

Диагностическая работа

по подготовке к ГИА

по ИНФОРМАТИКЕ

19 декабря 2013 года

По теме

«Логика и алгоритмы»

9 класс

Вариант ИН90401

Район.

Город (населённый пункт).

Школа

Класс

Фамилия.

Имя

Отчество

Инструкция по выполнению работы

На выполнение диагностической работы по информатике отводится 90 минут. Работа включает в себя 9 заданий. Задание 9.1 или 9.2 выполняется по выбору учащегося.

К каждому заданию с выбором ответа (1 и 3) приводится четыре варианта ответа, из которых только один верный. При выполнении такого задания обведите номер выбранного ответа в работе кружком. Если Вы обвели не тот номер, зачеркните обведённый номер крестиком, а затем обведите номер нового ответа.

Ответы к заданиям 2, 4–8 запишите в работе в отведённом для этого месте. В случае записи неверного ответа зачеркните его и запишите рядом новый.

Задания 9.1 и 9.2 выполняются на компьютере. Результатом выполнения этих заданий являются файлы, которые необходимо сохранить под именем, указанным организаторами экзамена, в формате, также установленном организаторами.

Советуем выполнять задания в том порядке, в котором они даны. В целях экономии времени пропускайте задание, которое не удаётся выполнить сразу, и переходите к следующему. Если после выполнения всей работы у Вас останется время, то Вы сможете вернуться к пропущенным заданиям. Постарайтесь выполнить как можно больше заданий.

Желаем успеха!

- 5 Запишите значение переменной s , полученное в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
<pre> алг нач цел s, k s := 10 нц для k от 6 до 10 s := s + 6 кц вывод s кон </pre>	<pre> DIM k,s AS INTEGER s = 10 FOR k = 6 TO 10 s = s + 6 NEXT k PRINT s </pre>	<pre> Var s,k: integer; Begin s := 10; for k:=6 to 10 do s := s + 6; writeln(s); End. </pre>

Ответ: _____.

- 6 В таблице Dat хранятся данные о количестве покупателей в автосалоне на прошлой неделе ($Dat[1]$ – число покупателей в понедельник, $Dat[2]$ – во вторник и т. д.). Определите, какое число будет напечатано в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Паскаль	Бейсик
<pre> Var k, m: integer; Dat: array[1..7] of integer; Begin Dat[1] := 11; Dat[2] := 20; Dat[3] := 22; Dat[4] := 34; Dat[5] := 10; Dat[6] := 24; Dat[7] := 15; m := 0; for k := 1 to 7 do if Dat[k] > 20 then begin m := Dat[k] end; writeln(m) End. </pre>	<pre> DIM Dat(7) AS INTEGER DIM k,m AS INTEGER Dat(1) = 11 Dat(2) = 20 Dat(3) = 22 Dat(4) = 34 Dat(5) = 10 Dat(6) = 24 Dat(7) = 15 m = 0 FOR k = 1 TO 7 IF Dat(k) > 20 THEN m = Dat(k) END IF NEXT k PRINT m </pre>

Алгоритмический язык

```

алг
нач
  целтаб Dat[1:7]
  цел k, m
  Dat[1] := 11
  Dat[2] := 20
  Dat[3] := 22
  Dat[4] := 34
  Dat[5] := 10
  Dat[6] := 24
  Dat[7] := 15
  m := 0
  нц для k от 1 до 7
    если Dat[k] > 20 то
      m := Dat[k]
    все
  кц
  вывод m
кон

```

Ответ: _____.

- 7 У исполнителя Вычислитель две команды, которым присвоены номера:

1. прибавь 1
2. умножь на 3

Первая из них увеличивает число на экране на 1, вторая – увеличивает его в 3 раза.

Составьте алгоритм получения из числа 2 числа 33, содержащий не более пяти команд. В ответе запишите только номера команд.

(Например, **22211** – это алгоритм

- ```

умножь на 3
умножь на 3
умножь на 3
прибавь 1
прибавь 1,

```

который преобразует число 1 в 29.)

Если таких алгоритмов более одного, то запишите любой из них.

