

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 88, апрель 2024 г.

011000100010101001001111100100011001001000101001001001011

ОГЭ

О заданиях 5 ОГЭ по информатике

1. Общие вопросы.

В спецификации контрольных измерительных материалов для проведения в 2024 году Основного государственного экзамена по информатике в качестве проверяемого результата обучения применительно к заданию 5 указывается «Умение анализировать простые алгоритмы для конкретного исполнителя с фиксированным набором команд».

Тема в федеральном компоненте ГОС ООО – «Алгоритм, свойства алгоритмов, способы записи алгоритмов. Блок-схемы. Представление о программировании».

Уровень сложности – базовый.

Максимальный балл за задание – 1.

Примерное время выполнения задания (мин.) – 6.

В демонстрационных вариантах контрольных измерительных материалов ОГЭ по информатике нескольких последних лет и в открытом банке заданий ОГЭ представлены задания, общий вид которых такой:

«У исполнителя *<указывается имя исполнителя>* две команды, которым присвоены номера:

1. *<указывается первая команда>*
2. *<указывается вторая команда>*

Первая из них *<указывается, что выполняется по первой команде>*, вторая *<указывается, что выполняется по второй команде>*. Алгоритм для исполнителя – это последовательность номеров команд.

Составьте алгоритм получения из числа *<указывается число>* числа *<указывается число>*, содержащий не более *<указывается число>* команд. В ответе запишите только номера команд.

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

Примечание. Может быть также приведён пример алгоритма для конкретного исполнителя.

Возможные команды исполнителя:

- прибавь *<указывается число>*;
- вычти *<указывается число>*;
- умножь на *<указывается число>*;
- раздели на *<указывается число>*;
- возведи в квадрат;
- припиши *<указывается число>*;
- зачеркни справа.

Пример задания: «У исполнителя Делитель две команды, которым присвоены номера:

1. раздели на 2
2. прибавь 1

Первая из них уменьшает число на экране в 2 раза, вторая увеличивает его на 1. Алгоритм для исполнителя – это последовательность номеров команд. Исполнитель работает только с натуральными числами.

Составьте алгоритм получения из числа 23 числа 4, содержащий не более 5 команд. В ответе запишите только номера команд».

2. Примеры задач и их методика их выполнения

Обсудим решения ряда задач указанного типа.

2.1. Задача 1

Условие

У исполнителя Вычислитель две команды, которым присвоены номера:

1. умножь на 3
2. прибавь 2

Первая из них увеличивает число на экране в 3 раза, вторая увеличивает его на 2.

Составьте алгоритм получения из числа 3 числа 69, содержащий не более 5 команд. В ответе запишите только номера команд.

(Например, 11221 – это алгоритм:

умножь на 3

умножь на 3

прибавь 2

прибавь 2

умножь на 3,

который преобразует число 1 в 39.)

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

Решение

Если пытаться написать алгоритм, так сказать, «начиная с исходного числа (3)», то он будет получено много вариантов (из числа 3 можно получить 9 и 6, из каждого из этих двух чисел ещё по 2 числа и т. д.). Начинать лучше с конечного числа.

Число-результат 69 можно получить из двух чисел 67 (выполнив команду 2) и 23 (выполнив команду 1). Какой вариант лучше с точки зрения меньшего количества команд в алгоритме? – использование команды 1. Значит, последняя команда алгоритма будет 1.

Число 23 разделить на 3 нельзя (полученный результат будет нецелым, а по условию исполнитель может работать только с натуральными, то есть целыми положительными числами). Значит, для его получения должна быть применена команда 2 (вся конечная часть алгоритма будет такой: 21), то есть число, предшествующее числу 23, это 21.

Для числа 21 оптимальным предшествующим числом будет число 7, то есть третья справа команда алгоритма – 1, а вся его конечная часть – 121.

Продолжая аналогичные рассуждения, получим весь алгоритм решения задачи: 22121. Для его проверки можно выписать меняющиеся при выполнении алгоритма числа:

$3 \rightarrow 5 \rightarrow 7 \rightarrow 21 \rightarrow 23 \rightarrow 69$

Видно, что при решении мы использовали команды, так сказать, «противоположные» заданным: деление на 3 и вычитание числа 2.

Сразу же заметим, что не всегда последняя команда алгоритма – это команда, значительно уменьшающая или увеличивающая число, предшествующее конечному (умножь на 3, раздели на 2 и т. п.). Рассмотрим ещё пример.

2.2. Задача 2

Условие

У исполнителя Вычислитель две команды, которым присвоены номера:

1. вычти 1
2. умножь на 3

Первая из них уменьшает число на экране на 1, вторая утраивает его.

Составьте алгоритм получения из числа 4 числа 23, содержащий не более пяти команд.

В ответе запишите только номера команд.

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

Решение

Здесь число 23 не кратно трем, то есть последняя команда алгоритма не может быть умножь на 3. Значит, последняя команда – вычти 1, в которой уменьшалось число 24 (23 + 1). Число 24 кратно трем, то есть предпоследняя команда алгоритма может быть умножь на 3. Продолжая рассуждения, получим искомый алгоритм: 12121:

```
вычти 1
умножь на 3
вычти 1
умножь на 3
вычти 1
```

Проверка: $4 \rightarrow 3 \rightarrow 9 \rightarrow 8 \rightarrow 24 \rightarrow 23$.

Обратим внимание на то, что при решении также использовались команды, так сказать, «противоположные» заданным: увеличение числа на 1 и деление на 3.

2.3. Задача 3

Условие

У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат
2. вычти 4

Первая из них возводит число во вторую степень, вторая уменьшает число на 4.

Исполнитель работает только с натуральными числами.

Составьте алгоритм получения из числа 5 числа 69, содержащий не более пяти команд.

В ответе запишите только номера команд.

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

Решение

В данном случае задача облегчается тем, что конечное число (169) является точным квадратом. Это значит, что последняя команда алгоритма решения задачи – возведи в квадрат. Из предшествующего числа (13) получить начальное 5 можно, только «через» его квадрат (25), то есть весь искомый алгоритм 12221:

```
возведи в квадрат
вычти 4
вычти 4
вычти 4
возведи в квадрат
```

2.4. Задача 4

Условие

У исполнителя Конструктор две команды, которым присвоены номера:

1. припиши 2
2. раздели на 2

Первая из них приписывает к числу справа цифру 2, вторая делит его на 2.

Составьте алгоритм получения из числа 8 числа 6, содержащий не более 5 команд. В ответе запишите только номера команд.

Если таких алгоритмов более одного, запишите любой из них.

Решение

Здесь последней командой может быть только раздели на 2. Значит, число «перед числом 6» – 12. Оно может быть получено из числа 1 командой припиши 2 (команда раздели на 2, использующая число 24, явно не подходит). Осталось от числа 1 перейти к начальному числу 8. Это можно сделать, трижды применив умножение на 2 (команду, «противоположную» заданной команде раздели на 2)¹. Весь алгоритм решения задачи – 22212:

раздели на 2
раздели на 2
раздели на 2
припиши 2
раздели на 2

2.5. Задача 5

Условие

У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат
2. зачеркни справа

Первая из них возводит число на экране во вторую степень, вторая – удаляет крайнюю правую цифру числа.

Составьте алгоритм получения из числа 52 числа 16, содержащий не более 5 команд.

В ответе запишите только номера команд.

