



ЛОГИКА И АЛГОРИТМЫ



СООТВЕТСТВИЕ ЗАДАНИЙ ЕГЭ-2021 И ЕГЭ-2020

<input type="checkbox"/>	ЕГЭ-2021	ЕГЭ-2020	Сложность	Время	Материал
	1	3	Б	3	Анализ информационных моделей (графов)
	2	2	Б	3	Таблицы истинности логических функций
	3	4-1	Б	3	Поиск и сортировка в базах данных
	4	5	Б	2	Кодирование и декодирование
	5	6-1	Б	4	Выполнение и анализ простых алгоритмов
	6	8	Б	4	Анализ программы с циклом
	7	9-1	Б	5	Кодирование растровых изображений
	8	10	Б	4	Кодирование данных, комбинаторика
<input checked="" type="checkbox"/>	9	– (К10)	Б	6	Встроенные функции в электронных таблицах
<input checked="" type="checkbox"/>	10	–	Б	6	Поиск слов в текстовом документе

СООТВЕТСТВИЕ ЗАДАНИЙ ЕГЭ-2021 И ЕГЭ-2020

<input type="checkbox"/>	ЕГЭ-2021	ЕГЭ-2020	Сложность	Время	Материал
	11	13	П	3	Вычисления информационного объёма
	12	14	П	4	Выполнение алгоритмов для исполнителя
	13	15	П	3	Поиск количества путей в графе
	14	16	П	5	Позиционные системы счисления
	15	18	П	5	Основные понятия математической логики.
<input checked="" type="checkbox"/>	16	11 (К11)	П	9	Вычисление значений рекурсивной функции.
<input checked="" type="checkbox"/>	17	К4	П	15	Проверка делимости
<input checked="" type="checkbox"/>	18	–	П	6	Динамическое программирование
	19	26	П	6	Теория игр
	20	26	П	6	Теория игр
	21	26	П	10	Теория игр
	22	20	П	7	Анализ программы с циклами и ветвлениями
	23	22	П	8	Динамическое программирование

РАССМАТРИВАЕМЫЕ ЗАДАНИЯ ОГЭ

<input type="checkbox"/>	ОГЭ-2021	Сложность	Время	Материал
	1	Б	3	Единицы измерения количества информации
	2	Б	4	Кодирование и декодирование информации
	3	Б	3	Логические значения, операции, выражения
	4	Б	3	Моделирование объектов и процессов
	5	Б	6	Алгоритм, свойства алгоритмов, способы записи алгоритмов
	6	Б	4	
	7	Б	3	Сохранение информационных объектов из компьютерных сетей и ссылок на них для индивидуального использования (в том числе из Интернета)
	8	П	5	Поиск информации
	9	П	4	Проектирование и моделирование
	10	Б	3	Единицы измерения количества информации

РАССМАТРИВАЕМЫЕ ЗАДАНИЯ ОГЭ

<input type="checkbox"/>	ОГЭ-2021	Сложность	Время	Материал
	11	Б	6	Поиск информации в файлах и каталогах компьютера
	12	Б	6	Определение количества и информационного объёма файлов, отобранных по некоторому условию
	13	П	25	Создавать презентации (вариант задания 13.1) или создавать текстовый документ (вариант задания 13.2)
	14	В	30	Умение проводить обработку большого массива данных с использованием средств электронной таблицы
	15	В	45	Создавать и выполнять программы для заданного исполнителя (вариант задания 15.1) или на универсальном языке программирования (вариант задания 15.2)

НЕМНОГО ТЕОРИИ

- условные обозначения логических операций

$\neg A, \bar{A}$ не A (отрицание, инверсия)

$A \wedge B, A \cdot B$ A и B (логическое умножение, конъюнкция)

$A \vee B, A + B$ A или B (логическое сложение, дизъюнкция)

$A \rightarrow B$ импликация (следование)

$A \equiv B$ эквивалентность (равносильность)

- операцию «импликация» можно выразить через «ИЛИ» и «НЕ»:

$A \rightarrow B = \neg A \vee B$ или в других обозначениях $A \rightarrow B = \bar{A} + B$

- иногда для упрощения выражений полезны формулы де Моргана:

$\neg (A \wedge B) = \neg A \vee \neg B$ $\overline{A \cdot B} = \bar{A} + \bar{B}$

$\neg (A \vee B) = \neg A \wedge \neg B$ $\overline{A + B} = \bar{A} \cdot \bar{B}$

- если в выражении нет скобок, сначала выполняются все операции «отрицание», затем – «конъюнкция», затем – «дизъюнкция», «импликация», «эквивалентность»
- дизъюнкция $A + B + C + \dots$ равна 0 (выражение ложно) тогда и только тогда, когда все слагаемые одновременно равны нулю, а в остальных случаях равна 1 (выражение истинно)
- конъюнкция $A \cdot B \cdot C \cdot \dots$ равно 1 (выражение истинно) тогда и только тогда, когда все сомножители одновременно равны единице, а в остальных случаях равно 0 (выражение ложно)
- импликация $A \rightarrow B$ равна 0 тогда и только тогда, когда A (посылка) истинна, а B (следствие) ложно
- эквивалентность $A \equiv B$ равна 1 тогда и только тогда, когда оба значения одновременно равны 0 или одновременно равны 1

ЗАДАНИЕ 2

Уровень: базовый

Время: 3 мин

Тема: Анализ таблиц истинности логических выражений.

Что проверяется:

Умение строить таблицы истинности и логические схемы.

Основные способы решения:

- решение логического уравнения
- построение таблицы истинности с помощью Excel

2

Миша заполнял таблицу истинности функции $(x \vee y) \wedge \neg(y \equiv z) \wedge \neg w$, но успел заполнить лишь фрагмент из трёх **различных** её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

				$(x \vee y) \wedge \neg(y \equiv z) \wedge \neg w$
1		1		1
0	1		0	1
	1	1	0	1

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Функция задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имеет следующий вид.

