

Вариант № 724515

При выполнении заданий 1—3 является одна цифра, которая соответствует номеру правильного ответа.

Ответом на каждое из заданий 4—23 является число, последовательность букв или цифр. Ответ следует записывать без пробелов и каких-либо дополнительных символов.

Ответы на задания 24—27 в текстовом виде.

В 1. Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. Вот этот код: А–10, Б–001, В–0001, Г–110, Д–111.

Можно ли сократить для одной из букв длину кодового слова так, чтобы код по-прежнему можно было декодировать однозначно? Коды остальных букв меняться не должны. Выберите правильный вариант ответа.

- 1) это невозможно
- 2) для буквы В – 000
- 3) для буквы Б – 0
- 4) для буквы Г – 11

Пояснение.

Мы видим, что выполняется условие Фано: никакое кодовое слово не является началом другого кодового слова, поэтому однозначно можем раскодировать сообщение с начала.

Чтобы сократить код одной буквы, необходимо выполнение условия Фано в новом коде.

Вариант 3 не подходит, потому что 0 является началом кода 0001.

Вариант 4 не подходит, потому что код 1 является началом кода 111.

Вариант 2 подходит, так как не нарушает условия Фано.

Правильный ответ указан под номером 2.

Ответ: 2

В 2. Дан фрагмент таблицы истинности выражения F:

x1	x2	x3	x4	x5	x6	x7	F
0	1	0	1	1	1	0	0
0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	0

Каким из приведённых ниже выражений может быть F?

- 1) $x1 \wedge x2 \wedge \neg x3 \wedge \neg x4 \wedge x5 \wedge (x6 \vee \neg x7)$
- 2) $x1 \vee x2 \vee \neg x3 \vee \neg x4 \vee x5 \vee (x6 \wedge \neg x7)$

$$3) \neg x_1 \vee \neg x_2 \vee x_3 \vee x_4 \vee \neg x_5 \vee (\neg x_6 \wedge x_7)$$

$$4) \neg x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4 \wedge \neg x_5 \wedge (\neg x_6 \vee x_7)$$

Пояснение.

Проанализируем варианты ответов. Они представляют собой либо конъюнкцию, либо дизъюнкцию данных семи переменных или противоположных к ним (если x_1 — переменная, то противоположная к ней — это $\neg x_1$).

Сначала выясним, является F конъюнкцией или дизъюнкцией.

Каковы бы ни были логические переменные x_1, x_2, \dots, x_7 и отрицания к ним, их дизъюнкция может быть равна 0 только в одном случае — когда все они равны 0. Из таблицы истинности следует, что функция F принимает значение 0 для двух различных наборов переменных и их отрицаний, поэтому F не может быть дизъюнкцией. Тем самым, ответы 2 и 3 не подходят.

Вариант 4 (конъюнкция $(\neg x_6 \vee x_7), \neg x_1, \neg x_2, x_3, x_4, \neg x_5$):

В первой строке данной таблицы значение F равно 0. Это значит, что хотя бы одна переменная из $(\neg x_6 \vee x_7), \neg x_1, \neg x_2, x_3, x_4, \neg x_5$ должна быть равна 0, и такая и есть — это x_3 . Значит, по первой строке вариант 4 удовлетворяет функции F .

В второй строке данной таблицы значение F равно 1. Это значит, что все переменные из $(\neg x_6 \vee x_7), \neg x_1, \neg x_2, x_3, x_4, \neg x_5$ должны быть равны 1. Так как в третьей строке переменные, около которых стоит отрицание, равны 0, а переменные без отрицания равны 1, то по второй строке вариант 4 удовлетворяет функции F .

В третьей строке данной таблицы значение F равно 0. Это значит, что хотя бы одна переменная из $(\neg x_6 \vee x_7), \neg x_1, \neg x_2, x_3, x_4, \neg x_5$ должна быть равна 0, и такая и есть — это x_3 . Значит, по третьей строке вариант 4 удовлетворяет функции F .

Правильный ответ — 4.

Ответ: 4

В 3. Ниже представлены две таблицы из базы данных. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных фамилию и инициалы племянника Степаненко З. А. *Пояснение: племянником считается сын брата или сестры.*

Таблица 1			Таблица 2	
ID	Фамилия_И.О.	Пол	ID_Родителя	ID_Ребенка
28	Леоненко М.Д.	Ж	38	39
38	Грант А.Б	М	58	39
39	Грант Б.А.	М	39	40
40	Грант Б.Б.	М	78	40

48	Степаненко З.А.	Ж	38	48
49	Степаненко Т.Л.	Ж	58	48
50	Степаненко Л.Л.	М	48	49
58	Кривец Д.Н.	Ж	50	49
59	Кривец Р.Д.	М	28	50
60	Колос Е.Л.	Ж	48	60
61	Колос А.Е.	М	50	60
68	Кроха М.Б.	Ж	39	68
78	Волченко Т.Д.	Ж	78	68

- 1) Грант Б. А.
- 2) Кроха М. Б.
- 3) Грант Б. Б.
- 4) Степаненко Л. Л.

Пояснение.

Из первой таблицы ясно, что ID Степаненко З. А. равен 48. Найдем во второй таблице в графе «ID_ребенка» номер Степаненко З. А. Видно, что его родители имеют ID 38 и 58. Дети обладателей этих ID имеют ID 39 и 48. Обладатель ID 39 имеет детей, ID которых 40 и 68. Поскольку мы ищем племянника, проверим пол кадого ID: 40 — М, 68 — Ж. Из первой таблицы находим, что ID 40 соответствует Грант Б. Б.