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

Решение

Это самая сложная задача из рассмотренных. Дело в том, что здесь конечное число 16 может быть получено двумя способами:

- 1) из числа 169, являющегося полным квадратом (то есть оно может быть получено применением команды возведи в квадрат применительно к числу 13;
- 2) из числа 4.

В первом случае указанное число 13 за 3 команды не может быть «связано» с исходным числом 52. Это вариант можно не рассматривать.

Во втором случае мы пришли к более простой задаче – получения из числа 52 числа 4. Так как число 4 является точным квадратом, то эта, «промежуточная», задача может быть решена так:

зачеркни справа	получим 5
возведи в квадрат	получим 25
зачеркни справа	получим 2
возведи в квадрат	получим 4

Весь алгоритм получения числа 16: 21211.

3. Рекомендации по выполнению задания 5

Поиск ответа надо начинать с конечного числа.

Если последнее число можно получить из некоторого предшествующего числа только одной командой исполнителя, то, естественно, в алгоритме должна быть применена именно эта команда, а предшествующее ему число определяется с помощью «противоположных»

¹ Можно также подумать, как из числа 1 получить 8, используя заданные команды исполнителя.

действий (см. решение задач 2 и 4 выше). Примеры «противоположных» действий: для команды `умножь на 3` – деление на 3, вместо `прибавь 2` – вычитание числа 2, вместо `возведи в квадрат` – извлечение квадратного корня и т. п.

«Противоположные» действия удобно использовать для определения предшествующих чисел и на других этапах формирования алгоритма.

Если последнее число можно получить одной из двух команд, то надо использовать ту из них, которая меняет числа значительно (с умножением, делением и возведением в квадрат) – см. решение задачи 1. Такие команды для краткости в дальнейшем будем называть «быстрыми».

Если «быструю» команду применить нельзя, то надо стремиться получить число, которой можно получить с помощью «быстрой» команды. Например, для исполнителя с командами `возведи в квадрат` и `вычти 2` для последнего числа 47 следует использовать противоположное действие – прибавление 2, чтобы получить число – точный квадрат 49. При этом последней командой алгоритма должна быть `вычти 2`.

Аналогично надо поступать на всех дальнейших этапах формирования алгоритма до получения исходного числа.

Удобно при решении выписывать (справа налево) получающиеся предшествующие числа, а после формирование всего алгоритма проверить результат, как это делалось выше.

4. Задания для самостоятельной работы

1. У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат
2. вычти 4

Первая из них возводит число на экране во вторую степень, вторая уменьшает число на 4.

Исполнитель работает только с натуральными числами.

Составьте алгоритм получения из числа 2 числа 64, содержащий не более пяти команд.

В ответе запишите только номера команд.

2. У исполнителя Конструктор две команды, которым присвоены номера:

1. припиши 2
2. раздели на 2

Первая из них приписывает к числу на экране справа цифру 2, вторая делит его на 2.

Составьте алгоритм получения из числа 14 числа 9, содержащий не более 5 команд. В ответе запишите только номера команд.

3. У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат
2. зачеркни справа

Первая из них возводит число на экране во вторую степень, вторая – удаляет крайнюю правую цифру числа.

Составьте алгоритм получения из числа 3 числа 6, содержащий не более 5 команд. В ответе запишите только номера команд.

4. У исполнителя Умножатор две команды, которым присвоены номера:

1. умножь на 3
2. прибавь 1

Первая из них увеличивает число на экране в 3 раза, вторая прибавляет к числу 1.

Составьте алгоритм получения из числа 2 числа 84, содержащий не более 5 команд. В ответе запишите только номера команд.

(Например, 12212 – это алгоритм:

- умножь на 3
- прибавь 1
- прибавь 1

умножь на 3

прибавь 1

который преобразует число 2 в 25.)

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

5. У исполнителя Квадратор две команды, которым присвоены номера:

1. возведи в квадрат

2. прибавь 2

Первая из них возводит число на экране во вторую степень, вторая увеличивает его на

2.

Составьте алгоритм получения из числа 1 числа 85, содержащий не более 5 команд. В ответе запишите только номера команд.

(Например, 12221 – это алгоритм:

возведи в квадрат

прибавь 2

прибавь 2

прибавь 2

возведи в квадрат,

который преобразует число 3 в 225.)

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

6. У исполнителя Квадратор две команды, которым присвоены номера:

1. раздели на 2

2. возведи в квадрат

Первая из них уменьшает число на экране в 2 раза, вторая возводит число в квадрат.

Исполнитель работает только с натуральными числами.

Составьте алгоритм получения из числа 12 числа 81, содержащий не более 4 команд. В ответе запишите только номера команд.

(Например, 1212 – это алгоритм:

раздели на 2

возведи в квадрат

раздели на 2

возведи в квадрат

который преобразует число 20 в 2500.)

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

7. У исполнителя Дельта две команды, которым присвоены номера:

1. вычти 5

2. раздели на 2

Первая из них уменьшает число на 5, вторая – уменьшает число в 2 раза.

Исполнитель работает только с натуральными числами.

Составьте алгоритм получения из числа 65 числа 20, содержащий не более 4 команд. В ответе запишите только номера команд.

Комментарии к решению

Если, согласно сделанным выше рекомендациям по выполнению задания 5, для последнего числа 20 сразу применить действие, «противоположное» быстрой команде (раздели на 2), то полученное число 40 не позволит «прийти» к исходному числу 65 за оставшиеся 3 действия.

Но даже если сначала применить действие, «противоположное» команде 1 (увеличить число на 5), то и в этом случае полученное число 25 не позволит прийти к «исходному» числу за требуемое количество действий (убедитесь в этом!).

Оптимальный алгоритм: 1211:

вычти 5

раздели на 2

вычти 5

вычти 5

То есть в данном случае действие, «противоположное» быстрой команде, используется на третьем от конца алгоритма шаге. Это следует учитывать при выборе действия, «противоположного» быстрым командам.

5.5. Задания 5 в другой формулировке

В заключение заметим, что в демонстрационном варианте ОГЭ 2024 года [3] представлено задание в несколько другой формулировке: «У исполнителя Альфа две команды, которым присвоены номера:

1. прибавь 1
2. умножь на b

(b – неизвестное натуральное число; $b \geq 2$).

Первая из них увеличивает число на 1, вторая умножает его на b . Алгоритм для исполнителя Альфа – это последовательность номеров команд. Найдите значение числа b , при котором из числа 6 по алгоритму 11211 будет получено число 82».

Решение

О команде 1 известна вся информация, поэтому её можно дважды применить к начальному числу 6 – будет получено число 8, а к конечному числу 82 дважды применить действие, которое выше было названо «противоположным» этой команде – уменьшение числа на 1 (будет получено 80). По команде 2 (умножь на b) из числа 8 можно получить 80 при $b = 10$.

Ответ: 10.

Задания для самостоятельной работы

8. У исполнителя Омега две команды, которым присвоены номера:

1. раздели на x (x – неизвестное натуральное число; $x \geq 2$).
2. вычти 3

Первая из них уменьшает число в x раз, вторая – уменьшает 3. Алгоритм для исполнителя Омега – это последовательность номеров команд. Найдите значение числа x , при котором из числа 59 по алгоритму 22212 будет получено число 22.

9. У исполнителя Вычислитель две команды, которым присвоены номера:

1. прибавь x (x – неизвестное натуральное число; $x \geq 2$).
2. умножь на 2

Первая из них увеличивает число на x , вторая – увеличивает в 2 раза. Алгоритм для исполнителя – это последовательность номеров команд. Найдите значение числа x , при котором из числа 3 по алгоритму 22211 будет получено число 40.