		$\neg x \vee y$
0	1	0

В этом случае первому столбцу соответствует переменная y , а второму столбцу – переменная x . В ответе следует написать yx .

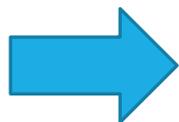
Ответ: _____.

СПОСОБЫ РЕШЕНИЯ ЗАДАНИЯ 2

Решение логического уравнения

$$F = (x \vee y) \wedge \neg(y \equiv z) \wedge \neg w = 1$$

$$\begin{cases} x \vee y = 1 \\ \neg(y \equiv z) = 1 \\ \neg w = 1 \end{cases}$$



$$\begin{cases} x \vee y = 1 \\ y \equiv z = 0 \\ w = 0 \end{cases}$$

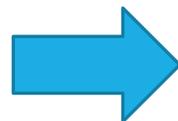


1.
$$\begin{cases} x \vee y = 1 \\ y = 0 \\ z = 1 \\ w = 0 \end{cases}$$



$$\begin{cases} x = 1 \\ y = 0 \\ z = 1 \\ w = 0 \end{cases}$$

2.
$$\begin{cases} x \vee y = 1 \\ y = 1 \\ z = 0 \\ w = 0 \end{cases}$$



$$\begin{cases} x = (0, 1) \\ y = 1 \\ z = 0 \\ w = 0 \end{cases}$$

$$(x, y, z, w) = (1, 0, 1, 0)$$

$$(x, y, z, w) = (0, 1, 0, 0)$$

$$(x, y, z, w) = (1, 1, 0, 0)$$

Ответ: $zyxw$

z	y	x	w	F
1	0	1	0	1
0	1	0	0	1
0	1	1	0	1

СПОСОБЫ РЕШЕНИЯ ЗАДАНИЯ 2

Построение таблицы истинности с помощью Excel

$$F = (x \vee y) \wedge \neg(y \equiv z) \wedge \neg w = 1$$

1. заполняем первую часть таблицы, перечисляя все комбинации переменных в порядке возрастания двоичного кода.
2. для каждой строчки определяем выражения, входящие в логическое произведение, а затем – значение функции.
3. сортируем строки таблицы по столбцу Н по убыванию.
4. удаляем строки, где функция равна 0; можно также скрыть вспомогательные столбцы E, F, G
5. дальше рассуждаем так же, как и при теоретическом решении

1	A	B	C	D	2	A	B	C	D	E	F	G	H
	x	y	z	w		x	y	z	w	x+y	не(y=z)	не(w)	F
1	0	0	0	0	2	0	0	0	0	=ИЛИ(A2;B2)	=НЕ(B2=C2)	=НЕ(D2)	=И(E2;F2;G2)
2	0	0	0	1	3	0	0	0	1	=ИЛИ(A3;B3)	=НЕ(B3=C3)	=НЕ(D3)	=И(E3;F3;G3)
3	0	0	1	0	4	0	0	1	0	=ИЛИ(A4;B4)	=НЕ(B4=C4)	=НЕ(D4)	=И(E4;F4;G4)
4	0	0	1	1	5	0	0	1	1	=ИЛИ(A5;B5)	=НЕ(B5=C5)	=НЕ(D5)	=И(E5;F5;G5)
5	0	1	0	0	6	0	1	0	0	=ИЛИ(A6;B6)	=НЕ(B6=C6)	=НЕ(D6)	=И(E6;F6;G6)
6	0	1	0	1	7	0	1	0	1	=ИЛИ(A7;B7)	=НЕ(B7=C7)	=НЕ(D7)	=И(E7;F7;G7)
7	0	1	1	0	8	0	1	1	0	=ИЛИ(A8;B8)	=НЕ(B8=C8)	=НЕ(D8)	=И(E8;F8;G8)
8	0	1	1	1	9	0	1	1	1	=ИЛИ(A9;B9)	=НЕ(B9=C9)	=НЕ(D9)	=И(E9;F9;G9)
9	1	0	0	0	10	1	0	0	0				
10	1	0	0	1	11	1	0	0	1				
11	1	0	1	0	12	1	0	1	0				
12	1	0	1	1	13	1	0	1	1				
13	1	1	0	0	14	1	1	0	0				
14	1	1	0	1	15	1	1	0	1				
15	1	1	1	0	16	1	1	1	0				
16	1	1	1	1	17	1	1	1	1				

3-4	A	B	C	D	E	F	G	H
	x	y	z	w	x+y	не(y=z)	не(w)	F
1	0	1	0	0	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА
2	1	0	1	0	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА
3	1	1	0	0	ИСТИНА	ИСТИНА	ИСТИНА	ИСТИНА

Ответ: zyxw

ЗАДАНИЕ 2 (ПРИМЕР)

P-15. Логическая функция F задаётся выражением $(x \vee \neg y \vee \neg z) \wedge (\neg x \vee y)$. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z ?

	?	?	?	F
1	0	0	0	1
2	0	0	1	0
3	0	1	0	1
4	0	1	1	1
5	1	0	0	1
6	1	0	1	0
7	1	1	0	0
8	1	1	1	1

В ответе напишите буквы x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала – буква, соответствующая 1-му столбцу; затем – буква, соответствующая 2-му столбцу; затем – буква, соответствующая 3-му столбцу). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

