



## VII Поволжская открытая олимпиада по информационным технологиям «Волга ИТ – 2014»

### Номинация «Прикладное программирование»

#### Задание финального этапа

#### Первый этап

#### Введение

Давным-давно, за морем, за океаном, в тридесятом царстве, в тридевятом государстве было озеро широкое-преширокое. Посреди того озера был остров, а на том острове гора высокая-превысокая. На вершине горы жил огнедышащий дракон старый-престарый и очень страшный. Берег дракон принцессу самую прекрасную невиданной красоты.



Неподалеку от острова располагался замок, в котором жил молодой принц, сильный и мужественный, желавший спасти прекрасную принцессу из заточения.

В день своего двадцатипятилетия, принц взял отцовский меч, одел рыцарский доспех, сел на лодку и отправился к острову посереде озера.



Добравшись до земли, принц зашел в глубокое подземелье, где ему предстояло пройти запутанную систему лестниц и коридоров, столкнуться с невиданными существами и повергнуть их в честном бою. К счастью принц обнаружил на стене карту подземелья той горы. Разведя костер, он

принялся искать подходящий маршрут. Помогите принцу спасти принцессу от страшного дракона и вернуться домой как можно скорее!



### **Постановка задачи**

Лабиринт имеет несколько уровней. Каждый уровень представляет из себя двумерную матрицу, состоящую из проходимых для принца клеток. Некоторые клетки могут быть разделены стенами. Сквозь стены проходить нельзя и те не занимают пространства. Уровни соединены между собой одной или несколькими двусторонними лестницами (по ним можно как спускаться, так и подниматься). Переход из одной клетки в другую по вертикали или горизонтали, спуск или подъем по лестнице занимает 1 ход. Так же в лабиринте заданы начальное положение принца и положение принцессы. Необходимо найти длину кратчайшего пути принца к принцессе и вывести ее на экран и в файл. Гарантируется наличие по меньшей мере одного такого пути.

На программу наложено ограничение по количеству потребляемой памяти в 1 ГБ и времени исполнения в 10 секунд.

### **Графический интерфейс**

Приложение должно отвечать следующим требованиям:

- Отображать лабиринт одним из двух способов:
  - Все уровни на экране.
  - Только один уровень на экране одновременно с возможностью переключения его на уровень выше или ниже.
- Иметь кнопку загрузки лабиринта из файла с возможностью ввода имени этого файла.
- Иметь кнопку расчета кратчайшего пути и вывода его на экран и в файл result.txt.
- Иметь возможность создавать и редактировать лабиринты, т.е.
  - Иметь функцию создания пустого лабиринта заданного пользователем размером (ширина, высота, количество уровней).
  - Иметь панель выбора текущей кисти — стена (слева, справа, сверху, снизу, со всех сторон), лестница (вверх, вниз) стартовая точка, конечная точка, и стерка. При нажатии на лабиринте левой кнопкой мыши в клетке, на которую наведен курсор, появляется выбранный в панели элемент. В случае рисования более одной стартовой или конечной точки, прежняя соответствующая ей точка исчезает.

### **Входные данные**

Файл имеет кодировку ASCII.

Первая строка содержит три числа — ширина ( $3 \leq W \leq 10$ ), высота ( $3 \leq H \leq 10$ ) и количество уровней ( $1 \leq D \leq 5$ ) в лабиринте. Далее в  $D$  секциях, состоящих из  $H$  строк и  $W$  столбцов, идет описание самого лабиринта. Каждая секция

описывает уровень, начиная с первого. Каждый символ ячейки может принимать значения:

«(»	Стена слева
«)»	Стена справа
«/»	Стена сверху
«\»	Стена снизу
«>»	Спуск на уровень ниже
«<»	Подъем на уровень выше
«#»	Стена со всех четырех сторон
«@»	Отправная точка принца
«&»	Местоположение принцессы
Любой другой печатный символ — проходимая клетка без специфичных свойств	

Стены и лестницы могут быть описаны как в одну сторону, так и в обе и в обоих случаях означают одно и то же — наличие стен и лестниц в обе стороны. Пустые строки в файле должны игнорироваться.

### Выходные данные

Файл result.txt должен иметь кодировку ASCII и содержать одно целое число — длину кратчайшего пути.

В случае, если лимит времени исчерпан, алгоритм должен прекратить свое выполнение и вывести в выходной файл значение 0.

Разрешается ставить лишние пробелы в конце строк и пустые строки в конце файла.

### Пример:

5 5 2

@....

////.

.#./.

.####

....>

&....

////.