10. У исполнителя Дельта две команды, которым присвоены номера:

1. вычти 4
2. припиши n

(n – неизвестное натуральное число).

Первая из них уменьшает число на 4, вторая приписывает к числу некоторое число n . Алгоритм для исполнителя Дельта – это последовательность номеров команд. Найдите значение числа n , при котором из числа 11 по алгоритму 11221 будет получено число 351.

11. У исполнителя Квадратор две команды, одной из которых присвоен номер 1, а второй – номер 2:

1. возведи в квадрат
- 2.

Команда 1 возводит число в квадрат. Алгоритм для исполнителя Квадратор – это последовательность номеров команд. Установите команду, которой присвоен номер 2, если по алгоритму 11221 из числа 5 будет получено число 36.

12. У исполнителя Дельта две команды, одной из которых присвоен номер 1, а второй – номер 2:

1. вычти 2
- 2.

Команда 1 уменьшает число на 2. Алгоритм для исполнителя Дельта – это последовательность номеров команд. Установите команду, которой присвоен номер 2, если по алгоритму 12111 из числа 9 будет получено число 43. Возможны 2 варианта ответа (найдите оба).

О заданиях 6 ОГЭ по информатике

1. Общие вопросы

В Спецификации контрольных измерительных материалов ОГЭ по информатике в качестве проверяемого результата обучения применительно к заданию 6 указывается «Умение формально исполнять алгоритмы, записанные на языке программирования».

Тема в федеральном компоненте ГОС ООО – «Алгоритм, свойства алгоритмов, способы записи алгоритмов. Блок-схемы. Представление о программировании».

Уровень сложности – базовый.

Максимальный балл за задание – 1.

Примерное время выполнения задания (мин.) – 4.

В демонстрационных вариантах контрольных измерительных материалов ОГЭ по информатике нескольких последних лет и в открытом банке заданий ОГЭ представлены задания 6 нескольких типов, простейший из которых имеет вид:

1) в начале записано: «Ниже приведена программа, записанная на пяти языках программирования» и приводятся соответствующие варианты программы. На школьном алгоритмическом языке программа такая:

алг

нач цел s, t | Могут быть указаны другие имена переменных величин

ввод s

ввод t

если \langle указывается условие \rangle

то вывод "YES"

иначе вывод "NO"

все

кон

2) в конце идет текст: «Было проведено 9 запусков программы, при которых в качестве значений переменных вводились следующие пары чисел (s, t):

\langle приводятся 9 пар чисел \rangle

Сколько было запусков, при которых программа напечатала "YES"?» (может быть также «...напечатала "NO"?»)

2. Примеры заданий и методика их выполнения

2.1. Пример 1

Условие

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль	Бейсик
<pre> алг нач цел s, t ввод s ввод t если s >= 2 * t то вывод "YES" иначе вывод "NO" все кон </pre>	<pre> var s, t: integer; begin readln(s); readln(t); if s >= 2 * t then writeln ('YES') else writeln ('NO') end. </pre>	<pre> DIM t, s AS INTEGER INPUT s INPUT t IF s >= 2 * t THEN PRINT "YES" ELSE PRINT "NO" END IF </pre>
Python	C++	
<pre> s = int(input()) t = int(input()) if s >= 2 * t: print("YES") else: print("NO") </pre>	<pre> #include <iostream> using namespace std; int main() { int s, t; cin >> s; cin >> t; if (s >= 2 * t) cout << "YES"; else cout << "NO"; return 0; } </pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел:

(1, 2); (8, 4); (6, -12); (-5, -5); (3, 11); (-10, 12); (-10, -2); (4, 1); (2, 5).

Сколько было запусков, при которых программа напечатала «YES»?

Решение

Анализ показывает, что в программе при выводе на экран использован полный условный оператор.

Условие – первое число в паре не меньше второго. Такие пары:

(8, 4); (6, -12); (-5, -5)²; (4, 1)

Всего таких пар – 4.

Еще ряд примеров.

2.2. Пример 2

Условие

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль	Бейсик
<pre> алг нач цел s, t ввод s ввод t если div(s, 2) = t </pre>	<pre> var s, t: integer; begin readln(s); readln(t); </pre>	<pre> DIM t, s AS INTEGER INPUT s INPUT t IF s \ 2 = t THEN PRINT "YES" </pre>

² Обратите внимание на то, что среди приведённых пар есть такая: (-5, -5).

<pre> то вывод "YES" иначе вывод "NO" все кон </pre>	<pre> if s div 2 = t then writeln ('YES') else writeln ('NO') end. </pre>	<pre> ELSE PRINT "NO" END IF </pre>
Python	C++	
<pre> s = int(input()) t = int(input()) if s // 2 == t: print("YES") else: print("NO") </pre>	<pre> #include <iostream> using namespace std; int main() { int s, t; cin >> s; cin >> t; if (s / 2 == t) cout << "YES"; else cout << "NO"; return 0; } </pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел:

(1, 1); (8, 4); (14, 10); (20, 1); (7, 3); (10, 5); (10, 2); (4, 1); (1, 0).

Сколько было запусков, при которых программа напечатала «YES»?

Решение

Прежде всего, обратим внимание на то, что в приведённых программах имеет место деление значения переменной нацело на 2. Именно – нацело. Это значит, что при учете пар, для которых условие $\text{div}(s, 2) = t$ является истинным, пара (7, 3) также должна быть учтена. Все подходящие пары:

(8, 4); (7, 3); (10, 5)

Ответ: 3.

2.3. Пример 3

Условие

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	П а с к а л ь	Python
<pre> алг нач цел s, t ввод s ввод t если s > 10 или t > 10 то вывод "YES" иначе вывод "NO" все кон </pre>	<pre> var s, t: integer; begin readln(s); readln(t); if (s > 10) or (t > 10) then writeln ('YES') else writeln ('NO') end. </pre>	<pre> s = int(input()) t = int(input()) if (s > 10) or (t > 10): print("YES") else: print("NO") </pre>
Бейсик	C++	
<pre> DIM t, s AS INTEGER INPUT s INPUT t IF s > 10 OR t > 10 THEN PRINT "YES" ELSE PRINT "NO" END IF </pre>	<pre> #include <iostream> using namespace std; int main() { int s, t; cin >> s; cin >> t; if (s > 10) or (t > 10) cout << "YES"; else cout << "NO"; } </pre>	

```
return 0;
}
```

Было проведено 9 запусков программы, при которых в качестве значений переменных вводились следующие пары чисел (s, t) :

$(1, 2)$; $(11, 2)$; $(1, 12)$; $(11, 12)$; $(-11, -12)$; $(-11, 12)$; $(-12, 11)$; $(10, 10)$; $(10, 5)$.

Сколько было запусков, при которых программа напечатала «YES»?

Решение

Обратим внимание на то, что в программе использовано сложное условие с логической связкой **или** (**or**), которое, как известно, будет истинным, если истинным является хотя бы одно из простых условий $s > 10$ или $t > 10$.

Значит, надо учитывать пары, в которых хотя бы одно число больше 10. Таких пар – 5.

$(11, 2)$; $(1, 12)$; $(11, 12)$; $(-11, 12)$; $(-12, 11)$

Подводя некоторые промежуточные итоги, заметим, что поскольку задание выполняется на компьютере, то удобно подходящие пары чисел учитывать на бумаге следующим образом:

| | ...

