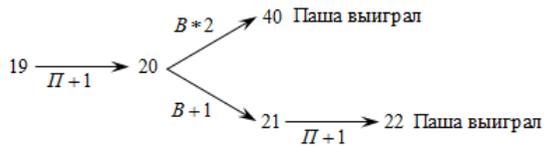


1 вариант		2 вариант		3 вариант		4 вариант	
1.	2	1.	545	1.	13	1.	2
2.	yzx	2.	zxy	2.	zyxw	2.	xzy
3.	14	3.	17	3.	16	3.	15
4.	2659	4.	123	4.	2	4.	3045
5.	18	5.	9	5.	1000	5.	9DA4
6.	13	6.	72	6.	59	6.	171
7.	70	7.	3	7.	3	7.	8
8.	512	8.	28	8.	26	8.	26
9.	30	9.	16	9.	1056	9.	2
10.	24	10.	324	10.	192	10.	5
11.	315	11.	3	11.	63	11.	128
12.	DFDH	12.	CFDH	12.	DECA	12.	240
13.	300	13.	2	13.	189	13.	56
14.	3	14.	152	14.	B5	14.	39
15.	38	15.	12	15.	11	15.	39
16.	1	16.	5	16.	1	16.	5
17.	830	17.	730	17.	20	17.	369
18.	23	18.	16	18.	24	18.	4
19.	4	19.	5	19.	1798	19.	5
20.	118	20.	35	20.	27	20.	1453
21.	9	21.	238	21.	-1	21.	2
22.	26	22.	24	22.	-17	22.	8
23.	22	23.	18	23.	512	23.	0100
24.	<p>Решение использует запись программы на Паскале. Допускается использование программы на других языках.</p> <p>1. При вводе числа 1.8 программа выведет число 1.</p> <p>2. Примеры чисел, при вводе которых программа выводит верный ответ: 0.6, 1.4.</p> <p><i>Примечание для проверяющего. Программа содержит две ошибки, одна из которых приводит к увеличению ответа, другая – к уменьшению.</i></p> <p>В некоторых случаях эти ошибки компенсируют друг друга, и ответ оказывается правильным. Это происходит, если <math>0 &lt; A \leq 1.5</math>.</p> <p>3. Программа содержит две ошибки.</p> <p>1) Неверная инициализация. Начальные значения S и K должны быть одинаковы: оба равны нулю или оба равны единице. В приведённом варианте вычисленная сумма оказывается на 1 больше правильного значения. Эту ошибку можно исправить двумя способами.</p> <p>Первый способ исправления: Строка с ошибкой: s := 1;</p> <p>Правильная строка: s := 0;</p>	<p>24</p> <p>1. При вводе N = 2018 программа выведет число 4.</p> <p>2. Примеры чисел, при вводе которых программа выводит верные ответы: 22 (ответ 4), 204 (ответ 4), 480 (ответ 8), 82 (ответ 10).</p> <p>3. Программа содержит две ошибки.</p> <p>1) Неверная инициализация минимума. Минимальная сумма двух цифр может оказаться больше 10. Начальное значение минимума должно быть не меньше 18.</p> <p>2) Неверно определяется последняя цифра. Вместо неё вычисляется число, которое получается при вычёркивании последней цифры из записи N.</p> <p><b>Пример исправления для языка Паскаль:</b> <b>Первая ошибка:</b> m := 10; Исправленная строка: m := 18; Допускается вместо 18 использовать любое число, которое больше 18.</p> <p><b>Вторая ошибка:</b> d1 := n div 10;</p>	<p>24</p> <p>Решение использует запись программы на Паскале.</p> <p>1. Программа выведет два числа: 2 и 999.</p> <p>2. Пример последовательности, содержащей нечётные числа, для которой программа работает правильно: 1 2 3 999.</p> <p>Замечание. В конце работы программы значение переменной maximum всегда равно 999. Соответственно, программа будет работать верно, если в последовательности есть 999. Выведенное количество нечётных чисел будет правильным в любом случае.</p> <p>3. В программе есть две ошибки.</p> <p>Первая ошибка: неверная инициализация maximum. Строка с ошибкой: maximum := 999; Верное исправление: maximum := 0; Вместо 0 может быть использовано любое число, меньшее или равное 1.</p> <p>Вторая ошибка: неверное присваивание при вычислении максимума. Строка с ошибкой:</p>	<p>24</p> <p>Решение использует запись программы на Паскале. Допускается использование программы на других языках.</p> <p>1. При вводе числа 0.4 программа выведет число 1.</p> <p>2. Программа не выведет ответ 2 ни при каком вводе, то есть ответ на задание: 0.</p> <p>Примечание для проверяющего. При вводе <math>A &lt; 1</math> тело цикла не будет выполнено ни разу и программа выведет ответ 1. При <math>A \geq 1</math> происходит заикливание. 3. Возможные варианты исправления для языка Паскаль следующие.</p> <p>Первая ошибка: while s&lt;=a do begin Исправленная строка: while s &lt; a do begin Вторая ошибка: s := 1.0/k; Исправленная строка: s := s + 1.0/k; В программах на других языках ошибочные строки и их исправления аналогичны.</p>			