СПОСОБЫ РЕШЕНИЯ ЗАДАНИЯ 2

Решение логического уравнения

$$F = (x \vee \neg y \vee \neg z) \wedge (\neg x \vee y) = 1$$

$$\begin{cases} x \vee \neg y \vee \neg z = 1 \\ \neg x \vee y = 1 \end{cases}$$

$$1) \begin{cases} x \vee \neg y \vee \neg z = 1 \\ x = 0 \\ y = 1 \end{cases}$$

$$2) \begin{cases} x \vee \neg y \vee \neg z = 1 \\ x = 0 \\ y = 0 \end{cases}$$

$$3) \begin{cases} x \vee \neg y \vee \neg z = 1 \\ x = 1 \\ y = 1 \end{cases}$$

$$\begin{cases} 0 \vee 0 \vee \neg z = 1 & z = 0 \\ x = 0 \\ y = 1 \end{cases}$$

$$(x, y, z) = (0, 1, 0)$$

$$\begin{cases} 0 \vee 1 \vee \neg z = 1 & z = (0, 1) \\ x = 0 \\ y = 0 \end{cases}$$

$$(x, y, z) = (0, 0, 0); (0, 0, 1)$$

$$\begin{cases} 1 \vee 0 \vee \neg z = 1 & z = (0, 1) \\ x = 1 \\ y = 1 \end{cases}$$

$$(x, y, z) = (1, 1, 1); (1, 1, 0)$$

Ответ: zyx

	?	?	?	F
3	0	1	0	1
4	0	1	1	1
5	1	0	0	1

СПОСОБЫ РЕШЕНИЯ ЗАДАНИЯ 2

Построение таблицы истинности с помощью Excel

1. заполняем первую часть таблицы, перечисляя все комбинации переменных в порядке возрастания двоичного кода.
2. для каждой строчки определяем выражения, входящие в логическое произведение, а затем – значение функции.
3. делаем сложную сортировку сначала по столбцу C, затем по столбцу B по возрастанию.
4. можно скрыть вспомогательные столбцы D, E
5. Получаем таблицу идентичную в задании

1

	A	B	C
1	x	y	z
2	0	0	0
3	0	0	1
4	0	1	0
5	0	1	1
6	1	0	0
7	1	0	1
8	1	1	0
9	1	1	1

2

	A	B	C	D	E	F
1	x	y	z	ИЛИ(x; НЕ(y); НЕ(z))	ИЛИ(НЕ(x); y)	$(x \vee \neg y \vee \neg z) \wedge (\neg x \vee y)$
2	0	0	0	ИСТИНА	ИСТИНА	ИСТИНА
3	0	0	1	ИСТИНА	ИСТИНА	ИСТИНА
4	0	1	0	ИСТИНА	ИСТИНА	ИСТИНА
5	0	1	1	ЛОЖЬ	ИСТИНА	ЛОЖЬ
6	1	0	0	ИСТИНА	ЛОЖЬ	ЛОЖЬ
7	1	0	1	ИСТИНА	ЛОЖЬ	ЛОЖЬ
8	1	1	0	ИСТИНА	ИСТИНА	ИСТИНА
9	1	1	1	ИСТИНА	ИСТИНА	ИСТИНА

3-5

	A	B	C	F
1	x	y	z	$(x \vee \neg y \vee \neg z) \wedge (\neg x \vee y)$
2	0	0	0	ИСТИНА
3	1	0	0	ЛОЖЬ
4	0	1	0	ИСТИНА
5	1	1	0	ИСТИНА
6	0	0	1	ИСТИНА
7	1	0	1	ЛОЖЬ
8	0	1	1	ЛОЖЬ
9	1	1	1	ИСТИНА

F
1
0
1
1
1
0
0
1

ЗАДАНИЕ 15 (ЕГЭ), ЗАДАНИЕ 3 (ОГЭ)

Уровень: повышенный

Время: 5 мин

Тема: Основные понятия математической логики.

Что проверяется:

Знание основных понятий и законов математической логики

Основные способы решения:

- аналитическое
- программное

Виды заданий:

- предикаты **ДЕЛ**(n, m)
- побитовая конъюнкция
- числовая плоскость
- множества

Уровень: базовый

Время: 3 мин

Тема: Основные понятия математической логики.

Что проверяется:

Логические значения, операции, выражения

Основные способы решения:

- аналитическое

Виды заданий:

3

Напишите наименьшее число x , для которого истинно высказывание:
 $(x > 16)$ **И НЕ** (x нечётное).

Ответ: _____

Решение:

$(x > 16)$ и (x чётно) $\Rightarrow x = 18$

ЗАДАНИЕ 15 (ПРЕДИКАТЫ ДЕЛ(N, M))

15

Обозначим через $\text{ДЕЛ}(n, m)$ утверждение «натуральное число n делится без остатка на натуральное число m ».

Для какого наибольшего натурального числа A формула

$$\neg \text{ДЕЛ}(x, A) \rightarrow (\text{ДЕЛ}(x, 6) \rightarrow \neg \text{ДЕЛ}(x, 9))$$

тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

Ответ: _____.

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАНИЯ 15

Решение задач с предикатом ДЕЛ(n, m)

- 1) введём обозначения: $\text{ДЕЛ}(x, A) = A$, $\text{ДЕЛ}(x, 6) = D_6$, $\text{ДЕЛ}(x, 9) = D_9$
- 2) перепишем выражение в виде $\overline{A} \rightarrow (D_6 \rightarrow \overline{D_9}) = 1$
- 3) раскроем импликацию: $A + \overline{D_6} + \overline{D_9} = 1$
- 4) согласно закону де Моргана получим: $A + \overline{D_6 \cdot D_9} = 1$
- 5) сведём выражение к единственной импликации: $D_6 \cdot D_9 \rightarrow A = 1$
- 6) сформулируем правило, которое мы получили: если значение x делится на 6 и делится на 9, то оно делится на A ;
- 7) если значение x делится на 6 и делится на 9, то оно делится на наименьшее общее кратное $\text{НОК}(6, 9) = 18$, поэтому наибольшее значение A , удовлетворяющее условию, равно 18

Ответ: 18

ПРОГРАММНОЕ РЕШЕНИЕ ЗАДАНИЯ

15

Решение задач с предикатом ДЕЛ(n, m)

1) преобразуем исходную формулу в

$$\text{ДЕЛ}(x, A) \vee \neg (\text{ДЕЛ}(x, 6) \vee \neg \text{ДЕЛ}(x, 9))$$

Так как формула должна быть тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x), то необходимо, чтобы выполнилось хоть одно условие в этом выражении.

