

**Демонстрационный вариант
контрольных измерительных материалов для
проведения в 2020 году итоговой промежуточной
аттестации по информатике в 2020 году (90 минут)**

Задание 1

Среди приведённых ниже трёх чисел, записанных в различных системах счисления, найдите минимальное и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно. Числа: 110000_2 , $E1_{16}$, 151_8 .

Задание 2

Напишите наибольшее целое число x , для которого истинно высказывание: (число <75) **И НЕ** (число чётное).

Задание 3

У исполнителя Альфа две команды, которым присвоены номера:

1. вычти а (a – неизвестное натуральное число; $a \geq 2$)

2. возведи в квадрат

Выполняя первую из них, Альфа уменьшает число на экране на a , а выполняя вторую, возводит число в квадрат. Программа для исполнителя Альфа – это последовательность номеров команд. Известно, что программа 21112 переводит число 4 в число 49.

Определите значение a .

Задание 4

Определите значение переменной a после выполнения алгоритма:

$a := 6;$

$b := 2;$

$b := a/2*b;$

$a := 2*a+3*b;$

В ответе укажите одно целое число – значение переменной a .

Задание 5

Ниже приведена программа, записанная на нескольких языках программирования:

Алгоритмический язык	Паскаль
<u>алг</u> <u>нач</u> <u>цел</u> x, y <u>ввод</u> x <u>ввод</u> y <u>если</u> $x > 5$ и $y < 10$ то вывод “Да” иначе вывод “Нет” <u>все</u> <u>кон</u>	$\text{var } x, y: \text{integer};$ begin $\text{readln } (x);$ $\text{readln } (y);$ $\text{if } (x > 5) \text{ and } (y < 10)$ $\text{then writeln } ('Да')$ $\text{else writeln } ('Нет')$ end.

Было проведено 10 запусков программы, при которых в качестве значений переменных x, y вводились пары чисел: $(-1; 11)$, $(6; 7)$, $(3; -6)$, $(10; 1)$, $(-3; -8)$, $(12; 1)$, $(5; 10)$, $(-5; -10)$, $(13; 9)$, $(7; 14)$. Сколько было запусков, при которых программа напечатала «Да».

Задание 6

Некоторый алгоритм из одной цепочки символов получает новую цепочку следующим образом:

- Сначала вычисляется длина исходной цепочки символов; если она **нечётна**, то удаляется **средний символ цепочки**, а если **чётна**, то **в начало цепочки добавляется символ Л**.

- В полученной цепочке символов каждая буква заменяется буквой, следующей за ней в русском алфавите (А — на Б, Б — на В и т. д., а Я — на А).
- Получившаяся таким образом цепочка является результатом работы алгоритма.

Например, если исходной была цепочка **РУКА**, то результатом работы алгоритма будет цепочка **МСФЛБ**, а если исходной была цепочка **СОН**, то результатом работы алгоритма будет **ТО**.

Дана цепочка символов **РОГ**. Какая цепочка символов получится, если к данной цепочке применить описанный алгоритм дважды (т. е. применить алгоритм к данной цепочке, а затем к результату вновь применить алгоритм)?

Алфавит: АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ.

Задание 7

Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может. У Робота есть девять команд. Четыре команды — это команды-приказы: **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 **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**. Эти команды 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. 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 связи **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

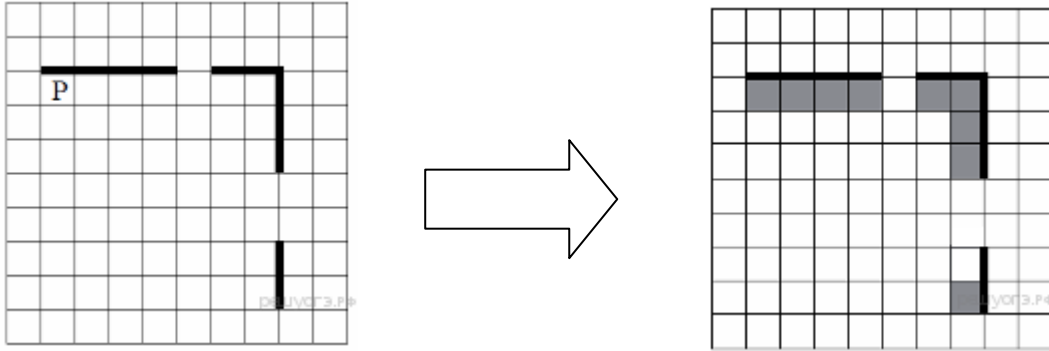
Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

нц пока справа свободно

вправо

кц

Выполните задание



На бесконечном поле есть горизонтальная и вертикальная стены. Правый конец горизонтальной стены соединён с верхним концом вертикальной стены. Длины стен неизвестны. В каждой стене есть ровно один проход, точное место прохода и его ширина неизвестны. Робот находится в клетке, расположенной непосредственно под горизонтальной стеной у её левого 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 ниже горизонтальной стены и левее вертикальной стены, кроме клетки, которая располагается ниже прохода в вертикальной стене. Проходы должны остаться незакрашенными. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок).

При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное положение 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. Сохраните алгоритм в текстовом файле.

Задание 8

Напишите программу, которая в последовательности натуральных чисел определяет максимальное число, кратное 5. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 5. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число — максимальное число, кратное 5.

Пример работы программы:

Входные данные	Выходные данные
3 10 25 12	25