<p>Второй способ исправления: Строка с ошибкой: k := 0; Правильная строка: k := 1;</p> <p>2) Неверное определение ответа. Приведённая программа находит не максимальное K, при котором выполняется неравенство, а минимальное, при котором оно не выполняется, то есть увеличивает верное значение на 1. Эту ошибку можно исправить двумя способами: скорректировать значение при выводе или использовать «предварительное вычисление» в условии цикла.</p> <p>Первый способ исправления: Строка с ошибкой: write(k); Правильная строка: write(k-1);</p> <p>Второй способ исправления: Строка с ошибкой: while s &lt; a do begin Правильная строка: while s+1/(k+1) &lt; a do begin</p> <p>Второй способ исправления возможен, только если первая ошибка была исправлена первым способом, то есть использована инициализация нулями.</p> <p>При инициализации единицами минимально возможный ответ оказывается равным 1, в то время как при A &lt; 1 правильный ответ 0. Первый способ исправления работает независимо от того, как исправлена первая ошибка.</p> <p>В программах на других языках ошибочные строки и их исправления аналогичны. При использовании второго способа исправления второй ошибки на языке Си следует обратить внимание на типы данных при делении: чтобы результат получился правильного типа (вещественный), необходимо обеспечить, чтобы хотя бы один из операндов деления был вещественным.</p> <p>Незначительной опiskeй, не влияющей на оценку, следует считать отсутствие знаков после содержательной части исправления.</p>	<p>Исправленная строка: d1 := n mod 10;</p> <p>В программах на других языках ошибочные строки и их исправления аналогичны.</p> <p>Незначительной опiskeй, не влияющей на оценку, следует считать отсутствие знаков после содержательной части исправления (например, точки с запятой в C++).</p>	<p>maximum = i; Верное исправление: maximum = x;</p>	<p>Незначительной опiskeй, не влияющей на оценку, следует считать отсутствие служебных слов и знаков после содержательной части исправления.</p>				
<p>25. Последний символ записи числа в позиционной системе счисления показывает остаток от деления этого числа на основание системы.</p> <p>Для решения задачи необходимо просмотреть все числа в массиве и выбрать минимальное среди тех, которые дают одинаковый остаток при делении на 10 и на 16.</p>	<p>25</p> <table border="1" data-bbox="705 1252 1108 1500"> <thead> <tr> <th>ПАСКАЛЬ</th> <th>БЕЙСИК</th> </tr> </thead> <tbody> <tr> <td> <pre>max:=a [1] ; max2 := a [2] ; if max &lt; max2 then begin max := a [2] ; max2 := a [1] end; for i := 3 to N do</pre> </td> <td> <pre>MAX = A(1) MAX2 = A(2) IF MAX &lt; MAX2 THEN MAX = A(2) MAX2 = A(1) ENDIF FOR I = 3 TO N IF A(I) &gt; MAX THEN</pre> </td> </tr> </tbody> </table>	ПАСКАЛЬ	БЕЙСИК	<pre>max:=a [1] ; max2 := a [2] ; if max &lt; max2 then begin max := a [2] ; max2 := a [1] end; for i := 3 to N do</pre>	<pre>MAX = A(1) MAX2 = A(2) IF MAX &lt; MAX2 THEN MAX = A(2) MAX2 = A(1) ENDIF FOR I = 3 TO N IF A(I) &gt; MAX THEN</pre>	<p>25</p> <p><b>На языке Паскаль</b></p> <pre>min := 1001; for i := 1 to N do if (a[i]&gt;0) and (a[i] mod 5=0) and (a[i] &lt; min) then min := a[i]; if min &lt; 1001 then writeln(min) else writeln('Не найдено');</pre>	<p>25</p> <p>Пример правильной программы на языке Паскаль.</p> <pre>x := 0; s := 0; for i := 1 to n do if a[i] &gt; 3 then begin s := s + a[i]; inc(x);</pre>
ПАСКАЛЬ	БЕЙСИК						
<pre>max:=a [1] ; max2 := a [2] ; if max &lt; max2 then begin max := a [2] ; max2 := a [1] end; for i := 3 to N do</pre>	<pre>MAX = A(1) MAX2 = A(2) IF MAX &lt; MAX2 THEN MAX = A(2) MAX2 = A(1) ENDIF FOR I = 3 TO N IF A(I) &gt; MAX THEN</pre>						