- 2) программа проверяет выражение на истинность каждое слагаемое полным перебором для возрастающих значений A ; предполагаем, что наибольшее A не превышает 1000
- 3) если после отработки внутреннего цикла переменная `countX` стала равна 1000, то это говорит о том, что при всех числах X хоть одно из слагаемых будет равно `True`; тогда текущее значение A подходит и записывается в массив `a`
- 4) после работы программы в массиве оказываются значения: [1, 2, 3, 6, 9, 18]

```
a = [] # массив хранения значений A
for A in range(1, 1001):
    countX = 0
    for x in range(1, 1001):
        if (x%A == 0) or not(x%6 == 0) or not(x%9 == 0):
            countX += 1
    if countX == 1000: # все числа X перебрали
        a.append(A)
print(a)
print(max(a))
```

Ответ : 18

ЗАДАНИЕ 15 (ПОБИТОВАЯ КОНЪЮНКЦИЯ)

P-24. Введём выражение $M \& K$, обозначающее поразрядную конъюнкцию M и K (логическое «И» между соответствующими битами двоичной записи). Определите наименьшее натуральное число a , такое что выражение

$$((x \& 28 \neq 0) \vee (x \& 45 \neq 0)) \rightarrow ((x \& 48 = 0) \rightarrow (x \& a \neq 0))$$

тождественно истинно (то есть принимает значение 1 при любом натуральном значении переменной x)?

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАНИЯ 15

Решение задач с поразрядными операциями

1) Введём обозначения:

$$Z_{28} = (x \& 28 = 0), \quad Z_{45} = (x \& 45 = 0), \quad Z_{48} = (x \& 48 = 0), \quad A = (x \& a = 0)$$

2) перепишем исходное выражение и преобразуем его, используя свойство импликации:

$$(\bar{Z}_{28} + \bar{Z}_{45}) \rightarrow (Z_{48} \rightarrow \bar{A}) = \overline{\bar{Z}_{28} + \bar{Z}_{45}} + (Z_{48} \rightarrow \bar{A}) = Z_{28} \cdot Z_{45} + \bar{Z}_{48} + \bar{A}$$

3) перейдем к импликации, используя закон де Моргана:

$$Z_{28} \cdot Z_{45} + \bar{Z}_{48} + \bar{A} = \overline{Z_{48} \cdot A} + Z_{28} \cdot Z_{45} = (Z_{48} \cdot A) \rightarrow Z_{28} \cdot Z_{45}$$

4) преобразуем выражение в правой части по формуле $Z_M = Z_K \text{ or } M$, выполнив поразрядную дизъюнкцию (операцию ИЛИ):

$$28 = 011100$$

$$45 = 101101$$

$$28 \text{ or } 45 = 111101 = 61$$

$$\text{Получаем } (Z_{48} \cdot A) \rightarrow (Z)_{61}$$

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАНИЯ 15

- 5) для того, чтобы выражение (1) было истинно для всех x , нужно, чтобы двоичная запись числа 48 or α содержала все единичные биты числа 61. Таким образом, с помощью числа α нужно добавить те единичные биты числа 61, которых «не хватает» в числе 48:

$$48 = 110000$$

$$\alpha = **11*1$$

$$61 = 111101$$

биты, обозначенные звездочками, могут быть любыми.

- 6) поскольку нас интересует минимальное значение α , все биты, обозначенные звездочкой, можно принять равными нулю.

$$\text{получается } A = 2^3 + 2^2 + 2^0 = 13$$

Ответ: 13.

ПРОГРАММНОЕ РЕШЕНИЕ ЗАДАНИЯ

15

Решение задач с поразрядными операциями

- 1) преобразуем исходную формулу в

$$(\bar{Z}_{28} + \bar{Z}_{45}) \rightarrow (Z_{48} \rightarrow \bar{A}) = \overline{\bar{Z}_{28} + \bar{Z}_{45}} + (Z_{48} \rightarrow \bar{A}) = Z_{28} \cdot Z_{45} + \bar{Z}_{48} + \bar{A}$$

Так как формула должна быть тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x), то необходимо, чтобы выполнилось хоть одно условие в этом выражении.

- 2) программа проверяет выражение на истинность каждое слагаемое полным перебором для возрастающих значений A ; предполагаем, что A не превышает 1000
- 3) если после отработки внутреннего цикла переменная **countX** стала равна 1000, то это говорит о том, что при всех числах X хоть одно из слагаемых будет равно **True**; тогда текущее значение **A** подходит и записывается в массив **a**
- 4) после работы программы будет выведено число 13

Ответ : 13

ПРОГРАММНОЕ РЕШЕНИЕ ЗАДАНИЯ

15

```
a = [] # массив хранения значений A
for A in range(1,1001):
    countX = 0
    for x in range(1,1001):
        if (x & 28 == 0) and (x & 45 == 0) or (x & 48 != 0) or (x & A != 0):
            countX += 1
    if countX == 1000:#все числа X перебрали
        a.append(A)
print(a)
print(min(a))
|
```

ЗАДАНИЕ 15 (ЧИСЛОВАЯ ПЛОСКОСТЬ)

Р-34. (С.С. Поляков) Укажите наименьшее целое значение A , при котором выражение

$$(5k + 6n > 57) \vee ((k \leq A) \wedge (n < A))$$

истинно для любых целых положительных значений k и n .