Правильный ответ указан под номером 3.

Ответ: 3

В 4. Для каждого из перечисленных ниже десятичных чисел построили двоичную запись. Укажите число, двоичная запись которого содержит ровно 3 единицы.

- 1) 1
- 2) 11
- 3) 3
- 4) 33

Пояснение.

Переведем числа в двоичную систему счисления:

1. $1_{10} = 1_2$
2. $11_{10} = 1011_2$
3. $3_{10} = 11_2$
4. $33_{10} = 100001_2$

Правильный ответ указан под номером 2.

Ответ: 2

В 5. Между четырьмя местными аэропортами: ЛУГОВОЕ, ДЯТЛОВО, НИКИТИНО и ОРЕХОВО, ежедневно выполняются авиарейсы. Приведён фрагмент расписания перелётов между ними:

Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
ДЯТЛОВО	ЛУГОВОЕ	10:15	10:55
ЛУГОВОЕ	НИКИТИНО	10:20	11:00
ОРЕХОВО	ЛУГОВОЕ	10:25	12:05
ЛУГОВОЕ	ДЯТЛОВО	10:30	11:15
НИКИТИНО	ЛУГОВОЕ	10:55	11:40
ОРЕХОВО	ДЯТЛОВО	11:10	11:55
ЛУГОВОЕ	ОРЕХОВО	11:50	13:30
ДЯТЛОВО	ОРЕХОВО	12:00	12:50
НИКИТИНО	ОРЕХОВО	12:05	12:55
ОРЕХОВО	НИКИТИНО	12:10	12:55

Путешественник оказался в аэропорту ЛУГОВОЕ в полночь. Определите самое раннее время, когда он может попасть в аэропорт ОРЕХОВО. Считается, что путешественник успевает совершить пересадку в аэропорту, если между временем прилета в этот аэропорт и временем вылета проходит не менее часа.

- 1) 12:05
- 2) 12:50
- 3) 12:55
- 4) 13:30

Пояснение.

Заметим, что есть прямой рейс из аэропорта ЛУГОВОЕ в аэропорт ОРЕХОВО (11-50 — 13-30).

Пересадку в аэропорту ДЯТЛОВО осуществить нельзя: ЛУГОВОЕ-ДЯТЛОВО (10-30 — 11-15). ДЯТЛОВО-ОРЕХОВО (12-00 — 12-50). На пересадку у путешественника меньше часа.

Можно лететь с одной пересадкой: ЛУГОВОЕ-НИКИТИНО (10-20 — 11-00). НИКИТИНО-ОРЕХОВО (12-05 — 12-55). На пересадку у путешественника есть 1 час 5 минут.

Самое раннее время прибытия 12-55 минут.

Следовательно, правильный ответ указан под номером 3.

Ответ: 3

В 6. Исполнитель Робот ходит по клеткам бесконечной вертикальной клетчатой доски, переходя по одной из команд вверх, вниз, вправо, влево в соседнюю клетку в указанном направлении. Робот выполнил следующую программу:

- вниз
- влево
- вниз
- влево
- вверх
- вправо
- вверх

Укажите наименьшее возможное число команд в программе, переводящей Робота из той же начальной клетки в ту же конечную.

Пояснение.

Команда "вниз"(1) компенсирует команду "вверх"(2) и наоборот, а команда "влево"(3) компенсирует команду "вправо"(4).

$$1\ 3\ 1\ 3\ 2\ 4\ 2 \Rightarrow (1)\ 3\ (1)\ 3\ (2)\ 4\ (2) \Rightarrow 3\ 3\ 4 \Rightarrow (3)\ 3\ (4) \Rightarrow 3.$$

Получается, что вся программа робота сводится к одной команде(3) "влево", т. е. нужна одна команда, чтобы перейти из начальной клетки в конечную.

Правильный ответ: 1.

Ответ: 1

В 7. В ячейки диапазона C2:F6 электронной таблицы записаны числа, как показано на рисунке.

	A	B	C	D	E	F
1						
2			1	10	100	1000
3			2	20	200	2000
4			3	30	300	3000
5			4	40	400	4000
6			5	50	500	5000

В ячейке C1 записали формулу =F\$2+\$E3. После этого ячейку C1 скопировали в ячейку A3. Какое число будет показано в ячейке A3?

Примечание: знак \$ используется для обозначения абсолютной адресации.

- 1) 1200
- 2) 34
- 3) 410
- 4) 4002

Пояснение.

При копировании формулы из ячейки B3 в ячейку C1:

F\$2: меняется столбец и не меняется номер строки.

\$E3: столбец не меняется, меняется номер строки.

Номер столбца A меньше номера столбца C на 2. Значит столбец F станет столбцом D.

Номер строки 1 на 2 больше номера строки 3, значит, строка 3 станет строкой 5.

$$\text{Окончательный вид } =D\$2+\$E5=10+400=410.$$

Правильный ответ указан под номером 3.

Ответ: 3

В 8. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бэйсик	Паскаль
<pre>DIM N, S AS INTEGER N = 1 S = 0 WHILE N <= 20 S = S + 33 N = N + 1 WEND PRINT S</pre>	<pre>var n, s: integer; begin n := 1; s := 0; while n <= 20 do begin s := s + 33; n := n + 1 end; write(s) end.</pre>
Си	Алгоритмический
<pre>#include void main() { int n, s; n = 1; s = 0; while (n <= 20) { s = s + 33; n = n + 1; } printf("%d", s); }</pre>	<pre>алг нач цел n, s n := 1 s := 0 нц пока n <= 20 s := s + 33 n := n + 1 кц ВЫВОД s кон</pre>

Пояснение.