<table border="1" style="width: 100%;"> <tr> <th style="text-align: center;">Паскаль</th> </tr> <tr> <td> <pre>m:=10000; for i:=1 to N do begin   if (a[i] mod 16 = a[i] mod 10) and (a[i] &lt; m)   then m := a[i]; end; writeln(m)</pre> </td> </tr> </table> <p>При использовании Python можно применить специальные средства этого языка, которые позволяют записать все действия в одной строке без явного задания цикла.</p> <table border="1" style="width: 100%;"> <tr> <th style="text-align: center;">Python</th> </tr> <tr> <td> <pre>print(min((m for m in a if m%10 == m%16), default=10000))</pre> </td> </tr> </table> <p>При написании подобной программы очень важно использовать параметр default (допустим, начиная с версии Python 3.4) или какой-то другой способ обработки ситуации, когда в исходных данных нет подходящих элементов.</p>	Паскаль	<pre>m:=10000; for i:=1 to N do begin   if (a[i] mod 16 = a[i] mod 10) and (a[i] &lt; m)   then m := a[i]; end; writeln(m)</pre>	Python	<pre>print(min((m for m in a if m%10 == m%16), default=10000))</pre>	<table border="1" style="width: 100%;"> <tr> <td> <pre>if a[i] &gt; max then begin max2 := max; max := a [i] end else if a[i] &gt; max2 then max2 := a[i]; writeln(max2);</pre> </td> <td> <pre>MAX2 = MAX MAX = A (I) ELSE IF A(I) &gt; MAX2 THEN MAX2 = A(I) ENDIF ENDIF , NEXT I PRINT MAX2</pre> </td> </tr> <tr> <td>СИ</td> <td>Алгоритмический язык</td> </tr> <tr> <td> <pre>max = a [0] ; max2 = a [1] ; if (max &lt; max2) { max = a [1] ; max2 = a [0] ; } for(i = 2; i &lt; N; i++) if(a[i] &gt; max) { max 2 = max; max = a [i]; } else if (a [i] &gt; max2 ) max2 = a [i] ; cout « max2 « endl;</pre> </td> <td> <pre>MAX := a [1] MAX2 := a [2] если MAX &lt; MAX2 то MAX := a [2 ] MAX2 := a [1] все нц для i от 3 до N если a[i]&gt;MAX то MAX2 := MAX MAX := a [i] иначе если a[i]&gt;MAX2 то MAX2 := a [i] все все кц вывод MAX2</pre> </td> </tr> </table>	<pre>if a[i] &gt; max then begin max2 := max; max := a [i] end else if a[i] &gt; max2 then max2 := a[i]; writeln(max2);</pre>	<pre>MAX2 = MAX MAX = A (I) ELSE IF A(I) &gt; MAX2 THEN MAX2 = A(I) ENDIF ENDIF , NEXT I PRINT MAX2</pre>	СИ	Алгоритмический язык	<pre>max = a [0] ; max2 = a [1] ; if (max &lt; max2) { max = a [1] ; max2 = a [0] ; } for(i = 2; i &lt; N; i++) if(a[i] &gt; max) { max 2 = max; max = a [i]; } else if (a [i] &gt; max2 ) max2 = a [i] ; cout « max2 « endl;</pre>	<pre>MAX := a [1] MAX2 := a [2] если MAX &lt; MAX2 то MAX := a [2 ] MAX2 := a [1] все нц для i от 3 до N если a[i]&gt;MAX то MAX2 := MAX MAX := a [i] иначе если a[i]&gt;MAX2 то MAX2 := a [i] все все кц вывод MAX2</pre>	<p><b>На алгоритмическом языке</b></p> <pre>min := 1001 нц для i от 1 до N если a[i]&gt;0 и mod (a[i], 5) = 0 и a[i] &lt; min to min := a[i] все кц если min &lt; 1001 то вывод min иначе вывод "Не найдено" все</pre> <p><b>На языке Бейсик</b></p> <pre>MIN = 1001 FOR I = 1 TO N IF A(I) &gt; 0 AND A(I) MOD 5=0 AND A(I) &lt; MIN THEN MIN = A(I) ENDIF NEXT I IF MIN &lt; 1001 THEN PRINT MIN ELSE PRINT "Не найдено" ENDIF</pre> <p><b>На языке Си</b></p> <pre>min = 1001; for (i = 0; i &lt; N; i++) if (a[i]&gt;0 &amp;&amp; a[i] % 5 == 0 &amp;&amp; a[i] &lt; min) min = a[i]; if { min &lt; 1001) printf ("%d", min); else cout &lt;&lt; "Не найдено";</pre>	<pre>end; write(s / x);</pre>
Паскаль													