особенность этой задачи – «уход» авторов от привычных обозначений переменных, x и y ; поскольку мы будем работать с графиками на плоскости, удобнее всё же вернуться к стандартным переменным x и y

$$(5x + 6y > 57) \vee ((x \leq A) \wedge (y < A))$$

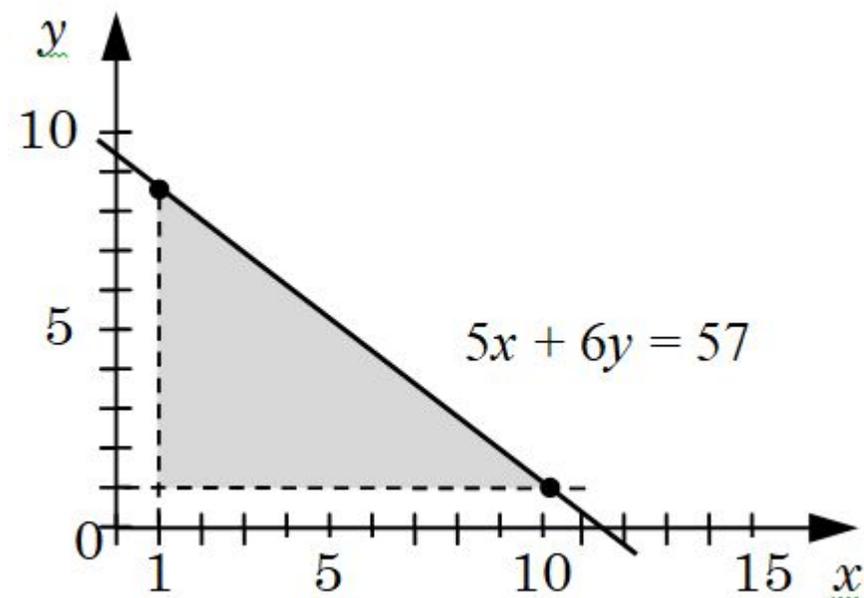
АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАНИЯ 15

- 1) первое выражение ($5x + 6y > 57$) не зависит от выбора A
- 2) таким образом, нам нужно выбрать значение A так, чтобы условие $(x < A)$ and $(y \leq A)$
- 3) выполнялось при всех x и y , для которых ложно $(5x + 6y > 57)$, то есть истинно $(5x + 6y \leq 57)$
- 4) Нужно также учесть, что x и y положительны и добавить ещё два ограничения:

$(x \geq 1)$ and $(y \geq 1)$, таким образом, получаем треугольник, ограниченный линиями

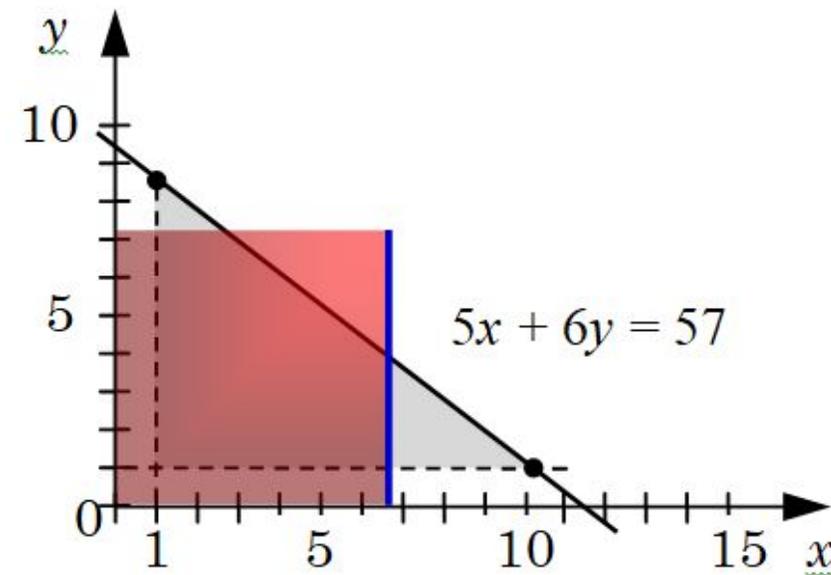
$(5x + 6y = 57)$ and $(x \geq 1)$ and $(y \geq 1)$

- 5) для всех точек этого треугольника с целочисленными координатами должно выполняться условие $(x \leq A) \wedge (y < A)$



АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАНИЯ 15

- 6) это значит, что треугольник (точнее, все его точки с целочисленными координатами) должен оказаться внутри квадрата со стороной A , причем в силу нестрогого неравенства ($x \leq A$) правая граница квадрата (она выделена жирной синей линией) может совпадать с точками треугольника.
- 7) находим точку пересечения прямых $5x + 6y = 57$ и $x = 1$: $y \approx 8,67$; поскольку нужно выполнить условие ($y < A$), получаем $A > 8$
- 8) находим точку пересечения прямых $5x + 6y = 57$ и $y = 1$: $x = 10,2$; поскольку нужно выполнить условие ($x \leq A$), получаем $A \geq 10$
- 9) оба условия нужно выполнить одновременно, поэтому выбираем наиболее жёсткое: $A \geq 10$, что даёт $A_{\min} = 10$.



Ответ: 10.

ПРОГРАММНОЕ РЕШЕНИЕ ЗАДАНИЯ

15

```
a = [] # массив хранения значений A
for A in range(1,101):
    countXY = 0
    for x in range(1,11):
        for y in range(1, 11):
            if (5*x + 6* y > 57) or (x <= A) and (y < A):
                countXY += 1

    if countXY == 100:#все числа X перебрали
        a.append(A)
print(a)
print(min(a))
```

ЗАДАНИЕ 16

Уровень: повышенный

Время: 9 мин

Тема: Рекурсия. Рекурсивные процедуры и функции **Что проверяется:**

Вычисление рекуррентных выражений

1.5.3. *Индуктивное определение объектов.*

1.1.3. *Умение строить информационные модели объектов, систем и процессов в виде алгоритмов.*

Основные способы решения:

- аналитическое
- программное
- с помощью Excel

16

Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$F(n) = 1$ при $n = 1$;

$F(n) = n + F(n - 1)$, если n – чётно,

$F(n) = 2 \times F(n - 2)$, если $n > 1$ и при этом n – нечётно.

Чему равно значение функции $F(26)$?

Ответ: _____.