Ответ: \_\_\_\_\_.

**8** Некоторый алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала вычисляется длина исходной цепочки символов; если она чётна, то в середину цепочки символов добавляется символ **А**, а если нечётна, то в конец цепочки добавляется символ **Б**. В полученной цепочке символов каждая буква заменяется буквой, предшествующей ей в русском алфавите (**Б** – на **А**, **В** – на **Б** и т. д., а **А** – на **Я**).

Получившаяся таким образом цепочка является результатом работы алгоритма.

Например, если исходной цепочкой была цепочка **ВРМ**, то результатом работы алгоритма будет цепочка **БПЛА**, а если исходной цепочкой была **ПД**, то результатом работы алгоритма будет цепочка **ОЯГ**.

Дана цепочка символов **ТОР**. Какая цепочка символов получится, если к данной цепочке применить описанный алгоритм дважды (то есть применить алгоритм к данной цепочке, а затем к результату вновь применить алгоритм)?

Русский алфавит:

**АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ.**

Ответ: \_\_\_\_\_.

## Часть 2

*Задания 9.1 и 9.2 выполняются на компьютере. Результатом исполнения задания является отдельный файл (для одного задания – один файл). Формат файла, его имя и каталог для сохранения Вам сообщат организаторы экзамена.*

*Выберите для выполнения ОДНО из предложенных ниже заданий: 9.1 или 9.2.*

**9.1** Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Ниже приведено описание Робота. У Робота есть четыре команды перемещения:

**вверх**

**вниз**

**влево**

**вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может. Если Робот получает команду передвижения через стену, то он разрушается.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится Робот:

**сверху свободно**

**снизу свободно**

**слева свободно**

**справа свободно**

Эти команды можно использовать вместе с условием **если**, имеющим следующий вид:

**если <условие> то**

*последовательность команд*

**все**

*Последовательность команд* – это одна или несколько любых команд, выполняемых Роботом. Например, для передвижения на одну клетку вправо, если справа нет стенки, можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**все**

В одном условии можно использовать несколько команд, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

Для повторения последовательности команд можно использовать цикл **пока**, имеющий следующий вид:

**нц пока** < условие >

*последовательность команд*

**кц**

Например, для безопасного движения вправо можно использовать следующий алгоритм:

**нц пока справа свободно**

**вправо**

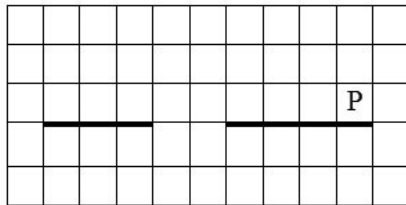
**кц**

Также у Робота есть команда **закрасить**, закрашивающая клетку, в которой Робот находится в настоящий момент.

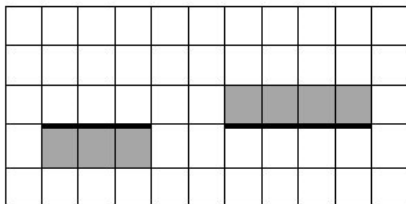
**Выполните задание.**

На бесконечном поле имеется горизонтальная стена. **Длина стены неизвестна**. В стене есть ровно один проход, точное место прохода и его ширина неизвестны. Других стен на поле нет. Робот находится в клетке, расположенной непосредственно над стеной у её правого конца.

На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»):



Напишите для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно выше участка стены справа от прохода, и все клетки, расположенные непосредственно ниже участка стены слева от прохода. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок):



Конечное расположение Робота может быть произвольным. Алгоритм должен решать задачу для произвольного размера поля и любого допустимого расположения стены внутри прямоугольного поля. При исполнении алгоритма Робот не должен разрушиться.

Алгоритм напишите в текстовом редакторе и сохраните в текстовом файле. Название файла и каталог для сохранения вам сообщит учитель.

### 9.2

Напишите программу, которая в последовательности натуральных чисел определяет максимальное число, делящееся нацело на 10. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, делящееся нацело на 10. Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число – максимальное число, оканчивающееся на 0.