<pre>m:=10000; for i:=1 to N do begin   if (a[i] mod 16 = a[i] mod 10) and (a[i] &lt; m)   then m := a[i]; end; writeln(m)</pre>													
Python													
<pre>print(min((m for m in a if m%10 == m%16), default=10000))</pre>													
<pre>if a[i] &gt; max then begin max2 := max; max := a [i] end else if a[i] &gt; max2 then max2 := a[i]; writeln(max2);</pre>	<pre>MAX2 = MAX MAX = A (I) ELSE IF A(I) &gt; MAX2 THEN MAX2 = A(I) ENDIF ENDIF , NEXT I PRINT MAX2</pre>												
СИ	Алгоритмический язык												
<pre>max = a [0] ; max2 = a [1] ; if (max &lt; max2) { max = a [1] ; max2 = a [0] ; } for(i = 2; i &lt; N; i++) if(a[i] &gt; max) { max 2 = max; max = a [i]; } else if (a [i] &gt; max2 ) max2 = a [i] ; cout « max2 « endl;</pre>	<pre>MAX := a [1] MAX2 := a [2] если MAX &lt; MAX2 то MAX := a [2 ] MAX2 := a [1] все нц для i от 3 до N если a[i]&gt;MAX то MAX2 := MAX MAX := a [i] иначе если a[i]&gt;MAX2 то MAX2 := a [i] все все кц вывод MAX2</pre>												
<p>26. 1. а) При следующих значениях: <math>S = 21</math> (добавить один камень) и <math>S \in \{11; 17\}</math> (удвоить число камней) — данные позиции будут выигрышными.  б) <math>S = 20</math> У Вали. Ему нужно будет добавить 1 камень, чтобы получить 22 (если Паша добавит 1 камень первым ходом). Иначе Валя выиграет, так как если Паша удвоит число камней, то их будет больше 34. Назовём эту позицию проигрышной, так как тот игрок, что делает в ней первый ход обречён на поражение.  <math>S = 19</math>. У Паши Два хода подряд, ему нужно просто добавить по одному камню:</p>	<p>26. 1. а) Паша может выиграть первым ходом, если <math>S = 12, 13, \dots, 30, 35</math>. При <math>S = 35</math> Паше достаточно добавить в кучу 1 камень, при <math>18 \leq S \leq 30</math> достаточно увеличить число камней в два раза, при остальных указанных значениях <math>S</math> нужно утроить количество камней. Замечание для проверяющего. При <math>S = 18, 19, 20</math> Паша может также утроить количество камней в куче. Ученик не обязан упоминать об этом.  б) При <math>S = 34</math> увеличивать количество камней в куче в два или три раза не имеет смысла, т. к. после такого хода выигрывает противник. Поэтому можно считать, что</p>	<p>26. 1. а) Петя может выиграть, если <math>S = 11, \dots, 30</math>. При меньших значениях <math>S</math> за один ход нельзя получить кучу, в которой не менее 31 камня. Пете достаточно увеличить количество камней в 3 раза и отнять два камня (действие Б).  б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет <math>S = 10</math> камней. Тогда после первого хода Пети в куче будет 11 камней или 28 камней. В обоих случаях Ваня выполняет действие Б и выигрывает в один ход.  2. Возможные значения <math>S</math>: 4, 9. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 10 камней (при <math>S = 4</math> он выполняет действие Б; при <math>S</math></p>	<p>26. Для удобства построим массив возможных состояний игры. Состояние — количество камней в кучке. Для каждого состояния укажем, выигрышное оно или проигрышное, то есть побеждает ли или проигрывает игрок, находясь в этом состоянии, при оптимальной игре обоих игроков. Состояние является проигрышным, если из него есть переходы только в выигрышные состояния. В противном случае состояние является выигрышным. Таким образом можно построить данный массив с конца, пометив все состояния с количеством камней большим 35 как про-</p>										