3. Задания 6 другой формулировки

В открытом банке заданий ОГЭ представлены также задания, отличающиеся от рассмотренных. Их общий вид:

1) в начале записано: «Ниже приведена программа, записанная на пяти языках программирования» и приводятся соответствующие варианты программы. На школьном алгоритмическом языке программа имеет вид:

алг

нач цел s, t, A | Могут быть указаны другие имена переменных величин

ввод s

ввод t

ввод A

если \langle указывается условие с переменными s, t, A \rangle

то вывод "YES"

иначе вывод "NO"

все

кон

2) в конце идет текст: «Было проведено 9 запусков программы, при которых в качестве значений переменных вводились следующие пары чисел (s, t) :

\langle приводятся 9 пар чисел \rangle

и собственно задание:

«Укажите целое значение параметра A , при котором для указанных входных данных программа напечатает "YES" \langle указывается число \rangle раз» (может быть указано и «напечатает "NO"»)

или

«Укажите количество целых значений параметра A , при которых для указанных входных данных напечатает "YES" \langle указывается число \rangle раз» (может быть указано и «напечатает "NO"»)

или

«Укажите наименьшее целое значение параметра A , при котором для указанных входных данных напечатает "YES" \langle указывается число \rangle раз» (может быть указано и «напечатает "NO"»)

или

«Укажите наибольшее целое значение параметра A , при котором для указанных входных данных напечатает «YES» <указывается число> раз» (может быть указано и «напечатает «NO»»).

4. Примеры заданий в другой формулировке и методика их выполнения

4.1. Пример 1

Условие

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль	Python
<pre>алг нач цел s, t, A ввод s ввод t если s > 10 или t > A то вывод "YES" иначе вывод "NO" все кон</pre>	<pre>var s, t, A: integer; begin readln(s); readln(t); readln(A); if (s > 10) or (t > A) then writeln ('YES') else writeln ('NO') end.</pre>	<pre>s = int(input()) t = int(input()) A = int(input()) if (s > 10) or (t > A): print("YES") else: print("NO")</pre>
Бейсик	C++	
<pre>DIM t, s, A AS INTEGER INPUT s INPUT t INPUT A IF s > 10 OR t > A THEN PRINT "YES" ELSE PRINT "NO" END IF</pre>	<pre>#include <iostream> using namespace std; int main() { int s, t, A; cin >> s; cin >> t; cin >> A; if (s > 10) or (t > A) cout << "YES"; else cout << "NO"; return 0; }</pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел:

(1, 2); (11, 2); (1, 12); (11, 12); (-11, -12); (-11, 12); (-12, 11); (10, 10); (10, 5).

Укажите целое значение параметра A , при котором для указанных входных данных программа напечатает «YES» четыре раза.

Решение

Сначала следует сравнить первое числа каждой пары с числом 10. Если оно больше 10, то для данной пары чисел «YES» будет выведено независимо от второго значения в ней. Таких пар – две:

(11, 2) и (11, 12)

Так как по условию «YES» должно быть напечатано четыре раза, то осталось найти такое значение A , при котором только в двух парах второе число будет больше этого значения. Удобно это сделать так.

Выпишем (с возможными повторами) вторые числа всех пар, кроме двух указанных чуть выше пар:

2 12 -12 12 11 10 5

и расположим их в порядке возрастания (это можно было сделать и сразу при выписывании чисел):

-12 2 5 10 11 12 12

Анализ показывает, что ещё две пары будут «нужными» при $A = 11^3$; это пары (1, 12) и (-11, 12).

Ответ: 11.

4.2. Пример 2

Условие

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	П а с к а л ь	Python
<pre>алг нач цел s, t, A ввод s ввод t если s > 11 или t > A то вывод "YES" иначе вывод "NO" все кон</pre>	<pre>var s, t, A: integer; begin readln(s); readln(t); readln(A); if (s > 11) or (t > A) then writeln ('YES') else writeln ('NO') end.</pre>	<pre>s = int(input()) t = int(input()) A = int(input()) if (s > 11) or (t > A): print("YES") else: print("NO")</pre>
<pre>Бейсик DIM t, s, A AS INTEGER INPUT s INPUT t INPUT A IF s > 11 OR t > A THEN PRINT "YES" ELSE PRINT "NO" END IF</pre>	<pre>C++ #include <iostream> using namespace std; int main() { int s, t, A; cin >> s; cin >> t; cin >> A; if (s > 11) or (t > A) cout << "YES"; else cout << "NO"; return 0; }</pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел:

(11, 2); (12, 2); (1, 15); (12, 12); (-11, -12); (-11, 15); (-12, 14); (10, 10); (11, 15).

Укажите целое значение параметра A , при котором для указанных входных данных программа напечатает «NO» четыре раза.

Решение

Здесь отличие в том, что задано количество выводов на экран слова «NO», то есть следует учитывать пары чисел (s, t) , при которых условие $s > 11$ **или** $t > A$ будет ложным.

Задачу удобно решить, так сказать, «от противного» – найти такое значение A , при котором для 9 заданных пар на экран будет выведено 5 раз слово «YES» ($5 = 9 - 4$). Аналогичная задача решалась до этого. По условию $s > 11$ подходят пары:

(12, 2) и (12, 12);

Еще только три вывода на экран слова «YES» «обеспечивают» пары, у которых второе число равно 15:

(1, 15); (-11, 15); (11, 15).

при $A = 14$.

³ Убедитесь в том, что при $A = 10$ слово «YES» будет напечатано пять раз.

Ответ: 14.

Второй способ решения – искать пары чисел (s, t) , при которых условие $s > 11$ **или** $t > A$ будет ложным (при которых будет выведено «NO»). Для этого надо вспомнить закон отрицания от конъюнкции. Указанное условие будет ложными при

$$s \leq 11 \text{ и } t \leq A$$

Итак, начнем. Сначала следует сравнить первое числа каждой пары с числом 11. Если оно меньше или равно 11, то для данной пары *может быть* выведено «NO». Такие пары отгнены:

(11, 2); (12, 2); (1, 15); (12, 12); (-11, -12); (-11, 15); (-12, 14); (10, 10); (11, 15).

Но чтобы действительно было выведено «NO», нужно, чтобы второе число этих пар было меньше либо равно A и при этом таких «подходящих» пар должно быть 4. При $A = 15$ таких пар больше, чем 4, а при $A = 14$ их – 4:

(11, 2); (12, 2); (1, 15); (12, 12); (-11, -12); (-11, 15); (-12, 14); (10, 10); (11, 15).

Ответ: 14.

Конечно, первый способ решения лучше.

4.3. Пример 3

Условие

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль	Python
<pre>алг нач цел s, t, A ввод s ввод t если s > 10 или t > A то вывод "YES" иначе вывод "NO" все кон</pre>	<pre>var s, t, A: integer; begin readln(s); readln(t); readln(A); if (s > 10) or (t > A) then writeln ('YES') else writeln ('NO') end.</pre>	<pre>s = int(input()) t = int(input()) A = int(input()) if (s > 10) or (t > A): print("YES") else: print("NO")</pre>
<pre>Бейсик DIM t, s, A AS INTEGER INPUT s INPUT t INPUT A IF s > 10 OR t > A THEN PRINT "YES" ELSE PRINT "NO" END IF</pre>	<pre>C++ #include <iostream> using namespace std; int main() { int s, t, A; cin >> s; cin >> t; cin >> A; if (s > 10) or (t > A) cout << "YES"; else cout << "NO"; return 0; }</pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел:

(1, 2); (11, 2); (1, 12); (11, 12); (-11, -12); (-11, 12); (-12, 11); (10, 10); (10, 5).