.#///

.\(((

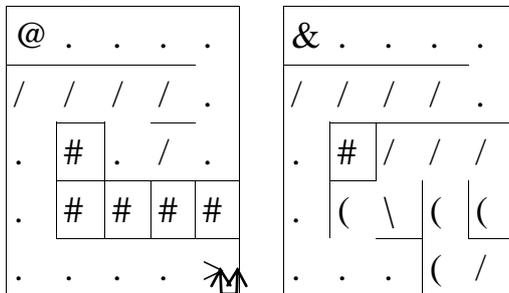
...(/

**Ответ:**

37

**Примечание:**

Лабиринт из примера имеет следующий вид:



### Дополнительные условия

Программа предоставляется в виде исполняемого файлов, исходных текстов, файлов проекта для среды разработки и readme.txt.

В файл readme.txt включите:

- Краткое описание алгоритма
- Инструкцию по сборке программы из исходных текстов (дополнительные условия, настройки среды - постарайтесь свести их к минимуму)
- Требования для запуска исполняемого файла (если не запустится, мы попробуем собрать заново из исходных текстов, но это минус)

Принимаются частичные решения: GUI реализован частично или отсутствует, отсутствие ограничения на время работы алгоритма, и пр. В таком случае в файле readme.txt необходимо указать какие ограничения наложены на программу.

### Оценка

Итоговая оценка будет складываться из результатов прогона на тестовых задачах (оценивается степень близости ответа программы к ожидаемому значению), скорости выполнения, результатов ручного тестирования GUI и, возможно, анализа исходных текстов программы.

### Баллы

Алгоритмическая часть — 200 баллов.

Графический интерфейс — 100 баллов.

## Второй этап

### Введение

Монстры в лабиринте оказались совсем не просты! Они бьют больно и вскоре раны дают о себе знать. Самое время принцу присмотреться к своим соперникам — может быть с некоторыми из них удастся избежать боя!



### Требования

Второй и следующие этапы проверяются только при полной реализации всех задач первого этапа.

### Постановка задачи

В лабиринте присутствуют противники, с которыми необходимо сражаться. У принца и его врагов есть два численных параметра — сила и количество жизней. Главный герой имеет силу 10 и 100 жизней.

### Противники:

Название	Сила	Жизни	Идентификатор
Скелет	12	28	S
Зомби	7	39	Z
Некромант	14	58	N
Дракон	21	116	D

### Битвы:

Каждый ход (это стоит учитывать при поиске кратчайшего маршрута) принц и его противник наносят друг по другу удар, отнимая оппоненту количество жизней, равное силе своего персонажа. Когда у одного из участников битвы жизней становится 0 и меньше, битва заканчивается. В случае поражения принца его дальнейшее путешествие невозможно. Начавшуюся битву нельзя прервать. Жизни, оставшиеся у главного героя на конец боя, переносятся на следующий и не восстанавливаются.

Например, принц, ввязавшись в битву с 10 силой и 100 жизнями, убьет зомби за 5 ходов (столько раз надо вычитать 10 из 42, что бы получить значение, меньшее или равное 0) и останется с 55 жизнями ( $100 - 9 * 5$ ).

## Основные требования к графическому интерфейсу:

Возможность ставить на карту противников.

## Дополнительные требования к графическому интерфейсу:

- Сделайте красиво! Замените цвета и буквы на картинки, придайте стиля окну, кнопкам и другим элементам управления.
  - Добавьте игровой режим:
    - Для входа в игровой режим необходимо нажать кнопку «Игра».
- Повторное нажатие кнопки начинает игру заново.
- Пользователь может управлять героем, используя стрелки для перемещения и пробел для пропуска хода.
  - При нападении на существо показывается подсказка с текущими силой и жизнями принца и его противника. Нажатие на пробел завершает один ход битвы.
  - Для перехода обратно в режим редактирования необходимо нажать кнопку «Редактировать».
  - Наличие звуковых эффектов, фоновой музыки и анимации приветствуется.

## Внимание!

Рекомендуем сохранить приложение с первого этапа в отдельном каталоге (как исходники, так и исполняемые файлы).

## Входные данные

В файле значение ячейки так же может быть одним из значений в таблице в столбце «Идентификатор». Всего на карте не более 10 монстров.

## Выходные данные

Прежние.

## Пример:

5 5 2

@....

////.

.#D/.

.####

....>

&....

////S

.#///

.\(((

...( /

## Ответ:

40

## **Баллы**

Алгоритмическая часть — 200 баллов.

Основные требования к графическому интерфейсу — 10 баллов.

Дополнительные требования к графическому интерфейсу:

- Дизайн — 80 баллов.
- Игровой режим — 170 баллов.

## Третий этап

### Введение



Бывший склеп — место, где оказался принц. Сотни лет здесь хоронили королей, а сейчас — пристанище личей, вампиров, некромантов и их творений. Ужас и страх закрадываются в душу, когда во мгле темных комнат вспыхивают глаза этих мерзких созданий.



Затаив дыхание, принц тихо крадется по извилистым коридорам лабиринта. Любой шум, будь то писк промчавшейся мимо летучей мыши или взявшийся из неоткуда гул сухого ветра, вынуждает героя останавливаться и подолгу всматриваться в окружающую его кромешную тьму.