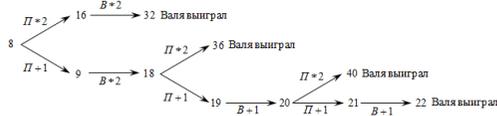


Назовём эту позицию выигрышной.  $S = 18$ . У Вали. Паша не сможет выиграть удвоением, а добавив один камень поставит Валу в выигрышную позицию (смотри предыдущий пункт). Эта позиция является проигрышной.

2.  $S = 10$ . Паша. Ему просто нужно удвоить число камней, и он поставит Валу в проигрышную позицию  $S = 20$ , которая рассмотрена в пункте 1б.

$S = 9$ . Паши. Ему тоже нужно удвоить число камней, и он поставит Валу в проигрышную позицию  $S = 18$ , которая рассмотрена в пункте 1б.

3.  $S = 8$  При правильной стратегии победит Валя. Изобразим дерево всех партий при этой стратегии:



единственно возможный ход – это добавление в кучу одного камня. После этого в куче станет 35 камней и Валя выиграт, добавив в кучу один камень. Таким образом, при  $S = 34$  выигрышная стратегия есть у Вали. При  $S = 33$  Паша может добавить в кучу один камень и получить кучу из 34 камней. В этой позиции выигрышная стратегия есть у игрока, который не должен делать ход, т.е. у самого Паши. Таким образом, при  $S = 33$  выигрышная стратегия есть у Паши. Аналогично, при  $S = 32$  выигрышная стратегия есть у Вали, а при  $S = 31$  – у Паши.

2. При  $S = 11$  после первого хода Паши в куче будет либо 12 камней, либо 22 камня, либо 33 камня. Во всех этих случаях (см. п. 1) выигрышная стратегия есть у игрока, который должен ходить, теперь это Валя. Таким образом, при  $S = 11$  выигрышная стратегия есть у Вали.

