

**ОЛИМПИАДА ШКОЛЬНИКОВ «Робофест» по ИНФОРМАТИКЕ**  
**МАТЕРИАЛЫ ЗАДАНИЙ 2015/16 учебного года**

В 2015/16 учебном году отборочный этап олимпиады «Робофест» по информатике проходил в форме очных робототехнических соревнований, задания которых представляли собой инженерные, конструкторские и робототехнические задачи. Оценка участникам выставлялась по результатам соревнований и результатам собеседований участников с экспертами – членами жюри олимпиады.

На заключительном этапе участники получали оценки по результатам робототехнических соревнований (I тур) и по результатам выполнения заданий теоретического (II-го) тура. По каждому туру жюри выставляло технические баллы: по I туру – в соответствии с техническим регламентом соревнования, по II – туру – в соответствии с критериями методической комиссии. Оценкой за каждый тур являлась сумма технических баллов, нормированная таким образом, что максимальная итоговая оценка составляла 50 баллов. Максимальная оценка участника за финальный этап составляла 100 баллов.

Во время выполнения заданий II-го тура каждому участнику было предоставлен персональный компьютер с необходимым программным обеспечением для разработки и запуска программ. Кроме того, участнику была доступна автоматизированная проверяющая система ejudge, зарекомендовавшая себя во множестве других олимпиад по информатике и программированию. Участники могли отправлять свои решения во время тура и получать отчет о правильности своей программы. Количество попыток отправки программ не ограничивалось.

На II-м туре участникам были предложены две задачи, разделенные на несколько этапов. Участник мог выполнять задания в произвольном порядке.

Для написания программ на II-м туре участник мог использовать следующие языки и среды программирования и компиляторы для них: Pascal ABC.NET (Mono) v1.8.0.513, Borland Delphi 6 (Kylix) 14.5, Java JDK 7 1.7.0\_80, Java JDK 1.8.0\_74, Mono C# 2.10.8.0, Mono Visual Basic .NET 0.0.0.5943, Perl 5.16.3, Ruby 2.0.0p353, PHP 5.5.20, Python 2.7.5, Python3 3.3.2, Python (PyPy) 2.7.3, Python3 (PyPy) 3.2.5, The Glasgow Haskell Compiler 7.4.2, Free Basic 0.90.1, clang C 3.7.0, clang C++ 3.7.0, GNU go 5.3.0, GNU C (32 bit) 5.3.0, GNU C++ (32 bit) 5.3.0, Free Pascal (32 bit) 2.6.2, Free Basic (32 bit) 0.90.1, clang C (32 bit) 3.7.0, clang C++ (32 bit) 3.7.0, Kумир, Kумир2.

**ЗАКЛЮЧИТЕЛЬНЫЙ (ФИНАЛЬНЫЙ) ЭТАП 2015/16 года**  
**II (ТЕОРЕТИЧЕСКИЙ) ТУР**

**БИЛЕТ № 01 (10-11 классы)**

**Задача А. Выйти из лабиринта-1**

**Формулировка задания**

Всем известно, что одно из самых частых заданий на соревнованиях для роботов - выбраться из лабиринта. Мы просим Вас написать упрощенную модель искусственного интеллекта робота для решения данной задачи. Представим поле, по которому движется робот, в виде таблицы. В каждый момент времени он может переместиться в одну из соседних по стороне клеток, если эта клетка ничем не занята. Добраться роботу нужно в клетку, обозначенную как выход.

По результатам проверки последовательности ходов, которые приводят в клетку с выходом, будет оценена её длина. Таким образом добраться до выхода требуется по возможности наименьшим числом ходов.

#### Формат входных данных

На вход программе вводятся два числа  $h$  и  $w$  ( $1 \leq w, h \leq 100$ ) — соответственно количество строк и столбцов в поле.

Затем в  $h$  строках по  $w$  символов вводится поле. Каждый символ означает: # - клетка со стеной, . - свободная клетка, @ - в этой клетке находится робот, E - выход.

#### Формат выходных данных

Вывести требуется последовательность команд робота, где каждый символ означает: R - переместиться на одну клетку вправо, L - влево, U - вверх, D - вниз, S - остаться на месте. Длина последовательности ходов не должна превышать 10000 символов.

#### Примеры

тест	ответ
3 3 ... .#. @#E	UURRDD

#### Решение

Решение может быть получено следующим алгоритмом:

- 1) Надо сопоставить каждой пустой клетке натуральное число, обозначающее минимальное количество ходов, необходимое, чтобы добраться в эту клетку.
- 2) Надо пройти обратно от выхода к входу, каждый раз выбирая клетку с числом, на единицу меньшим числа в текущей клетке.