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАНИЯ 16

Решение (ручной счёт от первого значения):

Начиная вычисления с малых значений n , сразу записываем в таблицу известное значение $F(1) = 1$, затем последовательно вычисляем

$$F(2) = 2 + F(1) = 3 \quad \text{по формуле для чётных } n$$

$$F(3) = 2 \cdot F(1) = 2 \quad \text{по формуле для нечётных } n$$

$$F(4) = \dots$$

...

$$F(26) = 26 + F(25) = 4122$$

Ответ: 4122

n	1	2	3	4	5	6	7	8	9	...	26
F(n)	1	3	2	6	4	10	8	16	16		4122

РЕШЕНИЕ С ИСПОЛЬЗОВАНИЕМ EXCEL И ПРОГРАММЫ

	A	B	C
1	n	F(n)	
2	1	=A2	
3	2	=A3+B2	n четно
4	3	=2*B2	n нечетно

	A	B
1	n	F(n)
2	1	1
3	2	3
4	3	2
5	4	6
6	5	4
7	6	10
8	7	8
9	8	16
10	9	16
11	10	26
12	11	32
13	12	44
14	13	64
15	14	78
16	15	128
17	16	144
18	17	256
19	18	274
20	19	512
21	20	532
22	21	1024
23	22	1046
24	23	2048
25	24	2072
26	25	4096
27	26	4122

```
def F(n):  
    if n == 1:  
        return 1  
    if n % 2 == 0:  
        return n + F(n - 1)  
    else:  
        return 2 * F(n - 2)  
  
print(F(26))
```

ЗАДАНИЯ 19 - 21

Уровень: повышенный

Время: 6 + 6 + 10 мин

Тема: Теория игр. Поиск выигрышной стратегии.

Что проверяется:

Умение анализировать алгоритм логической игры. Умение найти выигрышную стратегию игры. Умение построить дерево игры по

20 Для игры, описанной в предыдущем задании, найдите два таких значения S , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Найденные значения запишите в ответе в порядке возрастания.

Ответ:

19 Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) **один** камень или увеличить количество камней в куче в **два раза**. Например, пусть в одной куче 10 камней, а в другой 5 камней; такую позицию в игре будем обозначать $(10, 5)$. Тогда за один ход можно получить любую из четырёх позиций: $(11, 5)$, $(20, 5)$, $(10, 6)$, $(10, 10)$. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 77. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой в кучах будет 77 или больше камней.

В начальный момент в первой куче было семь камней, во второй куче – S камней; $1 \leq S \leq 69$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии **не следует** включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение S , когда такая ситуация возможна.

Ответ: _____.

21 Для игры, описанной в задании 19, найдите минимальное значение S , при котором одновременно выполняются два условия:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;
- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Ответ: _____.

ЗАДАНИЕ 22 (ЕГЭ), ЗАДАНИЕ 6 (ОГЭ)

уровень: повышенный, базовый

Время: 7 мин, 4 мин

Тема: Анализ программы, содержащей циклы и ветвления.

Что проверяется:

Умение анализировать алгоритм, содержащий ветвление и цикл

Основные способы решения:

- аналитическое
- программное

22

Ниже на четырёх языках программирования записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа: L и M . Укажите наибольшее число x , при вводе которого алгоритм печатает сначала 4, а потом 5.

C++	Python
<pre>#include <iostream> using namespace std; int main() { int x, L, M, Q; cin >> x; Q = 9; L = 0; while (x >= Q){ L = L + 1; x = x - Q; } M = x; if (M < L){ M = L; L = x; } cout <<L <<endl <<M <<endl; return 0; }</pre>	<pre>x = int(input()) Q = 9 L = 0 while x >= Q: L = L + 1 x = x - Q M = x if M < L: M = L L = x print(L) print(M)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач цел x, L, M, Q ввод x Q := 9 L := 0 <u>нц</u> пока x >= Q L := L + 1 x := x - Q <u>кц</u> M := x <u>если</u> M < L <u>то</u> M := L L := x <u>все</u> вывод L, M кон</pre>	<pre>var x, L, M, Q: integer; begin readln(x); Q := 9; L := 0; while x >= Q do begin L := L + 1; x := x - Q; end; M := x; if M < L then begin M := L; L := x; end; writeln(L); writeln(M); end.</pre>

Ответ: _____.

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАНИЯ 22

1) рассмотрим первый цикл:

$$Q = 9$$

$$L = 0$$

while $x \geq Q$:

$$L = L + 1$$

$$x = x - Q$$

- 2) поскольку переменная **L** сначала равна 0 и увеличивается на 1 с каждым шагом цикла, она играет роль счётчика повторения цикла
- 3) на каждой итерации цикла мы вычитаем **Q** из **x** до тех пор, пока **x** не станет меньше **Q**; фактически мы определяем, сколько раз «поместится» **Q** в **x**
- 4) из предыдущих рассуждений следует, что это операция деления, при этом после завершения цикла в переменной **L** находится частное, а в **x** – остаток от деления введённого значения на **Q**

5) рассмотрим строки после цикла:

$$M = x$$

if $M < L$:

$$M = L$$

$$L = x$$

- 6) их роль состоит в том (это легко проверить ручной прокруткой), что значения **M** и **L** меняются местами, если только $M < L$;
- 7) это означает, что значения частного и остатка (сначала **L**, потом **M**) будут выведены в порядке возрастания
- 8) нам нужно определить наибольшее число, при котором частное и остаток равны 4 и 5; для получения именно большего числа нам нужно взять как частное наибольшее из двух заданных чисел то есть 5 (соответственно, за остаток принять 4);
- 9) поскольку делили на 9, искомое число равно $5 \cdot 9 + 4 = 49$

Ответ: 49

ПРОГРАММНОЕ РЕШЕНИЕ ЗАДАНИЯ 22

1. Написать функцию, в которую заключить текст программы , кроме вывода значений L и M.
2. Функция возвращает TRUE, если $L == 4$ и $M == 5$
3. В основной программе написать цикл:

```
for x in range(1,1001):  
    if LM45(x):  
        print(x)
```

```
def LM45( x ):  
    Q = 9  
    L = 0  
    while x >= Q:  
        L = L + 1  
        x = x - Q  
    M = x  
    if M < L:  
        M = L  
        L = x  
    return L == 4 and M == 5  
  
for x in range(1,1001):  
    if LM45(x):  
        print(x)
```

Вывод программы: `=====
41
49`

Ответ: 49

РЕШЕНИЕ ЗАДАНИЯ €

Найти пары, у которых по крайней мере одно из значений >10 .

