чем остальные. Чтобы запустить перебор, Вове нужно знать, какой диапазон паролей должен перебирать каждый процессор. Помогите ему в этом.

Формат входного файла

В единственной строке записано два целых числа N и M ($1 \leq N, M \leq 50$). Гарантируется, что количество паролей не меньше количества процессоров.

Формат выходного файла

Вывод должен состоять из M строк. В i -й строке должен быть записан диапазон паролей для i -го процессора. Формат смотрите в примере.

Пример

input.txt	output.txt
5 4	a-fssts fsssu-mmmnn mmmmo-tgggg tgggh-zzzzz

Задача G. Сакура и статистика

Авторы: *Станислав Васильев и Сергей Пузырев*

Подготовка: *Сергей Пузырев*

Условие: *Стас Васильев*

В Токио Денис познакомился с японским цветоводом. Цветовод рассказал, что у его сакуры один из побегов утром иногда распрямляется, а к вечеру всегда снова склоняется вниз. Распрямляется побег не более одного раза в день, но время, когда он распрямится или согнется обратно, почти

невозможно предсказать. С помощью автоматической системы слежения, которая каждую секунду отмечает состояние побега, цветовод собирает статистику и публикует её на своем сайте. Статистика публикуется в виде картинки: белый пиксель означает согнутый побег, черный — распрямившийся, показания одного дня записываются в столбик сверху вниз. Получается довольно странная картинка — набор вертикальных отрезков, не более одного в столбце. Денис сказал, что рисовать такую картинку попиксельно неэкономично, эффективнее будет отрисовывать её с помощью прямоугольников, например, в JavaScript. Естественно, цветовод попросил Дениса написать программу, которая сама рисует картинку со статистикой. С программой на JavaScript Денис справится сам, а вам надо найти минимальное множество прямоугольников, объединение которых совпадает с исходной картинкой.

Формат входного файла

Первая строка содержит N и M — размеры картинки по вертикали и горизонтали соответственно ($1 \leq N, M \leq 50$). Последующие строки содержат саму битовую картинку. Каждая из следующих N строк содержит по M символов. Символом «1» обозначен черный пиксель, символом «0» — белый.

Формат выходного файла

В первой строке выведите количество найденных прямоугольников. В следующих строках выведите координаты противоположных углов этих прямоугольников. Считаем, что ось OX направлена вниз, OY — вправо. Если оптимальных решений несколько, выведите любое из них.

Примеры

input.txt	output.txt
3 3 010 111 010	2 1 2 3 2 2 1 2 3
4 5 00000 11100 11010 10000	4 2 1 2 3 3 1 3 2 2 1 4 1 3 4 3 4

Задача Н. Японский пазл

Автор: Станислав Васильев
Подготовка: Владимир Яковлев
Условие: Станислав Васильев

Один из преподавателей УрГУ приобрел в Японии забавную игрушку: маленький самолетик и много пластмассовых пластинок, из которых, как из кусочков пазла, можно сложить путь для самолетика. Складывать пластинки можно как угодно, но если сложить пазл правильно, то получится карта Японии, и самолетик будет объезжать по замкнутому маршруту все крупные достопримечательности Японии. Как вы думаете, что же сказала жена преподавателя, увидев такую игрушку — «Спасибо!» или «Зачем надо было тратить столько денег на всякую ерунду»? Нет, она сказала: «Жаль, что не купил несколько штук, можно было бы составить гораздо более длинный путь».

Представьте себе, что у вас есть много пластинок с отрезками пути. Какой самый длинный замкнутый путь вы сможете составить?

Кусочки пазла можно считать квадратами одинакового размера, путь всегда соединяет середины двух сторон квадрата. То есть кусочки бывают двух типов: прямые и повороты.

Формат входного файла

В единственной строке записаны два целых числа — количество прямых S и количество поворотов T ($0 \leq S, T \leq 1000, S + T > 0$).

Формат выходного файла

В первой строке выведите наибольшее количество кусочков N , которые можно использовать для построения пути. Во второй строке выведите сам путь в следующем формате: строка длины N , состоящая из букв «F», «L» и «R». Здесь «F» означает, что следующий кусочек пути должен идти прямо, «L» означает поворот налево, «R» означает поворот направо. Общее количество букв «F» в строке не должно превосходить S , суммарное количество «L» и «R» не должно превосходить T . Путь должен быть замкнут (последний квадратик должен стыковаться с первым), и квадраты не могут накладываться друг на друга. Если из имеющихся квадратиков невозможно сложить такой путь, то выведите «Atawazu»¹.

Примеры

input.txt	output.txt
5 6	10 FLLFRLFLF
49 3	Atawazu

Задача I. Пирамида декана-3

Автор: Евгений Кроталев

Подготовка: Евгений Кроталев

Условие: Станислав Васильев

Недавно, на финале чемпионата мира в Токио, Билл Пучер спросил декана математико-механического факультета УрГУ Магаза Оразкимовича:

Ну как там уральские студенты, не устали еще катать вашу пирамиду?