Расстановку чисел по пустым клеткам можно сделать так. Сопоставляем ближайшим клеткам к начальному расположению робота число 1, а далее сопоставляем каждой очередной клетке, до которой можно добраться за 1 ход от текущей клетки, число, на единицу большее числа, сопоставленного текущей клетке.

#### Критерии оценивания

Максимальная оценка за решение задачи: 20 технических баллов.

### Задача В. Выйти из лабиринта-2

#### Формулировка задания

Как в предыдущей задаче роботу нужно выйти из лабиринта. В лабиринте, кроме стен, установлены автоматические двери, которые обычно недоступны для перемещения, но раз в  $t$  хода открываются на 2 хода. Робот не должен оставаться в клетке с дверью в тот момент, когда она закрывается.

Считайте, что все двери открылись первый раз в момент первого хода.

#### Формат входных данных

В отличие от предыдущей задачи в первой строке находятся три числа  $h$ ,  $w$ ,  $t$  ( $1 \leq t \leq 10$ ) — соответственно количество строк, столбцов в поле и время между открытиями дверей. При вводе поля добавляются символы: С - закрытая дверь, О - открытая дверь.

#### Примеры

тест	ответ
3 3 4 .О. .#. @#E	UUSSRRDD

#### Комментарий к примеру

В примере за два хода движения вверх дверь закрылась, после этого робот подождал два хода, пока она не откроется.

#### Решение

Сначала пытаемся найти путь, не проходящий через двери, так же, как и в предыдущей задаче. Если это удалось сделать, используем его для ответа. Если такого пути нет, строим какой-либо путь, проходящий через двери. Далее рассчитываем, в какие моменты времени

робот окажется перед закрытой дверью, и вставляем в эти моменты нужное количество команд ожидания.

### Критерии оценивания

Максимальная оценка за решение задачи: 20 технических баллов.

### Задача С. Выйти из лабиринта-3

#### Формулировка задания

Как в предыдущей задаче, роботу нужно выйти из лабиринта. Но всей карты у робота нет, но есть камера, позволяет видеть поле на расстоянии меньше  $k$  клеток от робота по каждой из координат.

Чтобы смоделировать такую ситуацию, после каждого выведенного хода ваша программа сможет считать поле  $6 \times 6$  клеток, которую видит робот. Взаимодействие заканчивается, когда робот достигает выхода, либо совершает недопустимый ход.

#### Формат входных данных

На вход программе вводятся четыре числа  $h, w, t, k$  ( $1 \leq k \leq 6$ ).

Затем в  $2k + 1$  строках по  $2k + 1$  символов вводится видимое поле.

После каждого хода вводится новое видимое поле.

Все клетки за границами поля являются стенам.

#### Примеры

тест	ответ
<pre> 3 3 4 1 #.# #@# ### ----- #.# #@# #.# ----- ### #@C #.# ----- ### #@C #.# ----- ### #@O #.# ----- ### .#. .#. ----- ### OC# #.# ----- O.# #@# #E# </pre>	<pre> UUSSRRDD </pre>

#### Комментарий к примеру

В примере за два хода движения вверх дверь закрылась, после этого робот подождал два хода, пока она не откроется. Пунктирная черта означает ожидание ввода команды.

### Решение

В отличие от предыдущих задач робот не может применить алгоритм, описанный в задаче В, для всего лабиринта. Тем не менее, он может применить его для видимой части. Если в видимой части есть клетка выхода, применяем алгоритм, описанный в задаче В. Если в видимой части нет клетки выхода и из видимой части невозможно выбраться (вокруг только стены), запоминаем эту часть лабиринта и возвращаемся назад. Если в видимой части нет клетки выхода, но есть граница видимой части, не являющаяся стеной, применяем алгоритм из задачи В и выбираем ход в направлении выхода, запоминая видимую часть лабиринта. Если этот ход уже был использован, возвращаемся на шаг назад и пробуем другой ход. Так действуем до тех пор, пока нам не дадут видимую часть, в которой будет выход.

Можно поступить проще и сначала «прочитать» весь лабиринт, обходя его целиком и запоминая видимые части. Однако такое решение не приведет к выходу по возможности наименьшим числом ходов, как это требовалось в задании.

### Критерии оценивания

Максимальная оценка за решение задачи: 15 технических баллов.

## Задача D. Выйти из лабиринта-4

### Формулировка задания

Как в предыдущей задаче роботу нужно выйти из лабиринта. Но в этой задаче камера робота не может видеть за стенами. Клетка считается видимой, если какой-то из отрезков, соединяющих углы клетки, в которой находится робот, и углы заданной клетки, не проходит через занятые стеной или дверью клетки.

### Формат входных данных

Среди возможных символов, описывающих поле, добавляется \* - клетка, которую робот не видит.