Укажите количество целых значений параметра A , при которых для указанных входных данных программа напечатает «YES» шесть раз.

Решение

Как и ранее, сначала следует сравнить первое числа каждой пары с числом 10. Пар, в которых оно больше 10, – две:

(11, 2) и (11, 12)

Осталось найти все такие значение A , при которых только в *четырёх* парах второе число будет больше этого значения. Отсортируем в порядке возрастания все вторые числа пар, кроме одной двойки и одного числа 12

–12 2 5 10 11 12 12 12

Видно, что для пар с такими вторыми числами вывод слова «YES» четыре раза будет происходить только при $A = 10$, то есть для пар в которых вторые числа равны 11 и 12. При $A = 5$ это слово будет выведено семь раз ($2 + 5$), при $A = 11$ – пять раз ($2 + 3$), при $A = 12$ – два раза, что не соответствует условию.

Значит количество искомым значений A равно 1.

Ответ: 2.

4.4. Пример 4

Условие

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль	Python
<pre>алг нач цел s, t, A ввод s ввод t если s > A или t > 12 то вывод "YES" иначе вывод "NO" все кон</pre>	<pre>var s, t, A: integer; begin readln(s); readln(t); readln(A); if (s > A) or (t > 12) then writeln ('YES') else writeln ('NO') end.</pre>	<pre>s = int(input()) t = int(input()) A = int(input()) if (s > A) or (t > 12): print("YES") else: print("NO")</pre>
<pre>Бейсик DIM t, s, A AS INTEGER INPUT s INPUT t INPUT A IF s > A OR t > 12 THEN PRINT "YES" ELSE PRINT "NO" END IF</pre>	<pre>C++ #include <iostream> using namespace std; int main() { int s, t, A; cin >> s; cin >> t; cin >> A; if (s > A) or (t > 12) cout << "YES"; else cout << "NO"; return 0; }</pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел:

(13, 2); (11, 12); (–12, 12); (2, –2); (–10, –10); (6, –5); (2, 8); (9, 10); (1, 13).

Укажите наименьшее целое значение параметра A , при котором для указанных входных данных программа напечатает «YES» пять раз.

Решение

Здесь особенность в том, что со значением A сравнивается первое число каждой пары.

Задача решается аналогично предыдущей. По условию $t > 12$ подходит одна пара (1, 13).

Ищем ещё четыре пары. Выпишем первые числа оставшихся пар в порядке возрастания:

-12 -10 2 2 6 9 11 13

Видно, что для таких пар вывод слова «YES» четыре раза будет происходить при $A > 2$ и $A < 6$. Минимальное из таких значений равно 3.

Ответ: 3.

4.5. Пример 5

Условие

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	П а с к а л ь	Python
<pre>алг нач цел s, t, A ввод s ввод t если s > A или t > 11 то вывод "YES" иначе вывод "NO" все кон</pre>	<pre>var s, t, A: integer; begin readln(s); readln(t); readln(A); if (s > A) or (t > 11) then writeln ('YES') else writeln ('NO') end.</pre>	<pre>s = int(input()) t = int(input()) A = int(input()) if (s > A) or (t > 11): print("YES") else: print("NO")</pre>
Бейсик	С++	
<pre>DIM t, s, A AS INTEGER INPUT s INPUT t INPUT A IF s > A OR t > 11 THEN PRINT "YES" ELSE PRINT "NO" END IF</pre>	<pre>#include <iostream> using namespace std; int main() { int s, t, A; cin >> s; cin >> t; cin >> A; if (s > A) or (t > 11) cout << "YES"; else cout << "NO"; return 0; }</pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел:

(-9, 11); (2, 7); (5, 12); (2, -2); (7, -9); (12, 6); (9, -1); (7, 11); (11, -5).

Укажите наибольшее целое значение параметра A , при котором для указанных входных данных программа напечатает «YES» три раза.

Решение

Пара чисел, «подходящая» по условию $t > 11$, одна: (5, 12).

Первые числа остальных пар:

-9 2 2 7 7 9 11 12

Ещё два раза слово «YES» будет напечатано при $A = 9$ и при $A = 10$. Максимальное из них и является искомым ответом.

Ответ: 10.

О заданиях 15 ОГЭ по информатике (вариант 15.1)

1. Общие вопросы

В Спецификации контрольных измерительных материалов ОГЭ по информатике в качестве проверяемого результата обучения применительно к варианту 15.1 задания 15 указывается «Умение создавать и выполнять программы для заданного исполнителя».

Темы в федеральном компоненте ГОС ООО – «Алгоритм, свойства алгоритмов, способы записи алгоритмов. Блок-схемы. Представление о программировании. Алгоритмические конструкции. Логические значения, операции, выражения. Разбиение задачи на подзадачи, вспомогательный алгоритм».

Уровень сложности – высокий.

Максимальный балл за задание – 2.

Примерное время выполнения задания (мин.) – 45.

В демонстрационных вариантах контрольных измерительных материалов ОГЭ по информатике нескольких последних лет и в открытом банке заданий ОГЭ представлены задания с исполнителем Робот. В общем виде они такие:

«На бесконечном⁴ клетчатом поле умеет перемещаться Робот. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может. Есть горизонтальные и вертикальные стены. Концы горизонтальных и вертикальных стен могут быть соединены. Длины стен неизвестны. В стенах может быть один проход. Точное место каждого прохода и его ширина неизвестны. Робот находится в некоторой заданной клетке. Требуется написать для Робота алгоритм, закрашивающий те или иные клетки». (Система команд Робота (набор команд, которые можно использовать в алгоритме) указывается в условии.)

Пример

Робот находится в клетке, обозначенной буквой Р (см. рис. 1 слева). Требуется написать для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно ниже горизонтальной стены и левее вертикальной стены, кроме клетки, в которой находится Робот перед выполнением программы. Проходы должны остаться незакрашенными. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого слева рисунка Робот должен закрасить следующие клетки (см. рис. 1 справа). При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное расположение Робота может быть произвольным.

Алгоритм должен решать задачу для любого допустимого расположения стен и любого расположения и размера проходов внутри стен.

Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

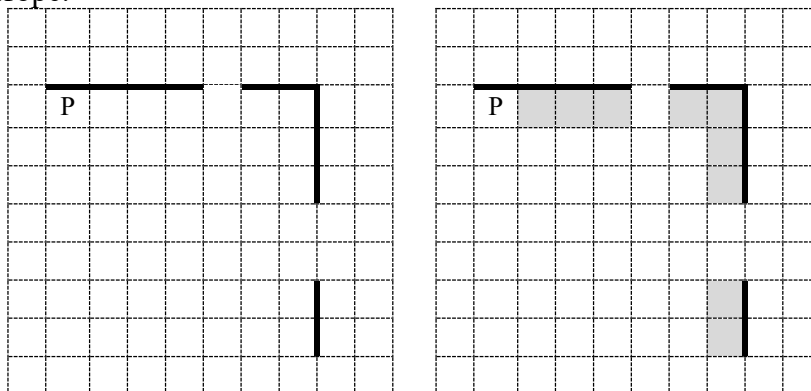


Рис. 1

⁴ Обратите внимание на слово «бесконечном».