Не устанут, она же совсем легкая, — ответил декан.

Ну да легкая, она же не пустая, а какой-то водой заполнена, — возмутился переводивший разговор Денис.

— А какого объема ваша пирамида? — спросил тогда Билл Пучер.

Сегодня вам предстоит ответить на вопрос Билла Пучера.

Обратите внимание, что пирамида декана на самом деле является подставкой для ручки, и имеет сквозное цилиндрическое отверстие. Если пирамида стоит на своем основании, то ось отверстия строго вертикальна.

Формат входного файла

В первой строке указаны два числа H и W — высота пирамиды и длина стороны основания (напомним, что пирамида декана — это правильная четырёхугольная пирамида). Во второй строке находятся два числа X и Y — положение центра отверстия (считаем, что оси направлены вдоль ребер основания и центр координат находится в центре основания пирамиды). В третьей строке указан радиус отверстия R . Известно, что отверстие для ручки не пересекает ребер пирамиды. ($0 < H, W < 10^4, |X| < W/2, |Y| < W/2, 0 < R < W/2$)

¹ «невозможно», яп.

Формат выходного файла

Выведите объем пирамиды с точностью до 10^{-3} . Толщиной стенок пренебречь.

Пример

input.txt	output.txt
3.0 3.0 -1.0 -0.7 0.1	8.96858

Задача J. Турнир по сумо

Автор: *Сергей Путьрев*

Подготовка: *Владимир Яковлев*

Условие: *Владимир Яковлев*

В Токио проходит турнир по сумо, в котором участвуют 2^N спортсменов. В каждом поединке обязательно есть победитель, а проигравший выбывает из турнира. Таким образом, чтобы определить победителя турнира, необходимо провести N раундов.

Организаторы хотят, чтобы в как можно большем числе раундов все поединки прошли между сумоистами из разных префектур Японии. Для этого они могут подтасовать результаты жеребьевки произвольным образом.

Формат входного файла

В первой строке записано число N ($1 \leq N \leq 10$). В каждой из следующих 2^N строк записаны имя сумоиста и префектура, которую он представляет. Имя и префектура являются последовательностями латинских букв длиной не более 30.

Формат выходного файла

Выведите максимальное количество раундов, в которых сумоисты из одной префектуры гарантированно не встретятся (то есть найдите наибольшее возможное число K , такое, что при какой-то начальной расстановке соперников, независимо от исхода поединков, по крайней мере в K раундах все поединки будет сыграны между сумоистами из разных префектур).

Пример

input.txt	output.txt
3 Homasho Ishikawa Tamakasuga Tokyo Futeno Tochigi Takekaze Tokyo Kasugao Yamaguchi Kotoshogiku Ishikawa Kotomitsuki Tokyo Miyabiyama Shizuoka	1

Решения задач XI чемпионата Урала по спортивному программированию

Задача А. Автодополнение

Данная задача может решаться различными способами, позволяющими оптимизировать поиск пужных слов. Жюри предлагает решение, основанное на боре (*trie*, в английской терминологии).

Бором называется дерево с выделенным корнем, в котором каждое ребро помечается буквой. Очевидно, что по пути от вершины до листа читается некоторое слово. Логично каждый лист пометить, этим читаемым словом.

Некоторый словарь определяет бор, в котором словами из данного словаря и только ими помечены листья.

А в этом боре легко искать требуемый в задаче набор из десяти наиболее популярных слов, начинающихся с одного префикса. Действительно, слова, начинающиеся с одного префикса, определяют поддереву, нашего бора. И найти 10 самых популярных слов в этом дереве можно, например, перебором, использующим очередь с приоритетами (для этого дополнительно в каждом узле будем хранить максимальную частоту слова среди всех слов, которые читаются через этот узел).

Задача В. Танцевальная революция

Задача является чисто технической — требуется реализовать все, что описано в условии.

Задача С. Одноклассники—3

Нетрудно доказывается, что оптимального времени переключения всех компьютеров на один тип протокола можно достичь, давая команды на переключение протоколов всего одному компьютеру. Далее поиском в ширину от выбранного компьютера легко узнать время, за которое произойдет смена протокола на всех компьютерах данной сети, если давать команды смены протокола только на нем. Осталось перебором выяснить, какой компьютер использовать для этой цели эффективнее всего. Сложность получившегося решения $O(N^3)$.

Задача D. Иероглифы

Задача настолько проста, что тут нечего рассказывать!

Задача Е. Сортировка по-японски

Вы наверняка догадались, что способ сортировки совпадает с тем, что используется в Проводнике Windows XP. Именно его правила и нужно было выяснить. Правила довольно просты. Мысленно

добавим в числа ведущие нули так, чтобы длины всех чисел в строках были одинаковыми (числами считаются максимальные подстроки, состоящие из цифр). Отсортируем эти строки лексикографически. Строки, которые равны по такому критерию, по тем не менее, различны, сравниваются особо: находим в строках первое несовпадение — это будет различие в количестве ведущих нулей, раньше идет та строка, у которой больше ведущих нулей. Показателем такой пример: 01w01, 01w1, 1w01, 1w1, 01w01a. Неправильно считать, что особое сравнение — обычное лексикографическое, вот контр-пример: z000, z00, z0. Кроме того, в этой задаче важен выбор алгоритма сортировки, наиболее подходящий — быстрая сортировка с рандомизацией.