3. При  $S = 10$  выигрышная стратегия есть у Паши. Своим первым ходом Паша может добавить в кучу один камень и получить кучу, в которой 11 камней. В этой позиции выигрышная стратегия есть у игрока, который не должен ходить (см. п.2), то есть у Паши.

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

Положения после очередных ходов					
Исх. положение	1-й ход Паши (только ход по стратегии)	1-й ход Вали (все ходы)	2-й ход Паши (только ход по стратегии)	2-й ход Вали (все ходы)	3-й ход Паши (только ход по стратегии)
10	$10+1=11$	$11+1=12$	$12*2=24$		
		$11*2=22$	$22*2=44$		
		$11*3=33$	$33+1=34$	$34+1=35$	$35+1=36$
			$34*2=68$	$34*3=102$	

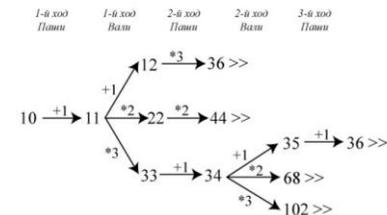


Рис.1. Дерево всех партий, возможных при Пашиней стратегии. Знаком >> обозначены позиции, в которых партия заканчивается

= 9 – добавляет 1 камень т. к. выполняет действие А). Эта позиция 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 выигрывает.

3. Возможное значение  $S$ : 8. После первого хода Пети в куче будет 9 камней или 22 камня. Если в куче станет 22 камня, Ваня выполнит действие Б и выигрывает своим первым ходом. Ситуация, когда в куче 9 камней, разобрана в п. 2. В этой ситуации 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				
Исх. полож.	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)
8	$8+1=9$	$9+1=10$	$10+1=11$	$3*11-2=31$
			$3*10-2=28$	$3*28-2=82$
			$3*8-2=22$	$3*22-2=64$

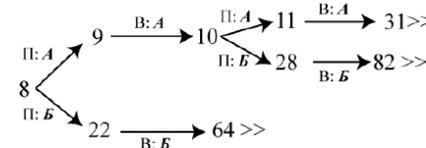


Рис.1. Дерево всех партий, возможных при Ваниней стратегии. Знаком >> обозначены позиции, в которых партия заканчивается

игрышные. Построим же массив. Пометим ячейку зелёным, если состояние выигрышное, и красным, если проигрышное.

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36

Теперь с помощью этого массива можно легко ответить на поставленные вопросы.

1.а) При всех  $S \geq 18$  Катя может выиграть, просто увеличив количество камней в кучке в два раза. При меньших значениях какой бы ход она не сделала, игра не закончится, а значит, и победить она не сможет.

1.б)  $S = 17$ . Катя может сделать 18, 19 или 34 камней. Для неё это не победа, зато для Иры – вполне. Для этого той надо просто увеличить количество камней вдвое.

2)  $S = 15$ ,  $S = 16$ . Катя сделает в кучке 17 камней, после чего повторится сценарий из (1.б)

3)  $S = 14$ . Построим таблицу. В ячейке будем указывать, кто ходил (И – Ира, К – Катя) и сколько камней на текущем ходу.