Обратим внимание на важный момент – во всех заданиях длины стен, расположение проходов и их ширина и другие размеры неизвестны.

Прежде чем описывать методику разработки алгоритмов таких задач, представим систему команд Робота (она будет приведена в условии задания экзамена).

2. Система команд Робота

У Робота есть девять команд. Пять команд – это команды-приказы. Четыре из них управляют перемещениями Робота:

вверх
вниз
влево
вправо

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится.

Также у Робота есть команда-приказ закрасить, при которой закрашивается клетка, в которой Робот находится в настоящий момент.

Ещё четыре команды – это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

сверху свободно
снизу свободно
слева свободно
справа свободно

В результате выполнения этих команд Робот получает ответ **да** (если в соответствующем направлении от него нет стены) или **нет** (в противном случае).

Команды проверки условий можно использовать вместе с условием **если**, имеющим следующий вид:

если *условие* **то**
последовательность команд
все

Здесь *условие* – одна из команд проверки условия, *последовательность команд* – это одна или несколько любых команд-приказов. Если в результате выполнения команды проверки условия будет получен ответ **да**, то будет выполнена указанная последовательность команд. В противном случае (будет получен ответ **нет**) – никаких действий не будет.

Например, для передвижения на одну клетку вправо, если справа нет стенки, и закрашивания клетки можно использовать такой алгоритм:

если справа свободно **то**
вправо
закрасить
все

В одном условии можно использовать несколько команд проверки условий, 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

```

Последовательности команд цикла, записываемых между служебными словами **нц** и **кц**, называют «тело цикла».

Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

```

нц пока справа свободно
    вправо
кц

```

В теле цикла можно использовать любые команды, в том числе команды проверки условий, например:

```

нц пока справа свободно
    если сверху свободно то
        закрасить
    все
    вправо
кц

```

Вопрос: какая задача решается в последнем алгоритме?

Для выполнения заданий ОГЭ описанного типа надо уметь решать ряд частных, более простых, задач. Их можно объединить в три группы.

3. Группа 1 задач «Задачи на перемещение Робота до ближайшей стены»

Четыре возможных варианта задач приведены на рис. 2.

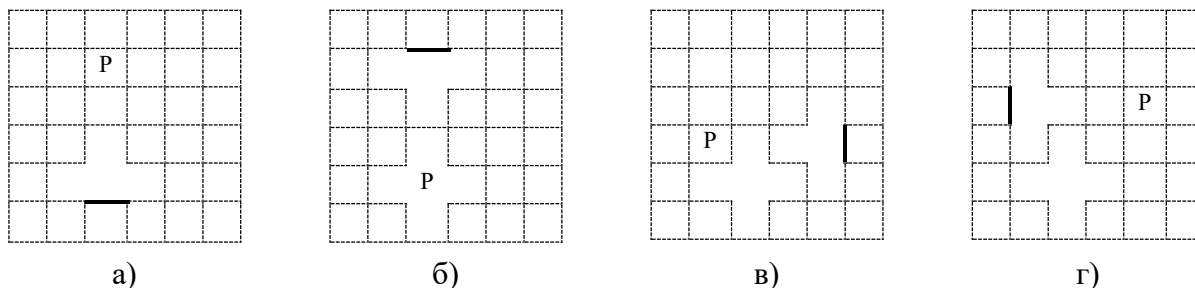


Рис. 2

Пусть надо закрасить все клетки между Роботом и стеной на рис. 2а. Клетку с Роботом закрашивать не нужно.

Решение

Прежде всего, ясно, что понадобится повторение команд, то следует применить в алгоритме цикл **пока** с проверкой условия снизу свободно:

```

нц пока снизу свободно
    . . .
кц

```

Так как клетку с Роботом закрашивать не нужно, то первой командой тела цикла (см. выше) должна быть команда вниз:

```

алг5
нач

```

⁵ В дальнейшем служебные слова **алг**, **нач** и **кон** мы приводить не будем.

```

нц пока снизу свободно
  вниз
  закрасить
кц
кон

```

Можно также сначала сместить Робота вниз, после чего повторять действия, начиная с закрашивания:

```

вниз
нц пока снизу свободно
  закрасить
  вниз
кц

```

Однако при этом клетка со стеной не будет закрашена (почему?!). Её можно закрасить отдельной командой:

```

вниз
нц пока снизу свободно
  закрасить
  вниз
кц
закрасить

```

Конечно, первый вариант алгоритма компактнее.

Задание для самостоятельной работы

Разработайте алгоритмы решения аналогичных задач для обстановок на поле Робота, представленных на рис. 2б–2г. Отдельно разработайте алгоритмы решения задач, в которых исходная клетка с Роботом также должна быть закрашена.

4. Группа 2 задач «Задачи на перемещение Робота до конца стены»

Восемь возможных вариантов приведены на рис.

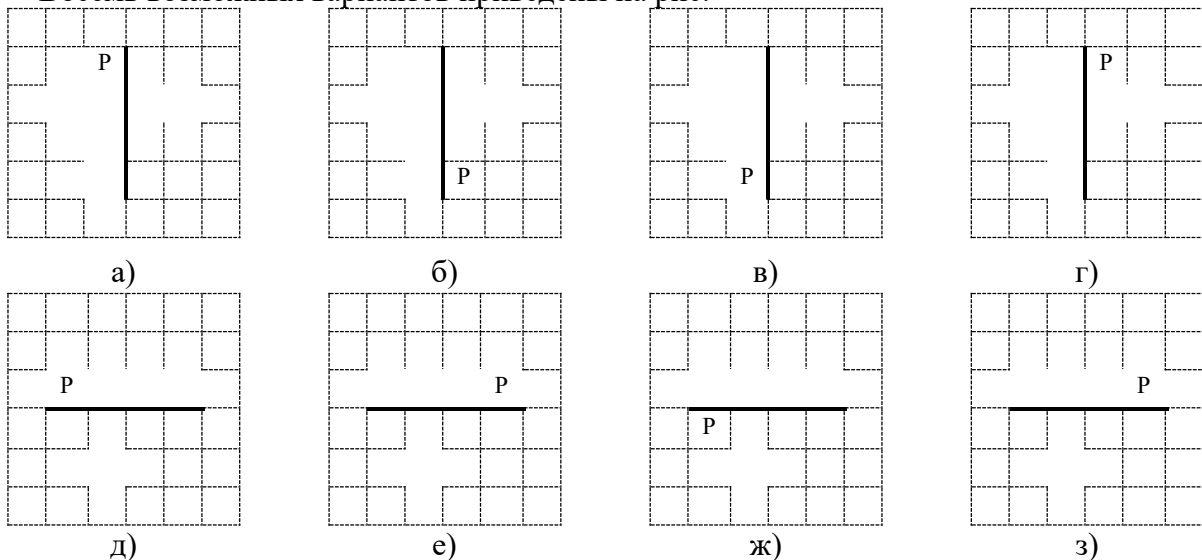


Рис. 3

Пусть надо закрасить все клетки слева от стены на рис. 3а. Клетку с Роботом закрашивать не нужно.

Решение

В данном случае условие перемещения Робота такое: **не** справа свободно. Но алгоритм

нц пока не справа свободно
 вниз
 закрасить
кц

решит задачу неправильно – будет закрашена «лишняя» клетка (убедитесь в этом!).

Правильный вариант – сначала сместить Робот на одну клетку вниз и в теле цикла начинать с закрашивания:

вниз
нц пока не справа свободно
 закрасить
 вниз
кц

Задание для самостоятельной работы

Разработайте алгоритмы решения аналогичных задач для обстановок на поле Робота, представленных на рис. 3б–3з. Отдельно разработайте алгоритмы решения задач, в которых исходная клетка с Роботом также должна быть закрашена.

5. Группа 3 задач «Задачи на перемещение Робота рядом с проходом»

Восемь возможных вариантов приведены на рис. 4.

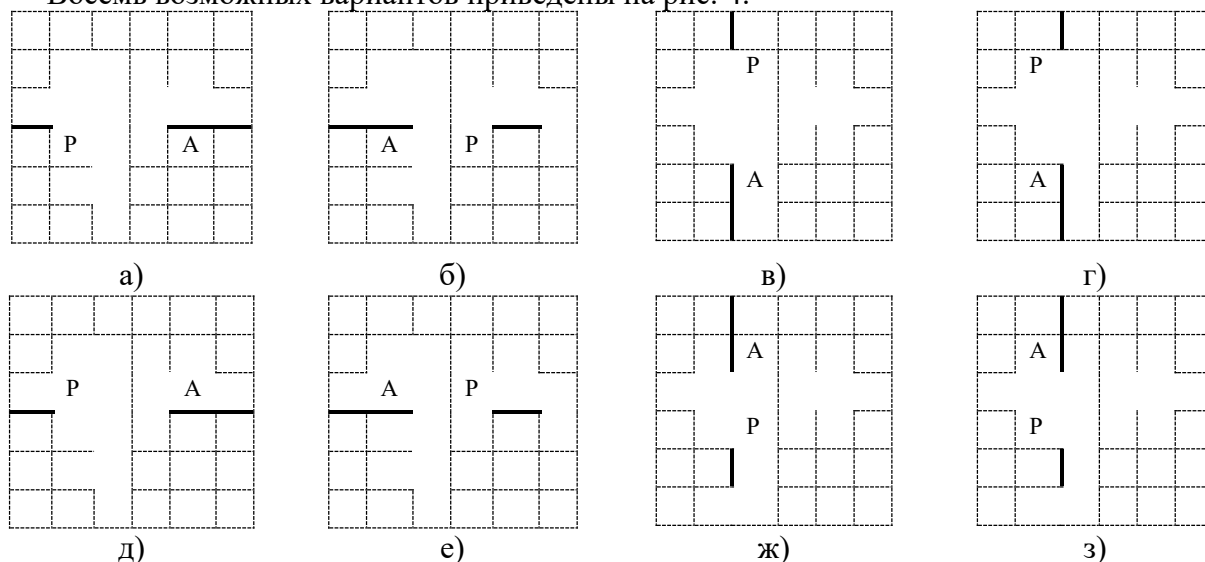


Рис. 4

Во всех случаях следует переместить Робот из клетки с буквой Р в клетку с буквой А. Напомним, что ширина проходов неизвестна. Алгоритмы решения задач разработайте самостоятельно

6. Алгоритм решения задачи применительно к обстановке на рис. 1

В решении задачи можно выделить ряд частных задач (этапов) и соответствующих алгоритмов:

- 1) закрашивание всех клеток под левой верхней стеной, кроме клетки, в которой находится Робот; это алгоритм решения задачи на рис. 3ж;
- 2) пропуск всех клеток под «горизонтальным» проходом; это алгоритм решения задачи на рис. 4а;
- 3) закрашивание всех клеток под правой верхней стеной; при этом клетку в углу можно не закрашивать (она будет закрашена на следующем этапе);

4) закрашивание всех клеток слева от правой верхней стены; это алгоритм решения задачи на рис. 3а, но с закрашиванием и исходной (угловой) клетки;

5) пропуск всех клеток слева от «вертикального» прохода; это алгоритм решения задачи на рис. 4г;

6) закрашивание всех клеток слева от правой нижней стены; это алгоритм решения задачи на рис. 3а, но с закрашиванием и исходной (под проходом) клетки.

Весь алгоритм:

алг

нач

| Этап 1

нц пока не сверху свободно
вправо
закрасить

кц

| Этап 2

нц пока сверху свободно
вправо

кц

| Этап 3

нц пока справа свободно
закрасить
вправо

кц

| Этап 4

нц пока не справа свободно
закрасить
вниз

кц

| Этап 5

нц пока справа свободно
вниз

кц

| Этап 6

нц пока не справа свободно
закрасить
вниз

кц

кон

А нельзя уменьшить размер алгоритма? Оказывается – можно.

Как объединить этапы 1, 2 и 3? Можно сказать, что в них решается задача закрашивания всех клеток между Роботом и вертикальной стеной, кроме клеток под проходом. Алгоритм решения такой задачи может быть оформлен следующим образом:

вправо

нц пока справа свободно
если не сверху свободно **то**
закрасить

все

вправо | Вправо Робот смещается в любом случае

кц

А можно ли объединить этапы 4, 5 и 6. На них решается задача закрашивания всех клеток от угла, в котором в данный момент находится Робот, и ниже, справа от которых есть стена, кроме клеток под проходом. На первый взгляд, можно эту задачу решить так:

```

нц пока снизу свободно
  если не справа свободно то
    закрасить
  все
  вниз | Вниз Робот смещается в любом случае
кц

```

Однако, если поле бесконечное в направлении «вниз», то алгоритм никогда не закончится, что противоречит условию. С учетом этого весь алгоритм решения задачи должен иметь вид:

```

алг
нач
  вправо
  нц пока справа свободно
    если не сверху свободно то
      закрасить
    все
    вправо
  кц
  | Этап 4
  нц пока не справа свободно
    закрасить
  вниз
  кц
  | Этап 5
  нц пока справа свободно
    вниз
  кц
  | Этап 6
  нц пока не справа свободно
    закрасить
  вниз
  кц
кон

```

Как видно, экономия размера алгоритма (а значит, и времени его написания) существенная!

Вопрос: будет ли решена задача при выполнении приведенного алгоритма, если прохода в одной из стен не будет? А если проходов не будет в обеих стенах?

7. Задания для самостоятельной работы

Для обстановок на поле Робота, показанных на рис. 5–9 слева, разработайте алгоритмы решения задач получения ситуаций, показанных на рис. 5–9 справа. При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное расположение Робота может быть произвольным. Алгоритм должен решать задачу для любых допустимых размеров стен и любых размеров проходов внутри стен.

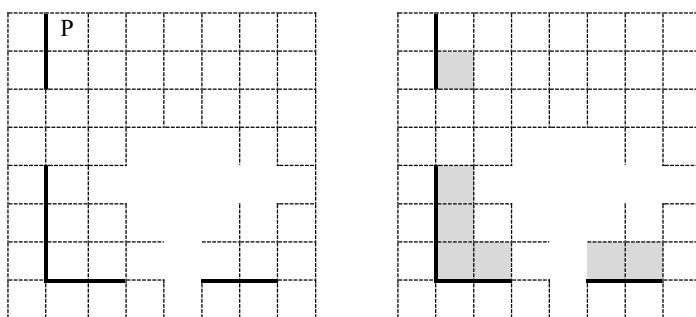


Рис. 5

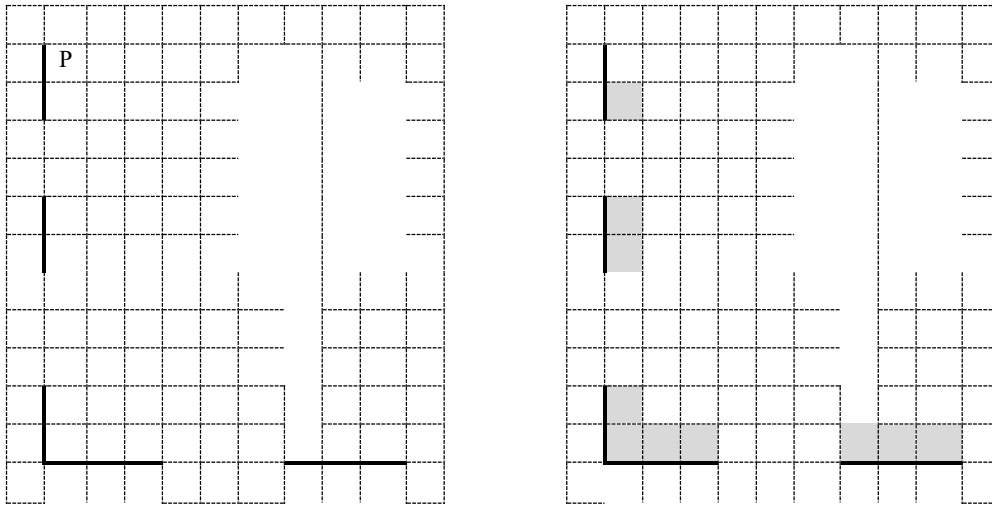


Рис. 6

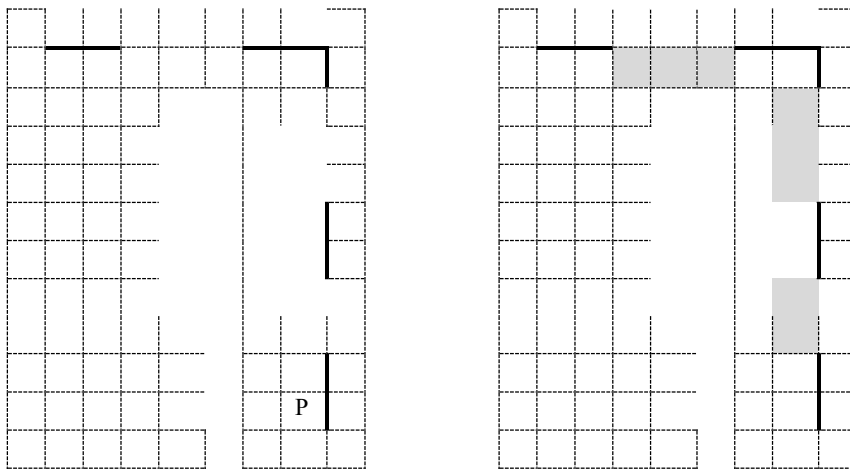


Рис. 7

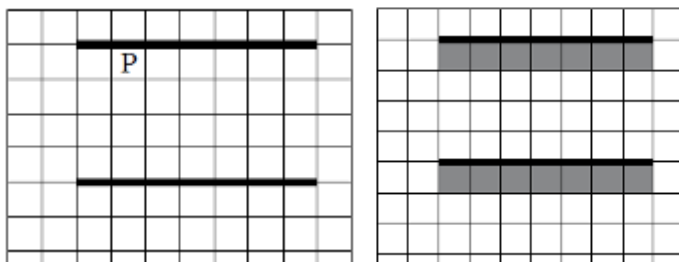


Рис. 8

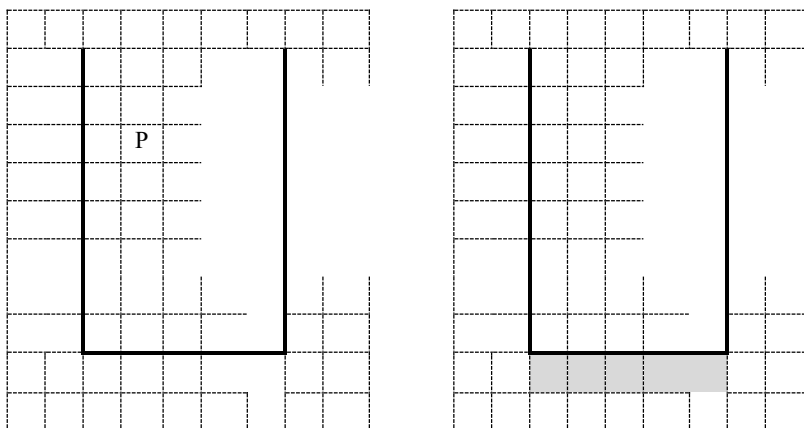


Рис. 9

Итоги конкурса № 65

Напомним, что в качестве задания конкурса было предложено решить три лингвистические задачи, связанные с венгерским языком.

Ответы

В венгерском языке время «3 часа 3 минуты» пишется как «három perc múlt három», время «3 часа 19 минут» – как «négy perc múlt négyed négy».

Победителями конкурса признаны и будут награждены дипломами:

— Бородин Антон и Фаворов Игорь, Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецкая Т.С.**;

— Веселов Иван и Снегирев Максим, Республика Татарстан, Мамадышский р-н, совхоз «Мамадышский», политехнический колледж, преподаватель **Порываева Н.С.**;

— Ковальчук Иван, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Гладков Родион и Шевченко Матвей, Ставропольский край, г. Зеленокумск, школа № 1, учитель **Букина Е.Ю.**;

— Доронина Жанна, Лубков Виктор и Стороженко Степан, г. Пенза, школа № 73, учитель **Диков А.В.**;

— Турков Роман, Владимирская обл., г. Струнино, школа № 10, учитель **Рунов Д.Н.**;

— Яснова Дарья, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**

Поздравляем!

Уважаемые читатели!

В июньском номере нашего журнала возобновится публикация рубрики «Доска почета “Мира информатики”», в которой будут опубликованы фотографии ВСЕХ учащихся, в 2023-2024 учебном году принявших участие в конкурсах и в решении задач и головоломок.

Просим присылать фотографии (можно и не индивидуальные, лучше – неформальные) по адресу mirinfo@infojournal.ru (тема письма: «Доска почёта»). Учащиеся могут присылать фотографии самостоятельно. Срок представления – 31 мая 2024 г.

Ответы и решения задач и головоломок, опубликованных в интернет-журнале «Мир информатики» № 86 (февраль 2024 г.), представили:

— Александрова Кристина и Галкина Анастасия, Владимирская обл., г. Струнино, школа № 11, учитель **Волкова Т.П.**;

— Амосов Максим и Зубрицкая Мария, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Ачылов Степан, Ковальчук Иван, Королёва Кристина, Лозгачев Егор, Мурзина Евгения, Норкулова Ёдгорабону, Печникова Анастасия, Печникова Арина, Смирнова Полина, Саргсян Диана, Сысуев Илья, Трофименко Ирина, Цыплаков Михаил и Фролов Максим, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Бакин Андрей и Джафаров Вагиф, Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецкая Т.С.**;

— Бугаев Иван и Голубева Марина, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Данильчук Алексей, Пусь Иван и Рогов Максим, г. Пенза, школа № 73, учитель **Диков А.В.**;

— Судаков Владислав, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Турков Роман, Владимирская обл., г. Струнино, школа № 10, учитель **Рунов Д.Н.**

СПАСИБО ВСЕМ! МОЛОДЦЫ!