Задача F. Перебор паролей

Сначала вычислим количество паролей, которое должен перебирать каждый компьютер. После этого останется только реализовать функцию, которая возвращает пароль с конкретным номером по порядку. Для этого решения потребуется реализация длинная арифметика.

Задача G. Сакура и статистика

Решение этой задачи описывается в работе D. Franzblau, D. Kleitman, "An algorithm for constructing regions with rectangles: Independence and minimum generating sets for collections of intervals", опубликованной в 1984 году.

Задача H. Японский пазл

Легко убедиться, что поворотов в замкнутом пути четное число (смена с вертикали на горизонталь происходит четное число раз). Общее число кусочков тоже четно (пусть есть шахматная раскраска, тогда каждый участок пути переводит из черного в белый, в конце приходим в начало, значит всего четное число смен цвета.) Если есть хотя бы два прямых куса и четыре уголка, то можно составить любой путь, где тех и тех четное число. Например, так: сначала строим квадратик LLLL, потом две стороны расширяем всеми оставшимися буквами F (LF...FLLF...FL), дальше добавляем по два уголка следующим образом: последний FL заменяем на RLLF, потом RLLF можно заменить на RLFLRL, потом так же поступаем с тем LF, который в середине последней замены. Когда нет двух прямых кусков, становится посложнее: можно построить квадратик (четыре уголка LLLL), потом можно построить крестик из 12 уголков (LRLRLRLRLRL) и потом добавлять по четыре уголка заменяя, например, последнюю LRL на RLLRLRL и так же поступая со средней LRL в получившейся комбинации. Не кратные четырем комбинации из уголков построить нельзя (доказательство чуть посложнее, здесь мы его описывать не будем), восемь уголков без прямых тоже не построить, если меньше четырех уголков вообще ничего не построить независимо от числа прямых. Таким образом, все возможные случаи разобраны.

Задача I. Пирамида декана—3

Требуется вычислить объем пирамиды и вычесть из него объем цилиндра со скошенным верхним основанием. Нетрудно заметить, что объем такого цилиндра равен объему обычного цилиндра с тем же радиусом основания и с такой же высотой оси. Таким образом, вся сложность задачи заключается в нахождении длины оси цилиндра.

Задача J. Турнир по сумо

Назовем раунд, в котором встретились сумоисты из одной префектуры, плохим. Пусть самое большое количество борцов из одной префектуры равно M , и число k таково, что $2k < M \leq 2k + 1$. Тогда при любой жеребьевке, по крайней мере, $k + 1$ раундов могут быть плохими. Действительно, если, например, эти сумоисты сильнее всех остальных, они должны будут бороться друг с другом. В одном плохом раунде из них будут исключено не более половины, значит, чтобы в конце остался всего один, плохих раундов должно быть больше, чем k . Если же сделать такую начальную расстановку, чтобы, например, в первых $k + 1$ раундах всегда (по возможности) боролись вместе сумоисты из одной префектуры, то даже если бы в начале их было $2k + 1$, то к $k + 2$ раунду в турнире останется не более одного борца из каждой префектуры, значит, в оставшихся раундах будут встречаться уже только сумоисты из разных префектур. Ответ: $N - k - 1$, где k указано выше.

Статистика решения задач на XI чемпионате Урала по спортивному программированию

Статистика	A	B	C	D	E	F	G	H	I	J
Пытавшихся команд	22	7	11	45	20	20	2	12	39	43
Решивших команд	8%	6%	6%	100%	8%	35%	0%	4%	75%	75%
Зачтено решений	4	3	3	45	4	16	0	2	34	34
Не зачтено	103	19	18	7	83	18	6	31	42	57
Процент успешных попыток	3%	13%	14%	86%	4%	47%	0%	6%	44%	37%
Довели задачу до конца	18%	42%	27%	100%	20%	80%	0%	16%	87%	79%

Содержание

Программа четвертьфинала Командного студенческого чемпионата мира по программированию ACM-ICPC 2008	5
Рейтинг вузов по итогам соревнований по программированию в 2004/05-2006/07 годах	6
Команды Северо-Восточного региона в финалах чемпионатов мира	11
Университеты Урала в полуфиналах чемпионатов мира	12
Университеты Урала в финалах чемпионатов мира	13
Задачи XI чемпионата Урала по спортивному программированию	15
Автодополнение	15
Танцевальная революция	16
Одноклассники-3	18
Иероглифы	19
Сортировка по-японски	20
Перебор паролей	21
Сакура и статистика	21
Японский пазл	22
Пирамида декана-3	23
Турнир по сумо	24
Решения задач XI чемпионата Урала по спортивному программированию	25
Статистика решения задач на XI чемпионате Урала по спортивному программированию	28