### Примеры

тест	ответ
<pre> 3 3 4 2 *#.O* *#.##* *#@##* *####* ***** ----- *##### *#.O. *#@##* *#.##* *####* ----- ***** *##### *#@C* *#.##* *#.##* ----- И Т. Д. </pre>	<pre> UUSSRRDD </pre>

### Решение

Может быть построено аналогично решению предыдущей задачи.

### Критерии оценивания

Максимальная оценка за решение задачи: 15 технических баллов.

## Задача E. Посадить марсоход-1

### Формулировка задания

В этой задаче вам необходимо "приземлить" спускаемый аппарат с марсоходом на поверхность. Представим поверхность планеты как *себя связанное полимино* (плоская геометрическая фигура, образованная путём соединения нескольких одноклеточных квадратов по их сторонам). Представим марсоход в виде единичного квадратика, который требуется приземлить на ровную горизонтальную поверхность, состоящую минимум из 3 клеток. Таким образом, в точке посадки высота соседних клеток должна быть одинаковой. У марсохода есть горизонтальная и вертикальная скорость  $v_x, v_y$  метров в секунду, а на Марсе действует ускорение свободного падения  $g$ .

Для успешной посадки в момент приземления необходимо, чтобы скорости были небольшими, т.е.  $v_x^2 + v_y^2 \leq b$ .

У спускаемого аппарата есть запас топлива  $U$ , который может расходоваться для изменения каждой компоненты скорости. Причем на изменение единицы скорости тратится одна единица топлива.

Для простоты будем считать, что все скорости и ускорения целые, начальная скорость спускаемого аппарата нулевая. Считаем, что каждую секунду скорость меняется на заданное ускорение, затем происходит перемещение аппарата в соответствии со скоростью.

### Формат входных данных

На вход программы в первой строке подаются целые числа  $N, x, y$  ( $3 \leq N \leq 1000, 1 \leq x \leq N, 1 \leq y \leq 1000$ ), длина рассматриваемого отрезка поверхности планеты и начальная координата спускаемого аппарата.

Во второй строке находятся целые числа  $g$  - ускорение свободного падения,  $b$  - ограничение на скорость посадки,  $U$  ( $0 \leq U \leq 100000$ ) запас топлива.

В третьей строке находятся  $N$  чисел  $h_i$  - высота поверхности в точке  $i$ .

### Формат выходных данных

Для каждой секунды полета требуется вывести 4 числа - сколько топлива тратится для изменения скорости влево, вправо, вверх, вниз, соответственно.

Если приземлить аппарат невозможно, то выведите одну строку -1 -1 -1 -1.

### Примеры

тест	ответ	пояснение
3 2 10	0 0 0 0	свободное падение из точки (2, 10) в точку (2, 6)
4 1 100	0 0 5 0	перемещение из (2, 6) в (2, 3)
1 1 1	0 0 6 0	перемещение из (2, 3) в (2, 2) - приземление

### Решение

Решение может быть получено, например, вычислением условной параболы, начало которой находится в начальной координате, а конец - на горизонтальной площадке.

### Критерии оценивания

Максимальная оценка за решение задачи: 20 технических баллов.

### Задача F. Посадить марсоход-2

#### Формулировка задания

В этой задаче вам необходимо "приземлить" спускаемый аппарат с марсоходом на поверхность. В отличие от предыдущей задачи, аппарат не имеет информации обо всем рельефе поверхности планеты, а получает в каждый момент времени информацию о свободных и занятых окружающих его клетках.

#### Формат входных данных

В начале работы решения вводятся нечетное целое  $n$  - размер квадрата, который "видит аппарат", целое число  $g$  - ускорение свободного падения,  $b$  - ограничение на скорость посадки,  $U$  ( $0 \leq U \leq 100000$ ) запас топлива.

Затем вводится таблица  $n \times n$  с информацией о текущем положении корабля со следующими обозначениями: M - спускаемый аппарат, \* - поверхность планеты, . - свободное пространство.

После вывода каждой команды на изменение скорости таблица с информацией о текущем положении вводится повторно.

### Примечание

Гарантируется, что аппарат посадить с текущим запасом топлива всегда возможно.

### Примеры

тест	ответ
5 1 1 100 .....* .....* ..М.* .....* .....* -----	0 0 0 0
.....* .....* ..М.* .....* ***.* -----	1 0 2 0
.....* .....* .М..* .....* ***.* -----	0 1 0 0
.....* .....* .М..* ***.* *****	

### Решение

Решение может быть построено аналогично предыдущей задаче с учетом видимой части пространства.

### Критерии оценивания

Максимальная оценка за решение задачи: 10 технических баллов.

Максимальная сумма технических баллов за работу на теоретическом туре: 100.