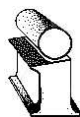




Московский авиационный институт  
(государственный технический  
университет)



Федеральная целевая программа  
«Интеграция»

**Microsoft®**

Корпорация Microsoft

# Восемнадцатая личная олимпиада МАИ по программированию

14 мая 2006 года

## Правила проведения

### Основные положения

Данные правила аналогичны правилам чемпионата мира по программированию, проводящегося ACM.

Длительность олимпиады 5 часов. Первенство сугубо личное. Допускается работать под выделенным *username* на указанной для олимпиады машине. Запрещается открывать более, чем одну рабочую сессию. Результаты данной олимпиады учитываются на экзамене по курсу информатики, поэтому запрещается:

- Использование любой литературы.
- Общение с другими участниками олимпиады включая передачу сообщений по электронной почте.

Ответы на вопросы по задачам следует искать в их формулировках, то есть никаких комментариев со стороны жюри по условиям задачи не будет за исключением особых случаев.

Со всеми проблемами технического характера обращаться к организаторам олимпиады, которые будут находиться в каждом зале.

Нарушение правил проведения олимпиады категорически нежелательно.

### Работа с системой проверки задач

Решение оформляется в виде исходного текста программы, полностью содержащегося в одном файле.

На проверку посылается исходный текст программы, на проверочном сервере он компилируется и тестируется.

Для того, что бы послать решение задачи на проверку, необходимо выполнить команду зайти на сайт `dalian.umc8.mai.ru` при помощи браузера *links* с терминала или любого другого браузера с иных систем. Следовать инструкциям на сайте для отправки решений и получения информации о корректности решения, положения участников и состоянии турнира.

Варианты решений жюри:

- `accepted`: решение засчитано.
- `compilation error`: ошибка при компиляции.
- `runtime error`: программа не смогла корректно завершить свою работу.
- `wrong answer`: неправильный ответ.
- `time limit exceeded`: решение не укладывается в ограничения по времени.
- `presentation error`: программа использует неверный формат вывода.

Призеры олимпиады будут представлены к награждению грамотами и ценными подарками корпорации Microsoft (<http://www.microsoft.com/ru>).

## Задача А. Counterstrike

**Ограничения по времени: 1 с**

Перед битвой под Дьен-Бьен-Фу французские артиллеристы заявили, что смогут без особых проблем контрбатарейным огнем подавить артиллерию Зиапа. Для этого на передовую были отправлены наблюдатели. Каждый у каждого из них есть рация, через которую он сообщает направление, в котором был слышал выстрел. Таким образом, два наблюдателя могут обеспечить информацией свою батарею для ведения ответного огня.

### Формат входного файла

Первые две строки входного файла содержат координаты наблюдателей — пары целых чисел. Остальные строки содержат пеленг<sup>1</sup>, переданный первым и вторым наблюдателем соответственно.

### Формат выходного файла

Необходимо вывести целочисленные координаты орудий противника. Если сделать это невозможно, то необходимо вывести слово ERROR.

### Пример

stdin	stdout
-100 0	0 100
100 0	0 173
45 315	
30 330	

## Задача В. Cypher

**Ограничения по времени: 20 с**

До определенного момента вьетнамцы могли свободно открытым текстом передавать сведения не опасаясь, что их поймут американцы. Однако число владеющих вьетнамским языком среди американцев неуклонно росло, поэтому вопрос о секретности был поднят на самом высоком уровне.

Отныне, чтобы передавать сообщения, вьетнамцы разбивают их на две части. Первая часть представляет собой шифр, в котором каждому знаку, будь то буква, знак препинания или пробел, сопоставляется последовательность из таких же букв длиной до 50 знаков. Вторая часть представляет собой собственно сообщение, в котором каждая буква заменена кодом, и для усиления между кодовыми последовательностями случайно добавлены буквы так, что ни одна из таких случайных последовательностей не является подпоследовательностью кода. Обратное также верно.