Какая судьба уготована здесь принцу? Что ждет его в этом путешествии? Какие преграды ему предстоит преодолеть? Будет еще много вопросов, на которые нам предстоит ответить.



### Постановка задачи

Этап состоит из двух заданий. Можно выполнить как одно любое задание, так и оба одновременно. Реализации обоих заданий по отдельности не принимаются.

#### Задание 1:

В лабиринте появилось оружие и баночки с лечебными зельями.

#### Оружие:

Оружие устанавливает атаку (не увеличивает), равную той, что указана в таблице. Смена оружия занимает один ход. Прежнее оружие выкидывается и больше не может быть использовано.

Название	Дает силы	Идентификатор
Топор орков	15	a
Молот Бездны	20	m
Огненный меч	25	f

#### Баночки лечения:

Баночки с лечебным снадобьем восстанавливают здоровье (в том числе больше

100). Выпивание содержимого занимает один ход.

Название	Восстанавливает здоровья	Идентификатор
Зелье	40	р
Эликсир	80	е

### **Пояснение:**

Подбирать предметы не обязательно и можно пройти мимо них, в то время как битва с монстрами необходима, если принц оказывается в одной клетке с одним из них.

### **Задание 2:**

В лабиринте присутствуют два новых типа монстров, способных передвигаться из клетки в клетку.

Название	Сила	Жизни	Идентификатор	Движение
Лич	19	46	L	направо
Вампир	13	63	V	вниз

Каждое из перечисленных созданий начинает движение из своей начальной точки в направлении, указанном в таблице, доходит до стены, разворачивается и идет в обратном направлении, и так по кругу. Принц не может без битвы попасть на клетку, в которой либо находился противник на начало хода, либо будет находится на конец хода. Битва начинается в клетке, в которую ступит принц. Для совершения тактических маневров главный герой может пропускать ходы. Так как нежить любит тишину и спокойствие, их пути (все клетки, которые проходит передвигающийся монстр) не пересекаются и не соприкасаются по вертикали или горизонтали с другими движущимися и недвижимыми монстрами и их путями.

### **Основные требования к графическому интерфейсу:**

Возможность ставить на карту предметы и новых противников.

### **Дополнительные требования к графическому интерфейсу:**

В игровой режим добавляются предметы и движущиеся монстры. Для поднятия предметов используется клавиша «Enter». Запрещается ставить движущихся монстров так, что бы их пути пересекались или соприкасались с другими монстрами (должна выводиться ошибка).

### **Входные данные**

На карте присутствуют личи, вампиры и предметы. Всего на карте не более 10 монстров и предметов суммарно.

### Выходные данные

Прежние.

#### Пример для задания 1:

5 5 2

@....

////.

.#f/e

.####

....>

&....

////D

.#///

.\(((

...(/

#### Ответ:

46

#### Пример для задания 2:

3 3 1

##@

.L.

##&

#### Ответ:

4

#### Примечание:

В примере необходимо подождать два хода пока лич не пойдет в другую сторону, после чего начать движение к конечной точке.

### Баллы

Алгоритмическая часть:

- Задание 1 — 200 баллов.
- Задание 2 — 200 баллов.
- Оба задания — 400 баллов.

Основные требования к графическому интерфейсу — 10 баллов.

Дополнительные требования к графическому интерфейсу:

- Дизайн — 20 баллов.
- Игровой режим — 60 баллов.

## Алгоритмы

### Волновой алгоритм (алгоритм Ли)

Волновой алгоритм позволяет построить путь между двумя элементами в лабиринте. Сам процесс можно разделить на три этапа:

1. Инициализируем все проходимые ячейки значением  $\infty$  и стартовую ячейку значением 0.
2. От стартовой позиции делаем все возможные шаги во все стороны, заполняем их значением 1 и запоминаем. От всех запомненных ячеек со значением N ( $N = 1, 2, 3, \dots$ ) делаем все возможные шаги во все стороны. Если в ячейке, в которую сделали шаг, значение больше N+1 (в том числе  $\infty$ ), заполняем ее значением N+1 и запоминаем. Повторяем процедуру для запомненных позиций со значением 2, 3, 4 и т.д. Продолжаем процесс пока не достигнем конечной точки.
3. Строится кратчайший маршрут путем поэлементного перехода от конечного элемента к начальному, направляясь каждый раз в сторону уменьшения значения.

9	10		10	9	8	9	10	11	12	13	14
8	9		9	8	7	8	9	10	11	12	13
7	8	9	8	7	6	7	8	9	10	11	12
6	7	8	7	6	5	6	7			10	11
5					4	5	6	7	8	9	10
4	3	2	1	2	3	4	5	6			11
3	2	1	0	1	2	3	4	5			10
4	3	2	1	2	3	4	5	6	7	8	9

Дополнительно рекомендуем посмотреть алгоритм Дейкстры.