Таких пар 5: (11, 2), (1, 12), (11, 12), (-11, 12), (-12, 11)

Ответ: 5

6 Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль
<pre>алг нач цел s, t ввод s ввод t ввод A 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); readln(A); if (s > 10) or (t > 10) then writeln("YES") else writeln("NO") end.</pre>
Бейсик	Python
<pre>DIM s, t AS INTEGER INPUT s INPUT t INPUT A IF s > 10 OR t > 10 THEN PRINT "YES" ELSE PRINT "NO" ENDIF</pre>	<pre>s = int(input()) t = int(input()) A = int(input()) if (s > 10) or (t > 10): print("YES") else: print("NO")</pre>
C++	
<pre>#include <iostream> using namespace std; int main() { int s, t; cin >> s; cin >> t; cin >> A; if (s > 10) or (t > 10) cout << "YES" << endl; else cout << "NO" << endl; return 0; }</pre>	

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»?

Ответ: _____.

ЗАДАНИЕ 23 (ЕГЭ), ЗАДАНИЕ 5 (ОГЭ)

уровень: повышенный, базовый

Время: 8 мин, 6 мин

Тема: Динамическое программирование.

Что проверяется:

Умение анализировать результат исполнения алгоритма

Основные способы решения:

- аналитическое
- программное
- с помощью Excel

23

Исполнитель преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1
2. Умножить на 2

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для исполнителя – это последовательность команд.

Сколько существует программ, для которых при исходном числе 1 результатом является число 20, и при этом траектория вычислений содержит число 10?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

Ответ: _____.

5

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 экране 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.

Ответ: _____.

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАНИЯ 23

Искомое количество программ равно произведению количества программ, получающих из числа 1 число 10, на количество программ, получающих из числа 10 число 20.

Пусть $R(n)$ — количество программ, которые число 1 преобразуют в число n , а $P(n)$ — количество программ, которые число 10 преобразуют в число n .

1. Если n не делится на 2, то тогда

$$R(n) = R(n - 1).$$

Аналогично $P(n) = P(n - 1)$

2. Если n делится на 2, тогда

$$R(n) = R(n - 1) + R(n / 2).$$

Аналогично $P(n) = P(n - 1) + P(n / 2)$

$$R(1) = 1$$

$$R(2) = R(1) + R(2/2) = 1 + 1 = 2$$

$$R(3) = R(2) = 2$$

$$R(4) = R(3) + R(4/2) = 2 + 2 = 4$$

$$R(5) = R(4) = 4$$

$$R(6) = R(5) + R(6/2) = 4 + 2 = 6$$

$$R(7) = R(6) = 6$$

$$R(8) = R(7) + R(8/2) = 6 + 4 = 10$$

$$R(9) = R(8) = 10$$

$$R(10) = R(9) + R(10/2) = 10 + 4 = 14$$

$$P(10) = 1$$

$$P(11) = P(10) = 1$$

$$P(12) = P(11) + P(12/2) = 1 + 0 = 1$$

ПРОГРАММНОЕ РЕШЕНИЕ ЗАДАНИЯ

23

1) создаём массив

```
T = 20
```

```
K = [0]*(T+1)
```

константа **T** означает наибольшее число, которое нужно получить; размер массива на единицу больше, элемент **K[0]** мы использовать не будем, так удобнее (ведь нас интересуют числа, начиная с 1)

2) записываем в первый элемент единицу: **K[1] = 1**

3) функция, которая заполняет по приведённым выше формулам элементы массива с индексами от **s+1** до **n** (элемент **K[s]** должен быть уже заполнен!)

```
def fillArr(s, n):
```

```
    for i in range(s+1, n+1):
```

```
        K[i] = K[i-1]
```

```
        if i % 2 == 0 and i // 2 >= s:
```

```
            K[i] += K[i//2]
```

в условие добавилась вторая часть (**i//2 >= s**) для того, чтобы не захватывать значения **K[i]** при **i < s**

4) основная программа заполняет две части массива и выводит результат:

```
fillArr( 1, 10 )
```

```
fillArr( 10, 20 )
```

```
print( K[T] )
```

обратите внимание, что после первого вызова процедуры **fillArr** значение **K[10]** уже определено, так что всё готово для заполнения второй части

Ответ: 28.

```
T = 20
K = [0]*(T + 1)
K[1]=1
def fillArr(s, n):
    for i in range(s+1, n+1):
        K[i] = K[i-1]
        if i % 2 == 0 and i//2 >= s:
            K[i] += K[i//2]

fillArr(1,10)
fillArr(10,20)
print(K)
print(K[T])
|
```

```
[0, 1, 2, 2, 4, 4, 6, 6, 10, 10, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 28]
28
>>> |
```

РЕШЕНИЕ ЗАДАНИЯ 23 С ПОМОЩЬЮ EXCEL

	A	B	C	D	E	F	G	H	I	J	K	L
1	n	1	2	3	4	5	6	7	8	9	10	
2	R(n)	1	2	2	4	4	6	6	10	10	14	
3	R(n/2)		1	#Н/Д	2	#Н/Д	2	#Н/Д	4	#Н/Д	4	
4			ЛОЖЬ	ИСТИНА	ЛОЖЬ	ИСТИНА	ЛОЖЬ	ИСТИНА	ЛОЖЬ	ИСТИНА	ЛОЖЬ	
5	R(n/2)		1	0	2	0	2	0	4	0	4	
6												
7	n	10	11	12	13	14	15	16	17	18	19	20
8	P(n)	1	1	1	1	1	1	1	1	1	1	2
9	P(n/2)		#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д	1
10			#####	ИСТИНА	ЛОЖЬ							
11	P(n/2)		0	0	0	0	0	0	0	0	0	1
12												
13	Ответ		28									