Цикл while выполняется до тех пор, пока истинно условие $n \leq 20$, т. е. переменная n определяет, сколько раз выполнится цикл.

Цикл выполнится $\frac{20-1}{1} + 1 = 20$ раз ("+1" потому, что в дроби мы не учитываем то, что при 20 он тоже выполнится). Следовательно, значение s будет равно 660.
 Ответ: 660

В 9. Документ объёмом 8 Мбайт можно передать с одного компьютера на другой двумя способами:

- А) сжать архиватором, передать архив по каналу связи, распаковать;
- Б) передать по каналу связи без использования архиватора. Какой способ быстрее и на сколько, если

- средняя скорость передачи данных по каналу связи составляет 2^{20} бит в секунду,
- объём сжатого архиватором документа равен 12,5% от исходного,
- время, требуемое на сжатие документа. — 14 секунд, на распаковку — 2 секунды?

В ответе напишите букву А, если способ А быстрее или Б, если быстрее способ Б. Сразу после буквы напишите, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Слов «секунд», «сек.», «с» к ответу добавлять не нужно.

Пояснение.

Способ А.

Общее время складывается из времени сжатия, распаковки и передачи. Время передачи t рассчитывается по формуле $t = Q / q$, где Q — объём информации, q — скорость передачи данных.

Найдём сжатый объём: $8 \cdot 0,125 = 1$ Мбайт. Переведём Q из Мбайт в биты: 1 Мбайт = $1 \cdot 2^{20}$ байт = $1 \cdot 2^{23}$ бит.

Найдём общее время: $t = 14 \text{ с} + 2 \text{ с} + 1 \cdot 2^{23} \text{ бит} / 2^{20} \text{ бит/с} = 16 + 1 \cdot 2^3 \text{ с} = 24 \text{ с}$.

Способ Б.

Переведём Q из Мбайт в биты: 8 Мбайт = $8 \cdot 2^{20}$ байт = $8 \cdot 2^{23}$ бит.

Найдём общее время: $t = 8 \cdot 2^{23} \text{ бит} / 2^{20} \text{ бит/с} = 8 \cdot 2^3 \text{ с} = 64 \text{ с}$.

Видно, что способ А быстрее на $64 - 24 = 40$ с.

Ответ: А40

В 10. Для передачи аварийных сигналов договорились использовать специальные цветные сигнальные ракеты, запускаемые последовательно. Одна последовательность ракет — один сигнал; в каком порядке идут цвета — существенно. Какое количество различных сигналов можно передать при помощи запуска ровно четырёх таких сигнальных ракет, если в запасе имеются ракеты четырёх различных цветов (ракет каждого вида неограниченное количество, цвет ракет в последовательности может повторяться)?

Пояснение.

Если в алфавите M символов, то количество всех возможных «слов» (сообщений) длиной N равно $Q = M^N$.

$N=4, M=4$. Следовательно, $Q = 4^4 = 256$.

Ответ: 256

В 11. Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими соотношениями:

$$F(n) = 1 \text{ при } n \leq 2;$$

$$F(n) = F(n - 1) + 2 \times F(n - 2) \text{ при } n > 2.$$

Чему равно значение функции $F(7)$? В ответе запишите только натуральное число.

Пояснение.

Последовательно находим:

$$\begin{aligned}
F(1) &= 1; \\
F(2) &= 1; \\
F(3) &= 1 + 2 = 3; \\
F(4) &= 3 + 2 = 5; \\
F(5) &= 5 + 6 = 11; \\
F(6) &= 11 + 10 = 21; \\
F(7) &= 21 + 22 = 43.
\end{aligned}$$

Таким образом, ответ $F(7) = 43$.

Ответ: 43

В 12. В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 64.128.194.208

Маска: 255.255.224.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса сети и запишите в нужном порядке соответствующие им буквы без использования точек.

A	B	C	D	E	F	G	H
0	64	128	192	194	208	224	255

Пример. Пусть искомый IP-адрес: 192.168.128.0, и дана таблица:

A	B	C	D	E	F	G	H
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: **HBAF**.

Пояснение.

1. Запишем числа маски сети в двоичной системе счисления.

$$255_{10} = 11111111_2$$

$$224_{10} = 11100000_2$$

$$0_{10} = 00000000_2$$

2. Адрес сети получается в результате поразрядной конъюнкции чисел маски и чисел адреса узла (в двоичном коде). Так как конъюнкция 0 с чем-либо всегда равна 0, то на тех местах, где числа маски равны 0, в адресе узла стоит 0. Аналогично, там, где числа маски равны 255, стоит само число, так как конъюнкция 1 с любым числом всегда равна этому числу.

3. Рассмотрим конъюнкцию числа 194 с числом 224.

$$194_{10} = 11000010_2$$

$$224_{10} = 11100000_2$$

Результатом конъюнкции является число $11000000_2 = 192$.

4. Сопоставим варианты ответа получившимся числам: 64, 128, 192, 0.

Таким образом, ответ: BCDA.

Ответ: BCDA

В 13. При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы из 12 буквенного набора А, В, Е, К, М, Н, О, Р, С, Т, У, Х. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируются одинаковым и минимально возможным количеством бит. Кроме пароля для каждого пользователя в системе хранятся дополнительные сведения, для чего отведено 12 байт.

Определите объём памяти, необходимый для хранения сведений о 50 пользователях.

- 1) 900 байт
- 2) 1000 байт
- 3) 1100 байт
- 4) 1200 байт

Пояснение.