S 14	K 15, 16	I 17	K 18	I 36
			K 19	I 38
			K 34	I 35
	K 28	I 56		

27] Произведение двух чисел делится на 14, если:  
 — один из сомножителей делится на 14 (второй может быть любым) либо  
 — ни один из сомножителей не делится на 14, но один из сомножителей делится на 7, а другой – на 2.  
 Поэтому программа, вычисляющая число X, может работать так.  
 Программа читает все входные данные один раз, не запоминая все данные в массиве. Программа для прочитанного фрагмента входной последовательности хранит значения четырёх величин:  
 M7 – самое большое число, кратное 7, но не кратное 2;  
 M2 – самое большое число, кратное 2, но не кратное 7;  
 M14 – самое большое число, кратное 14;  
 MAX – самое большое число среди всех элементов последовательности, отличное от M14 (если число M14 встретилось более одного раза, и оно же является максимальным, то MAX = M14).  
 После того как все данные прочитаны, искомое число X вычисляется как максимум из произведений M14\*MAX и M7\*M2.  
 Ниже приведён пример программы на языке Паскаль, которая реализует описанный алгоритм.  
 Кроме того, приведён пример программы на языке Бейсик, которая правильно решает задачу, но использует алгоритм, немного отличающийся от описанного выше. Возможны и другие правильные алгоритмы.  
 Допускаются решения, записанные на других языках программирования.  
 Пример правильной и эффективной программы на языке Паскаль

```
var M7,M2,M14,MAX,dat,res,i,N: longint;
begin
M7 := 0;
M2 := 0;
M14 := 0;
MAX := 0;
readln(N);
for i := 1 to N do
begin
readln(dat);
if ((dat mod 7) = 0) and ((dat mod 2) > 0) and (dat > M7) then
M7 := dat;
if ((dat mod 2) = 0) and ((dat mod 7) > 0) and (dat > M2) then
M2 := dat;
if (dat mod 14 = 0) and (dat > M14) then
```

27] При заданных ограничениях числа в наборе могут иметь длину от 1 до 8. Необходимо создать массив из 8 элементов с индексами от 1 до 8 и использовать его для подсчёта количества чисел соответствующей длины. Использование массива не делает программу неэффективной по памяти, так как размер массива не зависит от N. Затем нужно найти значение максимального элемента этого массива и вывести максимальный из индексов элементов, равных этому максимуму и сам максимум. Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC).

**Пример правильной и эффективной программы на языке Паскаль**

```
var
N: integer; {количество чисел}
a: integer; {очередное число}
d: array[1..8] of integer; {подсчет}
mx: integer; {максимальное количество}
imx: integer; {самая частая длина}
i,k: integer;
begin
for i:=1 to 8 do d[i]:=0;
readln(N);
for i:=1 to N do begin
readln(a);
k:=0;
while a>0 do begin
k := k+1;
a := a div 10;
end;
d[k] := d[k]+1;
end;
mx := 0;
for i:=1 to 8 do begin
if d[i] >= mx then begin
mx := d[i];
imx := i;
end;
end;
writeln(imx, ', ', mx)
end.
```

**Пример правильной, но неэффективной программы на языке Паскаль**

```
var
N: integer; {количество чисел}
val: integer; {самая частая длина}
a: array [1..1000] of integer;
```

27] Наименьшая возможная сумма цифр числа в заданном диапазоне равна 0, наибольшая — 27. Необходимо создать массив с индексами от 0 до 27 и использовать его для подсчёта встречающихся сумм. Использование массива не делает программу неэффективной по памяти, так как размер массива не зависит от N. Затем нужно найти значение минимального элемента этого массива и вывести его индекс. Ниже приведены реализующие описанный выше алгоритм программы на языке Паскаль (использована система программирования PascalABC) и языке Java (версия языка 5.0 или выше). Программы отличаются способом вычисления суммы цифр очередного числа: в программе на Паскале использован обычный цикл разложения числа на цифры десятичной записи, в программе на Java сумма вычисляется по формуле, учитывающей, что число содержит не более трёх цифр. Оба способа допустимы.  
 Далее последует пример кода, который работает за линейное время.  
 Язык — PascalABC.

```
var
N: integer; {количество чисел}
a: integer; {очередное число}
s: integer; {сумма цифр числа}
k: array [0..27] of integer; {подсчёт сумм}
mn: integer; {минимум в массиве k}
imn: integer; {самая редкая сумма}
i: integer;
begin
for i:=0 to 27 do k[i]:=0;
readln(N);
for i:=1 to N do begin
readln(a);
s := 0;
while a>0 do begin
s := s + a mod 10;
a := a div 10;
end;
k[s] := k[s]+1;
end;
mn := N+1;
for i:=0 to 27 do begin
if (k[i]>0) and (k[i]<=mn) then begin
mn := k[i];
imn := i;
end;
end;
write(imn);
```