**Пример работы программы:**

| Входные данные             | Выходные данные |
|----------------------------|-----------------|
| 4<br>120<br>10<br>4<br>199 | 120             |

**Диагностическая работа**

**по подготовке к ГИА**

**по ИНФОРМАТИКЕ**

**19 декабря 2013 года**

**По теме**

**«Логика и алгоритмы»**

**9 класс**

**Вариант ИН90402**

**Район.**

**Город (населённый пункт).**

**Школа**

**Класс**

**Фамилия.**

**Имя**

**Отчество**

### **Инструкция по выполнению работы**

На выполнение диагностической работы по информатике отводится 90 минут. Работа включает в себя 9 заданий. Задание 9.1 или 9.2 выполняется по выбору учащегося.

К каждому заданию с выбором ответа (1 и 3) приводится четыре варианта ответа, из которых только один верный. При выполнении такого задания обведите номер выбранного ответа в работе кружком. Если Вы обвели не тот номер, зачеркните обведённый номер крестиком, а затем обведите номер нового ответа.

Ответы к заданиям 2, 4–8 запишите в работе в отведённом для этого месте. В случае записи неверного ответа зачеркните его и запишите рядом новый.

Задания 9.1 и 9.2 выполняются на компьютере. Результатом выполнения этих заданий являются файлы, которые необходимо сохранить под именем, указанным организаторами экзамена, в формате, также установленном организаторами.

Советуем выполнять задания в том порядке, в котором они даны. В целях экономии времени пропускайте задание, которое не удаётся выполнить сразу, и переходите к следующему. Если после выполнения всей работы у Вас останется время, то Вы сможете вернуться к пропущенным заданиям. Постарайтесь выполнить как можно больше заданий.

**Желаем успеха!**

**Часть 1**

**1** Для какого из приведённых названий стран **ложно** высказывание: НЕ (Первая буква согласная) ИЛИ (Последняя буква гласная)?

- 1) Ангола      2) Египет      3) Конго      4) Судан

**2** Таня шифрует последовательности русских букв, записывая вместо каждой буквы её номер в алфавите (без пробелов). Номера букв даны в таблице.

|   |    |   |    |   |    |   |    |
|---|----|---|----|---|----|---|----|
| А | 1  | Й | 11 | У | 21 | Э | 31 |
| Б | 2  | К | 12 | Ф | 22 | Ю | 32 |
| В | 3  | Л | 13 | Х | 23 | Я | 33 |
| Г | 4  | М | 14 | Ц | 24 |   |    |
| Д | 5  | Н | 15 | Ч | 25 |   |    |
| Е | 6  | О | 16 | Ш | 26 |   |    |
| Ё | 7  | П | 17 | Щ | 27 |   |    |
| Ж | 8  | Р | 18 | Ъ | 28 |   |    |
| З | 9  | С | 19 | Ы | 29 |   |    |
| И | 10 | Т | 20 | Ь | 30 |   |    |

Некоторые шифровки можно расшифровать не одним способом. Например, 3312 может означать «ЯК», может – «ВЭБ», может – «ЯАБ», а может и «ВВАБ».

Вот четыре шифровки:

2029

2209

3392

7418

Только одна из них расшифровывается единственным способом. Найдите её и расшифруйте. То, что получилось, запишите в качестве ответа.

Ответ: \_\_\_\_\_.

**3** Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду **Сместиться на  $(a, b)$**  (где  $a, b$  – целые числа), перемещающую Чертёжника из точки с координатами  $(x, y)$  в точку с координатами  $(x + a, y + b)$ . Если числа  $a, b$  положительные, значение соответствующей координаты увеличивается, если отрицательные – уменьшается.

Например, если Чертёжник находится в точке с координатами  $(3, 1)$ , то команда **Сместиться на  $(-5, 2)$**  переместит Чертёжника в точку  $(-2, 3)$ .

Запись

**Повтори  $k$  раз**

**Команда1 Команда2 Команда3**

**конец**

означает, что последовательность команд **Команда1 Команда2 Команда3** повторится  $k$  раз.