Согласно условию, в пароле могут быть использованы 12 символов. Известно, что с помощью N бит можно закодировать 2^N различных вариантов. Поскольку $2^3 < 12 < 2^4$, то для записи каждого из 12 символов необходимо 4 бита.

Для хранения всех 15 символов номера нужно $4 \cdot 15 = 60$ бит, а т. к. для записи используется целое число байт, то берём ближайшее не меньшее значение, кратное восьми: это число $64 = 8 \cdot 8$ бит (8 байт).

Тогда 50 паролей занимают $8 \cdot 50 = 400$ байт. Сведения занимают $50 \cdot 12 = 600$ байт. Итого 1000 байт.

Правильный ответ указан под номером 2.

Ответ: 2

В 14. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости, включает в себя 4 команды-приказа и 4 команды проверки условия.

Команды-приказы:

вверх	вниз	влево	вправо
--------------	-------------	--------------	---------------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх \uparrow , вниз \downarrow , влево \leftarrow , вправо \rightarrow .

Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится, и программа прервётся.

Другие 4 команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
------------------------	-----------------------	-----------------------	------------------------

Цикл

ПОКА *условие*

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*

ТО *команда1*

ИНАЧЕ *команда2*

КОНЕЦ ЕСЛИ

выполняется команда1 (если условие истинно) или *команда2* (если условие ложно).

Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F1)?

1						
2						
3						
4						
5						
6						
	A	B	C	D	E	F

НАЧАЛО

ПОКА **справа свободно** ИЛИ **сверху свободно**

ЕСЛИ **справа свободно**

ТО **вправо**

ИНАЧЕ **вверх**

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

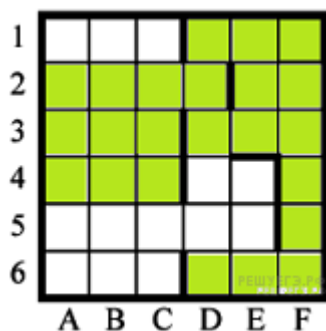
1) 14

2) 18

3) 23

4) 27

Пояснение.



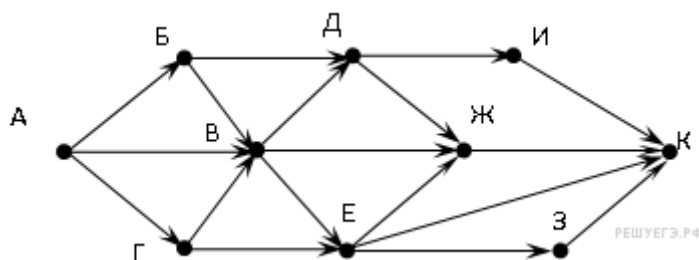
При данной программе РОБОТ поступает следующим образом: сперва РОБОТ проверяет, свободна ли клетка справа или сверху от него. Если это так, то РОБОТ переходит к первому действию внутри цикла. В этом цикле если у правой стороны клетки, в которой находится РОБОТ, нет стены, он движется вправо, в противном случае он перемещается вверх. После этого возвращается к началу внешнего цикла.

Проанализировав эту программу, приходим к выводу, что РОБОТ не может разбиться.

Проверив все клетки по выведенному нами правилу движения РОБОТА, выясняем, что число клеток, удовлетворяющих условию задачи равно 23: вся вторая и третья строчки, D1-F1, A4-C4, D6-F6, F2-F5.

Ответ: 3

В 15. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



Пояснение.

Начнем считать количество путей с конца маршрута – с города К. N_x — количество различных путей из города А в город X, N — общее число путей.

В "К" можно приехать из И, Ж, Е, или З, поэтому $N = N_K = N_I + N_J + N_E + N_Z$ (1)

Аналогично:

$$N_I = N_D;$$

$$N_J = N_D + N_B + N_E;$$

$$N_E = N_B + N_G;$$

$$N_Z = N_E.$$

Добавим еще вершины:

$$N_D = N_B + N_V;$$

$$N_B = N_A + N_B + N_G = 1 + 1 + 1 = 3;$$

$$N_G = N_A = 1;$$

$$N_B = N_A = 1.$$

Преобразуем первые вершины с учетом значений вторых:

$$N_{И} = N_{Д} = N_B + N_B = 1 + 3 = 4;$$

$$N_{Ж} = N_{Д} + N_B + N_E = 4 + 3 + 4 = 11;$$

$$N_E = N_B + N_{Г} = 3 + 1 = 4;$$

$$N_3 = N_E = 4.$$

Подставим в формулу (1):

$$N = N_{Ж} = 4 + 11 + 4 + 4 = 23$$

Ответ: 23

В 16. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 31 оканчивается на 11.

Пояснение.

Так как число в системе счисления с основанием N кончается на 11, то число 31 в десятичной системе счисления при делении на N должно давать остаток 1 (т. е. $31 = Ny + 1$, y — любое целое неотрицательное число, N — основание искомой системы счисления) и частное от этого деления y также должно давать остаток 1 при делении на N (т. е. $y = Nz + 1$, z — любое целое неотрицательное число). Следовательно, $31 = N^2z + 1 + N$, $\Rightarrow 30 = N^2z + N$, где z — неотрицательное целое число, а $N \geq 2$.

Иначе говоря, $30 - N$ должно быть кратным N^2 . Отбросим сразу те N , которые при вычитании из 30 дают простые числа, а также те, квадраты которых больше 30: 1, 6, 7, 8, 9, 10, и так далее до бесконечности. Но при этом 30 тоже является решением данной задачи, так как 30 — особый случай, ведь $30 - N = 0$. Итого остается еще 2, 3, 4 и 5. Из них подходят 2, 3, 5.