У вас в руках оказались обе части сообщения, и вам надо его расшифровать.

### Формат входного файла

Первая строка входного файла содержит число  $N$  — количество закодированных букв,  $0 < N \leq 96$ . Следующие  $N$  строк содержат описание кода. Первым в строке идет знак сообщения, после него через пробел записан шифр. После кодов идет строка с сообщением, которое надлежит расшифровать. Длина этого сообщения не превышает 10 000 000 символов.

### Формат выходного файла

Необходимо вывести одну строку — расшифрованное сообщение.

<sup>1</sup>Угол между направлением на объект и направлением на север, отсчитывается по часовой стрелке.

### Пример

stdin	stdout
6 C y+a e gBS h CRO p 3T{ r McR y .UH r1S\$Ny+at.UH603T{CROV~I[gBSg+/McRfWXfj	Cypher

### Задача C. The Game

**Ограничения по времени: 1 с**

В перерыве между заданиями все развлекаются как могут. В последнее время особой популярностью среди обитателей базы Нуи-Пек пользуется игра на основе системы D&D, где игровой процесс в основном определяется кубиками с различным числом граней. Кубики обозначаются  $dn$ , где  $n$  — число граней. На каждую грань нанесены числа от 1 до  $n$ .

Старшину Крокера заинтересовал такой факт: как часто ему удастся убить монстра и не получить при этом никаких повреждений? При встрече с монстром игрок выполняет оглушающий удар, для чего и игрок, и монстр выбрасывают по кубику  $d8$ . Если игрок пробил защиту монстра (т. е. у него очков больше, чем у монстра), то удар считается успешным и монстр теряет возможность наносить ответные удары в течение трех следующих ходов.

Далее игрок наносит обычные удары. Уровень атаки игрока определяется кубиком  $d20$ , защиты монстра — кубиком  $d8$ . Удар происходит, если игрок пробил защиту монстра. Для определения урона монстру используется кубик  $d10$ . Урон вычитается из здоровья монстра. Монстр гибнет, как только уровень его здоровья перестанет быть положительным.

Удар называется критическим, если атака игрока превышает 17. В этом случае последующий урон утраивается.

#### Формат входного файла

В единственной строке входного файла содержится здоровье монстра.

#### Формат выходного файла

Необходимо вывести вероятность уничтожения игроком монстра с точностью до 2 знаков.

### Пример

stdin	stdout
10	0.31

### Задача D. Safe Track

**Ограничения по времени: 1 с**

Группе сержанта Стива Р. Хокинса поставлена боевая задача: найти и уничтожить склад вьетконговцев. Но перед Стивом стоит одна проблема: до склада необходимо дойти. Путь лежит через джунгли, полные разных опасностей: змей, ловушек, мин и собственно вьетконговцев.

Но Стив является опытным разведчиком. Поэтому у него в планшете спрятана карта, разбитая на квадраты для удобства целеуказания. В каждом квадрате Стив на основании разведанных написал число от 0 до 10, которое отражает степень опасности данного квадрата. Всего на карте  $W$  квадратов по

горизонтали и  $H$  по вертикали, причем  $0 \leq W, H \leq 1000$ .

Сейчас группа находится в квадрате  $(1, 1)$ , а цель — в квадрате  $(W, H)$ . Вам необходимо указать самый безопасный путь к цели. Но поскольку никто не хочет оставаться в кишачих веткиговцах джунглях, то этот путь одновременно должен быть самым коротким. Это требование даже более важно, поскольку проще обезвредить десяток ловушек, чем такое же число веткиговцев.

### Формат входного файла

Первая строка входного файла содержит числа  $W$  и  $H$ . На остальных  $H$  строках расположено по  $W$  целых чисел.

### Формат выходного файла

Необходимо вывести путь в виде пар координат по паре на каждой строке. Если есть несколько путей, выбрать из них самый северный.

### Пример

stdin	stdout
2 2	1 1
1 2	2 1
2 3	2 2