	A	B	C
1	n	1	2
2	R(n)	1	=ЕСЛИ(ОСТАТ(C1;2) = 0;B2+C5;B2)
3	R(n/2)		=СМЕЩ(\$A\$2;0;ПОИСКПОЗ(C1/2;\$B\$1:\$K\$1;0))
4			=ЕОШИБКА(C3)
5	R(n/2)		=ЕСЛИ(C4;0;C3)
6			
7	n	10	11
8	P(n)	1	=ЕСЛИ(ОСТАТ(C7;2) = 0;B8+C11;B8)
9	P(n/2)		=СМЕЩ(\$A\$8;0;ПОИСКПОЗ(C7/2;\$B\$7:\$L\$7;0))
10			=ЕОШИБКА(C9)
11	P(n/2)		=ЕСЛИ(C10;0;C9)
12			
13	Ответ	=K2*L8	

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАНИЯ 5

Проведем анализ программы:
11211

Запишем траекторию вычисления программы из числа 6.

1	1	2	1	1
7	8		81	82

Значит искомое число $b = 10$, т.к.
 $8 * 10 = 80$

Ответ: 10

5

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 экране 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.

Ответ: _____.

ЗАДАНИЕ 15 (ОГЭ)

Уровень: высокий

Время: 45 мин

Тема: Программирование

Что проверяется:

Создавать и выполнять программы для заданного исполнителя (вариант задания 15.1) или на универсальном языке программирования (вариант задания 15.2)

ЗАДАНИЕ 15.1

15.1 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. 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

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 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

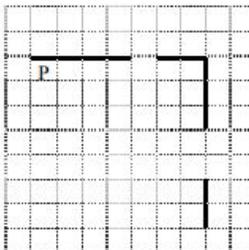
Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

нц пока справа свободно
вправо
кц

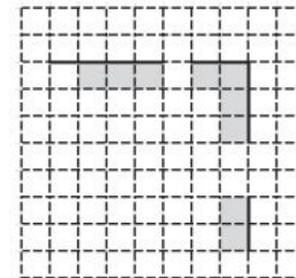
Выполните задание.

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, в которой находится Робот перед выполнением программы. 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 выполнении 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 файле. Название файла и каталог для сохранения Вам сообщает организаторы экзамена.

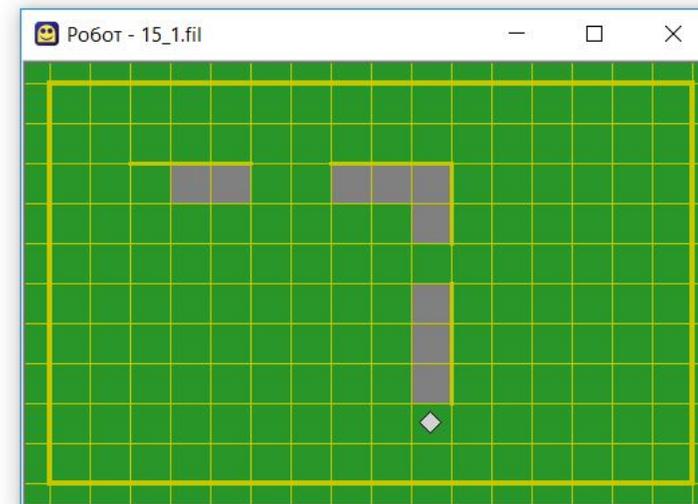
РЕШЕНИЕ ЗАДАНИЯ 15.1

Алгоритм

```
#Пропускаем клетку, в которой стоит Робот.  
вправо  
#Двигаемся вправо, пока не дойдём до прохода в горизонтальной стене.  
#Закрашиваем пройденные клетки.  
нц пока не сверху свободно  
  закрасить  
  вправо  
кц  
#Двигаемся дальше до горизонтальной стены.  
нц пока сверху свободно  
  вправо  
кц  
#Двигаемся вправо, пока не дойдём до вертикальной стены.  
#Закрашиваем пройденные клетки.  
нц пока справа свободно  
  закрасить  
  вправо  
кц  
#Двигаемся вниз, пока не дойдём до прохода в вертикальной стене.  
#Закрашиваем пройденные клетки.  
нц пока не справа свободно  
  закрасить  
  вниз  
кц  
#Двигаемся дальше до вертикальной стены.  
нц пока справа свободно  
  вниз  
кц  
#Двигаемся вниз, до конца вертикальной стены.  
#Закрашиваем пройденные клетки.  
нц пока не справа свободно  
  закрасить  
  вниз  
кц
```

Программа в исполнителе Кумир

```
1 использовать Робот  
2 алг  
3 нач  
4 - вправо  
5 - нц пока не сверху свободно  
6 - - закрасить  
7 - - вправо  
8 - кц  
9 - нц пока сверху свободно  
10 - - вправо  
11 - кц  
12 - нц пока справа свободно  
13 - - закрасить  
14 - - вправо  
15 - кц  
16 - нц пока не справа свободно  
17 - - закрасить  
18 - - вниз  
19 - кц  
20 - нц пока справа свободно  
21 - - вниз  
22 - кц  
23 - нц пока не справа свободно  
24 - - закрасить  
25 - - вниз  
26 - кц  
27 кон
```



ЗАДАНИЕ 15.2

15.2 Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 4, но не кратных 7. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 4 и не кратное 7. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число: количество чисел, кратных 4, но не кратных 7.

Пример работы программы:

Входные данные	Выходные данные
4 16 28 26 24	2

РЕШЕНИЕ ЗАДАНИЯ 15.2

Паскаль

```
var
n, i, a, k: integer;
begin
  readln(n);
  k := 0;
  for i := 1 to n do begin
    readln(a);
    if (a mod 4 = 0) and (a mod 7 <> 0) then
      k := k + 1;
  end;
  writeln(k)
end.
```

Python

```
n = int(input())
k=0
for l in range(n):
    a = int(input())
    if a % 4 == 0 and a % 7 != 0:
        k += 1
print(k)
```