Ответ: 2, 3, 5, 30

В 17. В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» — символ «&». В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет:

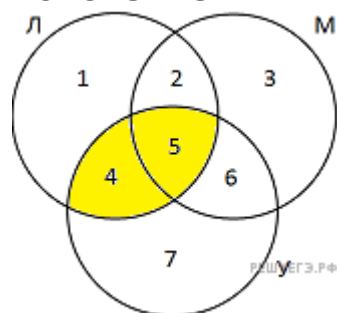
Запрос	Найдено страниц (в тысячах)
Львов & (Ужгород Мукачево)	269
Львов & Мукачево	147
Львов & Ужгород & Мукачево	56

Какое количество страниц (в тыс.) будет найдено по запросу

Львов & Ужгород?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Пояснение.



Обозначим количество запросов в данной области как N_i . Наша цель — $N_4 + N_5$. Тогда из таблицы находим, что:

$$\begin{aligned} N_2 + N_5 &= 147, \\ N_5 &= 56, \\ N_4 + N_5 + N_2 &= 269. \end{aligned}$$

Из первого и второго равенств находим: $N_2 = 91$, из последнего равенства: $N_4 + N_5 = 178$.

Ответ: 178.

Ответ: 178

В 18. На числовой прямой даны два отрезка: $P = [10, 20]$ и $Q = [5, 15]$. Выберите такой отрезок A , что формула

$$((x \in Q) \rightarrow (x \in P)) \wedge (x \in A)$$

тождественно ложна, то есть принимает значение 0 при любом значении переменной x .

- 1) $[0, 6]$
- 2) $[5, 8]$
- 3) $[7, 15]$
- 4) $[12, 20]$

Пояснение.

Логическое И ложно, если ложно хотя бы одно утверждение. Введем обозначения:

$$(x \in A) \equiv A; (x \in P) \equiv P; (x \in Q) \equiv Q.$$

Применив преобразование импликации, получаем:

$$(\neg Q \vee P) \wedge A$$

$P \vee \neg Q$ ложно тогда, когда $x \in [5; 10)$. Выражение A должно быть ложно на интервале $(-\infty, 5); [10, \infty)$. Поскольку все выражение должно быть ложно для ЛЮБОГО

x , следовательно, выражение A должно быть истинно на полуинтервале $[5;10)$ или на любом другом, полностью включающемся в этот интервал.

Из всех отрезков только отрезок $[5;8]$ удовлетворяет этому условию.
Ответ: 2

В 19. В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i := 0 to 10 do
```

```
A[i] := i;
```

```
for i := 1 to 11 do
```

```
A[i-1] := A[11-i];
```

Чему будут равны элементы этого массива?

1) 10 9 8 7 6 5 4 3 2 1 0

2) 11 10 9 8 7 6 5 4 3 2 1

3) 10 9 8 7 6 5 6 7 8 9 10

4) 11 10 9 8 7 6 7 8 9 10 11

Пояснение.

Первый цикл, заполняем массив:

$i=0$: $a[0]=0$,

$i=1$: $a[1]=1$,

$i=2$: $a[2]=2$,

...

$i=10$: $a[10]=10$.

Второй цикл, меняем значения всех элементов: обращаемся к значениям, полученным при заполнении массива.

$i=1$: $a[0]=a[10]=10$,

$i=2$: $a[1]=a[9]=9$,

...

$i=6$: $a[5]=a[5]=5$

$i=7$: $a[6]=a[4]=7$ (Начиная с этого элемента обращаемся к уже изменённым элементам)

$i=11$: $a[10]=a[0]=10$,

Поэтому в результате выполнения программы элементы массива будут равны: 10, 9, 8, 7, 6, 5, 6, 7, 8, 9, 10.

Правильный ответ указан под номером 3.

Ответ: 3

В 20. Ниже на четырёх языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа: a и b . Укажите наименьшее из таких чисел x , при вводе которых алгоритм печатает сначала 14, а потом 5.

Бейсик	Паскаль
DIM X, A, B, C AS INTEGER	var x, a, b, c: integer; begin

<pre> INPUT X A = 0: B = 10 WHILE X > 0 C = X MOD 10 A = A + C IF C < B THEN B = C X = X \ 10 WEND PRINT A PRINT B </pre>	<pre> readln(x); a := 0; b := 10; while x>0 do begin c := x mod 10; a := a+c; if c < b then b := c; x := x div 10; end; writeln(a); write(b); end. </pre>
Си	Алгоритмический
<pre> #include void main() { int x, a, b, c; scanf("%d", &x) ; a = 0; b = 10; while (x>0) { c = x%10; a = a+c; if (c<b) b = c;</b) x = x /10; } printf("%d\n%d", a, b); } </pre>	<pre> алг нач цел x, a, Ъ, c ввод X a := 0; b := 10 нц пока x>0 c := mod(x,10) a := a+c если c<b то b := c все x := div(x,10) кц вывод a, не, b кон </pre>

Пояснение.

Рассмотрим цикл, число шагов которого зависит от изменения переменной x :