Чертёжнику был дан для исполнения следующий алгоритм:

**Повтори 3 раз**

**Сместиться на  $(-2, 4)$  Сместиться на  $(0, -2)$  Сместиться на  $(1, 1)$**

**конец**

На какую одну команду можно заменить этот алгоритм, чтобы Чертёжник оказался в той же точке, что и после выполнения алгоритма?

- 1) Сместиться на  $(-1, 3)$       3) Сместиться на  $(-3, 9)$   
 2) Сместиться на  $(1, -3)$       4) Сместиться на  $(3, -9)$

**4** В программе «:=» обозначает оператор присваивания, знаки «+», «-», «\*» и «/» – соответственно операции сложения, вычитания, умножения и деления. Правила выполнения операций и порядок действий соответствуют правилам арифметики.

Определите значение переменной  $a$  после выполнения алгоритма:

$a := 3$

$b := 6$

$b := a*b - a - b$

$a := a + b/a$

В ответе укажите одно целое число – значение переменной  $a$ .

Ответ: \_\_\_\_\_.

- 5 Запишите значение переменной  $s$ , полученное в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

| Алгоритмический язык                                                                            | Бейсик                                                                          | Паскаль                                                                                              |
|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| <pre> алг нач   цел s, k   s := 5   нц для k от 5 до 10     s := s+10   кц   вывод s кон </pre> | <pre> DIM k,s AS INTEGER s = 5 FOR k = 5 TO 10   s = s+10 NEXT k PRINT s </pre> | <pre> Var s,k: integer; Begin   s := 5;   for k:=5 to 10 do     s := s+10;   writeln(s); End. </pre> |

Ответ: \_\_\_\_\_.

- 6 В таблице Dat хранятся данные о количестве посетителей тренажёрного зала на прошлой неделе (Dat[1] – число посетителей в понедельник, Dat[2] – во вторник и т. д.). Определите, какое число будет напечатано в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

| Паскаль                                                                                                                                                                                                                                                                                                       | Бейсик                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> Var k, m: integer; Dat: array[1..7] of integer; Begin   Dat[1] := 11;   Dat[2] := 20;   Dat[3] := 22;   Dat[4] := 34;   Dat[5] := 10;   Dat[6] := 24;   Dat[7] := 15;   m := 0;   for k := 1 to 7 do     if Dat[k] &lt; 15 then       begin         m := Dat[k]       end;     writeln(m)   End. </pre> | <pre> DIM Dat(7) AS INTEGER DIM k,m AS INTEGER Dat(1) = 11 Dat(2) = 20 Dat(3) = 22 Dat(4) = 34 Dat(5) = 10 Dat(6) = 24 Dat(7) = 15 m = 0 FOR k = 1 TO 7   IF Dat(k) &lt; 15 THEN     m = Dat(k)   END IF NEXT k PRINT m </pre> |

### Алгоритмический язык

```

алг
нач
 целтаб Dat[1:7]
 цел k, m
 Dat[1] := 11
 Dat[2] := 20
 Dat[3] := 22
 Dat[4] := 34
 Dat[5] := 10
 Dat[6] := 24
 Dat[7] := 15
 m := 0
 нц для k от 1 до 7
 если Dat[k] < 15 то
 m := Dat[k]
 все
 кц
 вывод m
кон

```

Ответ: \_\_\_\_\_.

- 7 У исполнителя Вычислитель две команды, которым присвоены номера:

1. прибавь 1
2. умножь на 3

Первая из них увеличивает число на экране на 1, вторая – увеличивает его в 3 раза.

Составьте алгоритм получения из числа 2 числа 25, содержащий не более 5 команд. В ответе запишите только номера команд.

(Например, **22211** – это алгоритм

- умножь на 3
- умножь на 3
- умножь на 3
- прибавь 1
- прибавь 1,

который преобразует число 1 в 29.)

Если таких алгоритмов более одного, то запишите любой из них.

Ответ: \_\_\_\_\_.