```

while x > 0 do begin
...
x:= x div 10;
end;

```

Т. к. оператор `div` оставляет только целую часть от деления, то при делении на 10 это равносильно отсечению последней цифры.

Из приведенного цикла видно, что на каждом шаге от десятичной записи x отсекается последняя цифра до тех пор, пока все цифры не будут отсечены, то есть x не станет равно 0; поэтому цикл выполняется столько раз, сколько цифр в десятичной записи введенного числа.

При этом, переменной c присваивается остаток от деления x на 10, переменная a увеличивается на остаток от деления x на 10. Если остаток от деления x на 10 меньше переменной b , то переменной b присваивается значение переменной c . Поскольку требуется, чтобы программа напечатала сначала 14, необходимо, чтобы сумма цифр числа x была равна 14. Поскольку вторым должно быть напе-

чатано число 5, необходимо, чтобы первой цифрой числа x была цифра 5. Наименьшее такое число — 59.

Ответ: 59

В 21. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для вашего удобства алгоритм представлен на четырёх языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) ENDIF NEXT T PRINT R FUNCTION F(x) F = 16 * (9 - x) * (9 - x) + 127; END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 16 * (9 - x) * (9 - x) + 127; end; begin a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t) < R) then begin M := t; R := F(t) end end; write(R); end. </pre>
Си	Алгоритмический язык
<pre> #include int F(int x) { return 16 * (9 - x) * (9 - x) + 127; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t = a; t <= b; t++){ if (F(t) < R) { M = t; R = F(t); } } printf("%d", R); } </pre>	<pre> алг нач цел a, b, t, R, M a := -20; b := 20 M := a; R := F(a) нц для t от a до b если F(t) < R то M := t; R := F(t) все кц вывод R кон алг цел F(цел x) нач знач := 16 * (9 - x) * (9 - x) + 127 кон </pre>

Пояснение.

1. Алгоритм ищет наименьшее значение функции $F(x)$ на интервале от a до b .

2. Имеем: $F(x) = 16(9-x)(9-x) + 127 = 16x^2 - 288x + 1423$, график этой функции – парабола, ветви которой направлены вверх, поэтому функция принимает наименьшее значение в вершине.

3. Найдем абсциссу вершины:

$$x_{\min} = \frac{-b}{2a} = \frac{-(-288)}{2 \cdot 16} = 9,$$

как видно, она лежит в нужном интервале.

4. Подставим $F(x) = 16 \cdot 9^2 - 288 \cdot 9 + 1423 = 1296 - 2592 + 1423 = 127$.

Ответ: 127

В 22. Определите значение переменной c после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите в виде целого числа.

Бейсик	Паскаль
<pre>a = 30 b = 6 a = a * 3 / b IF a < b THEN c = 4*a - b ELSE c = 4*a + b END IF</pre>	<pre>a := 30; b := 6; a := a * 3 / b; if a < b then c := 4*a - b else c := 4*a + b;</pre>
Си	Алгоритмический язык
<pre>a = 30; b = 6; a = a * 3 / b; if (a < b) c = 4*a - b; else c = 4*a + b;</pre>	<pre>a := 30 a := 30 b := 6 a := a * 3 / b если a < b то c := 4*a - b иначе c := 4*a + b все</pre>

Пояснение.

Выполним программу:

```
a := 30,
b := 6,
a := a * 3 / b = 15.
```

Условие $a < b$ не выполняется, поэтому далее выполним: $c := 4*a + b = 60 + 6 = 66$.

Ответ: 66

В 23. Сколько существует различных наборов значений логических переменных $x_1, x_2, x_3, x_4, x_5, x_6, y_1, y_2, y_3, y_4, y_5, y_6$, которые удовлетворяют всем перечисленным ниже условиям

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_3) \wedge (x_3 \rightarrow x_4) \wedge (x_4 \rightarrow x_5) \wedge (x_5 \rightarrow x_6) = 1$$

$$(y_2 \rightarrow y_1) \wedge (y_3 \rightarrow y_2) \wedge (y_4 \rightarrow y_3) \wedge (y_5 \rightarrow y_4) \wedge (y_6 \rightarrow y_5) = 1$$

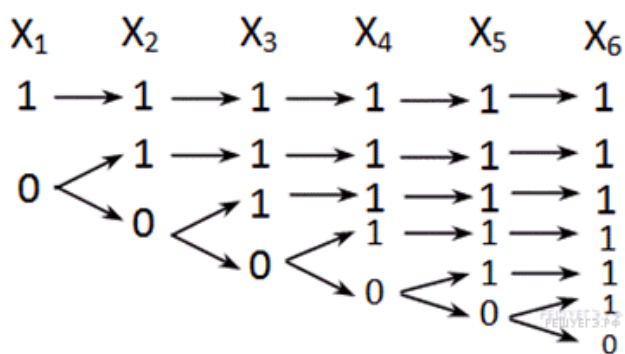
$$y_1 \rightarrow x_1 = 1$$

В ответе не нужно перечислять все различные наборы значений переменных $x_1, x_2, x_3, x_4, x_5, x_6, y_1, y_2, y_3, y_4, y_5, y_6$, при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

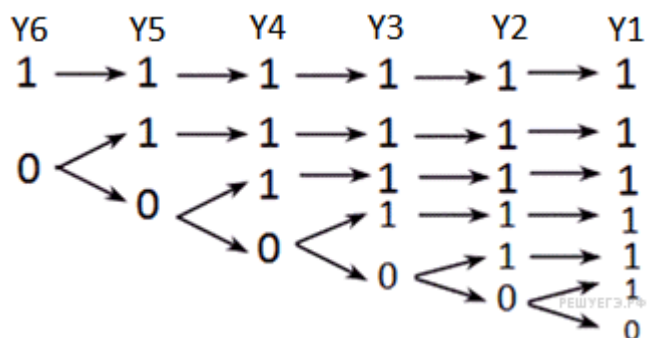
Пояснение.

Заметим, что первое и второе уравнения не связаны между собой ни через какие переменные. Рассмотрим последнее уравнение.

Рассмотрим первое уравнение. Для того, чтобы равенство выполнялось, необходимо, чтобы каждая скобка была истинной. Импликация ложна только тогда, когда посылка истинна, а следствие ложно. Представим решение этого уравнения в виде дерева.



Рассмотрим второе уравнение. Для того, чтобы равенство выполнялось, необходимо, чтобы каждая скобка была истинной. Дерево для решения этого уравнения будет выглядеть так:



Теперь рассмотрим третье уравнение. Импликация ложна только тогда, когда посылка истинна, а следствие ложно. Представим решения последнего уравнения в виде таблицы:

	y_1	x_1
A)	0	0

Б)	0	1
В)	1	1

Случай А). Согласно деревьям решений получим 7 наборов решений для переменных x и 7 наборов переменных y , но $x_1 = 0$ и $y_1 = 0$, следовательно из набора решений для x подойдут 6, а из набора решений для y — 1. Суммарное число наборов: $6 \cdot 1 = 6$.

Случай Б). Аналогично случаю А получим 1 набор переменных x и 1 набор переменных y , суммарное число наборов $1 \cdot 1 = 1$.

Случай В). Аналогично случаю А получим 1 набор переменных x и 6 наборов переменных y , суммарное число наборов $6 \cdot 1 = 6$.

Таким образом, получаем $6 + 1 + 6 = 13$ наборов переменных $x_1, x_2, x_3, x_4, x_5, x_6, y_1, y_2, y_3, y_4, y_5, y_6$, при которых выполнена данная система равенств.

Ответ: 13

24. С 1. Требовалось написать программу, которая решает неравенство $\frac{ax}{x+b} > 0$ относительно x для любого ненулевого числа a и любого неотрицательного числа b и $(a \neq 0, b \geq 0)$, введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var a,b,x: real; begin readln(a,b,x); if b = 0 then write('x > 0 или x < 0') else if a > 0 then write('x > 0 или x <', -b) else write(-b, '< x <0 ') ; end.</pre>	<pre>INPUT a, b, x IF b = 0 THEN PRINT "x > 0 или x <0" ELSE IF a>0 THEN PRINT "x >0 или x<", -b ELSE PRINT -b, "<x<0" ENDIF ENDIF END</pre>	<pre>void main(void) { float a,b,x; scanf ("%f%f%f", &a, &b, &x) ; if (b==0) printf ("x>0 или x<0"); else if (a > 0) printf ("x>0 или x<%f", -b); else printf("%f< x <0", -b); }</pre>

Последовательно выполните три задания:

1) Приведите пример таких чисел a, b, x , при которых программа неверно решает поставленную задачу.

2) Укажите, какая часть программы является лишней.

3) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

Пояснение.

1) $a = -1, b = 0, x = 0$.

Значение x может быть не указано. Значение a может быть любым отрицательным числом. Также допустим ответ, что программа работает неправильно при любых отрицательных a и $b = 0$).

2) Лишняя часть:

не нужно вводить x с клавиатуры; верно: `readln(a, b)`. Также не нужно задавать переменную x при описании переменных; верно `var a,b:real`.

3) Возможная доработка:

```
readln(a, b);
if b = 0 then
if a > 0 then
write('x > 0 или x < 0')
else
write('нет решений')
else
if a > 0 then
write('x > 0 или x <', -b)
else
write(-b, '< x < 0 ');
```

(могут быть и другие способы доработки).

25. С 2. Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от -1000 до 1000. Опишите на русском или на одном из языков программирования алгоритм, который позволяет подсчитать и вывести среднее арифметическое тех элементов массива, которые по своему значению меньше последнего элемента этого массива. Гарантируется, что в исходном массиве есть хотя бы один такой элемент. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

```
Паскаль
Const
N=30;
Var
A: array [1..N] of integer;
I, x, y:integer;
S:real;
Begin
for i:=1 to N do readln(a[i]);
...
End.
```

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Borland Pascal 7.0) или в виде блок-схемы. В этом случае вы должны использовать переменные, аналогичные переменным, используемым в алгоритме, записанном на естественном языке, с учетом синтаксиса и особенностей используемого вами языка программирования.

Пояснение.

```
x:=0;
y:=0;
for i:= 1 to 30 do
if a[i]< a[N] then begin
x:=x+1;
y:=y+a[i];
end;
s:=y/x;
write(s);
```

26. С 3. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу один или два камня или увеличить количество камней в куче в три раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 64. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 64 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 63$.

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.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

а) При каких значениях числа S Петя может выиграть в один ход? Укажите все такие значения.

б) Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите три таких значения S , при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, однако у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах — количество камней в позиции.

Пояснение.

1. а) Петя может выиграть, если $S = 22, \dots, 63$. При меньших значениях S за один ход нельзя получить кучу, в которой больше 63 камней.

б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет $S = 21$ камень. Тогда после первого хода Пети в куче будет 22 камня, или 23 камня, или 63 камня. Во всех случаях Ваня утраивает количество камней и выигрывает первым ходом.

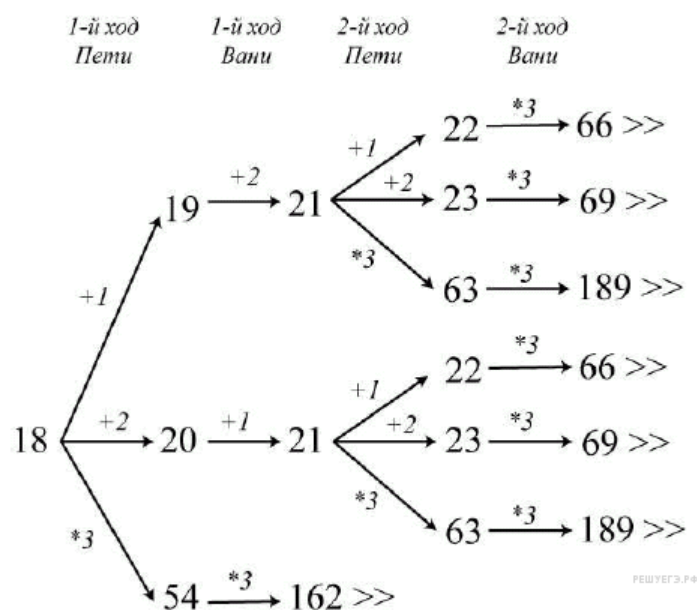
2. Возможные значения S : 19, 20, 7. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 21 камня (при $S = 19$ нужно добавить 2 камня, при $S = 20$ нужно добавить 1 камень, при $S = 7$ нужно

утроить количество камней). Ситуация, когда в куче 21 камень, разобрана в п. 1 б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя) следующим ходом выиграет.

3. Возможное значение S: 18. После первого хода Пети в куче будет 19, 20 или 54 камня. Если в куче станет 54 камня, Ваня утроит количество камней и выиграет первым ходом. Ситуации, когда в куче 19 или 20 камней, уже разобраны в п. 2. В этих ситуациях игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов					
И.п.	1-й ход Пети (разобраны все ходы)	1-й ход Вани (только ход по стратегии)	2-й ход Пети (разобраны все ходы)	2-й ход Вани (только ход по стратегии)	
18	$18 + 1 = \mathbf{19}$	$19 + 2 = \mathbf{21}$	$21 + 1 = \mathbf{22}$	$22 * 3 = \mathbf{66}$	
			$21 + 2 = \mathbf{23}$	$23 * 3 = \mathbf{69}$	
			$21 * 3 = \mathbf{63}$	$63 * 3 = \mathbf{189}$	
	$18 + 2 = \mathbf{20}$	$20 + 1 = \mathbf{21}$	$21 + 1 = \mathbf{22}$	$22 * 3 = \mathbf{66}$	
			$21 + 2 = \mathbf{23}$	$23 * 3 = \mathbf{69}$	
			$21 * 3 = \mathbf{63}$	$63 * 3 = \mathbf{189}$	
	$18 * 3 = \mathbf{54}$	$54 * 3 = \mathbf{162}$			



Дерево всех партий, возможных при Ваниной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

27. С 4. Радиотелескоп пытается получать и анализировать сигналы, поступающие из различных участков космоса, при этом различные шумы переводятся в последовательность целых неотрицательных чисел. Чисел может быть очень много, но не может быть меньше трёх. Все числа различны. Хотя бы одно из чисел нечётно.

В данных, полученных из одного участка, выделяется основное подмножество чисел. Это непустое подмножество чисел (в него могут войти как одно число, так и все поступившие числа), такое, что их сумма чётна и максимальна среди всех возможных непустых подмножеств с чётной суммой. Если таких подмножеств несколько, то из них выбирается то подмножество, которое содержит наименьшее количество элементов.

Вам предлагается написать эффективную, в том числе по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет обрабатывать результаты, приходящие из одного участка, находя основное подмножество. Перед текстом программы кратко опишите используемый Вами алгоритм решения задачи. На вход программе в первой строке подаётся количество сигналов N . В каждой из последующих N строк записано одно целое неотрицательное число, не превышающее 10^9 .

Пример входных данных:

```
5
123
2
1000
0
10
```

Программа должна вывести в порядке возрастания номера сигналов, которые принадлежат основному подмножеству данного участка. Нумерация элементов последовательности ведётся с единицы. *Пример выходных данных для приведённого выше примера входных данных: 2 3 5.*

Пояснение.

Основное подмножество состоит из всех значений сигналов, кроме 0, если он встречается, и кроме минимального нечётного значения, если таких значений нечётное число.

Программа читает все входные данные один раз, не запоминая все входные данные в массиве, размер которого равен N . Во время чтения данных запоминается номер 0, если он встретится (по условию все значения различны, поэтому 0 встречается не больше одного раза), подсчитывается количество нечётных значений и ищется минимальное нечётное значение. После окончания ввода распечатываются все номера, кроме номера 0 и номера минимального нечётного значения, но только в случае, если их количество нечётно.

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая. Ниже приведены примеры решения задания на языках Паскаль и Бейсик. Допускаются решения, записанные на других языках программирования.

Пример правильной и эффективной программы на языке Паскаль:

Пример правильной и эффективной программы на языке Бейсик:

```

var n,i,j,k,c,min,a: longint;
begin
readln(n);
min := 1000000001;
k := 0;
j := 0;
c := 0;
for i := 1 to n do
begin
readln(a);
if a = 0 then j := i;
if a mod 2 <> 0 then
begin
c := c + 1;
if a < min then
begin
min := a;
k := i;
end
end
end;
for i := 1 to n do
if (i <> j) and ((c mod 2 = 0) or (i <>
k)) then
write(i, ' ');
end.

```

```

INPUT n
min = 0
k = 0
j = 0
c = 0
FOR i = 1 TO n
INPUT a
IF a = 0 THEN j = i
IF a MOD 2 <> 0 THEN
c = c + 1
IF (min = 0) OR (a < min) THEN
min = a
k = i
END IF
END IF
NEXT i
FOR i = 1 TO n
IF (i <> j) AND ((c MOD 2 = 0) OR (i <> k))
THEN PRINT i
NEXT i
END

```