**8** Некоторый алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала вычисляется длина исходной цепочки символов; если она чётна, то в середину цепочки символов добавляется символ **Ч**, а если нечётна, то в начало цепочки добавляется символ **Н**. В полученной цепочке символов каждая буква заменяется буквой, предшествующей ей в русском алфавите (**Б** – на **А**, **В** – на **Б** и т. д., а **А** – на **Я**).

Получившаяся таким образом цепочка является результатом работы алгоритма.

Например, если исходной цепочкой была цепочка **ВРМ**, то результатом работы алгоритма будет цепочка **МБПЛ**, а если исходной цепочкой была **ПД**, то результатом работы алгоритма будет цепочка **ОЦГ**.

Дана цепочка символов **РОТ**. Какая цепочка символов получится, если к данной цепочке применить описанный алгоритм дважды (то есть применить алгоритм к данной цепочке, а затем к результату вновь применить алгоритм)?

Русский алфавит:

**АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ.**

Ответ: \_\_\_\_\_.

## Часть 2

*Задания 9.1 и 9.2 выполняются на компьютере. Результатом исполнения задания является отдельный файл (для одного задания – один файл). Формат файла, его имя и каталог для сохранения Вам сообщат организаторы экзамена.*

*Выберите для выполнения ОДНО из предложенных ниже заданий: 9.1 или 9.2.*

**9.1** Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Ниже приведено описание Робота. У Робота есть четыре команды перемещения:

**вверх**

**вниз**

**влево**

**вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может. Если Робот получает команду передвижения через стену, то он разрушается.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится Робот:

**сверху свободно**

**снизу свободно**

**слева свободно**

**справа свободно**

Эти команды можно использовать вместе с условием **если**, имеющим следующий вид:

**если <условие> то**

*последовательность команд*

**все**

*Последовательность команд* – это одна или несколько любых команд, выполняемых Роботом. Например, для передвижения на одну клетку вправо, если справа нет стенки, можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**все**

В одном условии можно использовать несколько команд, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

Для повторения последовательности команд можно использовать цикл **пока**, имеющий следующий вид:

**нц пока** < условие >

*последовательность команд*

**кц**

Например, для безопасного движения вправо можно использовать следующий алгоритм:

**нц пока справа свободно**

**вправо**

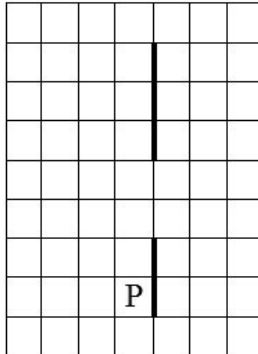
**кц**

Также у Робота есть команда **закрасить**, закрашивающая клетку, в которой Робот находится в настоящий момент.

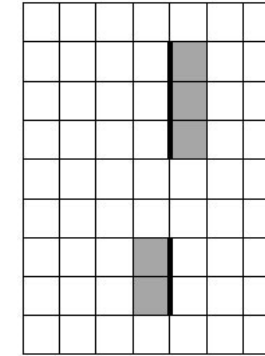
**Выполните задание.**

На бесконечном поле имеется вертикальная стена. **Длина стены неизвестна.** В стене есть ровно один проход, точное место прохода и его ширина неизвестны. Других стен на поле нет. Робот находится в клетке, расположенной непосредственно слева от стены у её нижнего конца.

На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»):



Напишите для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно левее участка стены ниже прохода, и все клетки, расположенные непосредственно правее участка стены выше прохода. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок):



Конечное расположение Робота может быть произвольным. Алгоритм должен решать задачу для произвольного размера поля и любого допустимого расположения стены внутри прямоугольного поля. При исполнении алгоритма Робот не должен разрушиться.

Алгоритм напишите в текстовом редакторе и сохраните в текстовом файле. Название файла и каталог для сохранения вам сообщит учитель.

**9.2**

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, делящееся нацело на 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, делящееся нацело на 6. Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число – минимальное число, делящееся нацело на 6.

**Пример работы программы:**

| Входные данные | Выходные данные |
|----------------|-----------------|
| 4              | 120             |
| 120            |                 |
| 10             |                 |
| 4              |                 |
| 300            |                 |