27] Основное подмножество состоит из всех значений сигналов, кроме 0, если он встречается, и кроме минимального нечётного значения, если таких значений чётное 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 из них. После окончания ввода распечатываются все номера, кроме номера 0 и номера минимального нечётного значения, но только в случае, если их количество чётно.  
 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:
<pre>var n,i,j,k,c,min,a: longint; begin readln(n); min := 100000001; 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 &lt;&gt; 0 then begin c := c + 1; if a &lt; min then begin min := a; k := i; end end NEXT i FOR i = 1 TO n IF (i &lt;&gt; j) AND ((c mod 2 &lt;&gt; 0) OR (i &lt;&gt; k)) THEN PRINT i NEXT i write(i, ' '); end.</pre>	<pre>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 &lt;&gt; 0 THEN c = c + 1 IF (min = 0) OR (a &lt; min) THEN min = a k = i END IF END IF NEXT i FOR i = 1 TO n IF (i &lt;&gt; j) AND ((c mod 2 &lt;&gt; 0) OR (i &lt;&gt; k)) THEN PRINT i NEXT i END</pre>

```

begin
if M14 > MAX then MAX := M14;
M14 := dat
end
else
if dat > MAX then
MAX := dat;
end;
if (M7*M2 < M14*MAX) then
res := M14*MAX
else
res := M7*M2;
writeln(res);
end.
Пример решения задачи А на языке Паскаль.
var
a: array[1..10000] of integer; {исходные данные}
N: integer; {количество элементов последовательности}
max: integer; {вычисляемое контрольное значение}
i, j: integer;
begin
readln(N);
max := 0;
for i := 1 to N do read(a[i]);
for i := 1 to N-1 do
for j := i+1 to N do
if (a[i]*a[j] > max) and (a[i]*a[j] mod 14 = 0) then
max := a[i] * a[j];
writeln(max);
end.

```

```

max_lenght: integer; {максимальное количество чисел}
i, j, k, lenght: integer;
begin
readln(N);
for i:=1 to N do read(a[i]);
for i:=1 to N do begin
k:=0;
while a[i]>0 do begin
k:=k+1;
a[i] := a[i] div 10;
end;
a[i]:=k;
end;
for i:=1 to N-1 do
for j:=1 to N-i do
if a[j] < a[j+1] then begin
k := a[j];
a[j] := a[j+1];
a[j+1] := k; end;
max_lenght := 0;
lenght := 1;
val := a[1];
for i := 1 to N do
if a[i]=a[i+1] then
lenght := lenght + 1
else if lenght>max_lenght then begin
max_lenght := lenght;
val := a[i];
lenght := 1;
end;
writeln(val, ', ', max_lenght);
end.

```

Вместо определения количества цифр с помощью последовательного деления на 10 можно использовать выбор по диапазону допустимых значений. Например, в приведённой выше программе можно вместо присваивания k:=0 и следующего за ним цикла while использовать такую последовательность действий:

```

if a<10 then k:=1
else if a<100 then k:=2
else if a<1000 then k:=3
else if a<10000 then k:=4
else if a<100000 then k:=5
else if a<1000000 then k:=6
else if a<10000000 then k:=7
else k:=8;

```

```

end.
import java.util.Scanner;
public class P27B {
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        int N = in.nextInt();
        int k[] = new int[28];
        for (int i = 0; i < N; ++i)
        {
            int a = in.nextInt();
            int s = a / 100 + a / 10 % 10 + a % 10;
            ++k[s];
        }
        int mn = N + 1;
        int imn = 0;
        for (int i = 0; i < k.length; ++i) {
            if (k[i] > 0 && k[i] <= mn) {
                mn = k[i];
                imn = i;
            }
        }
        System.out.print(imn);
    }
}

```

Пример правильной, но неэффективной программы на языке Паскаль.

```

var
N: integer; {количество чисел}
val: integer; {самая редкая сумма}
a: array [1..10000] of integer;
max_lenght: integer;
i, j, k, lenght: integer;
begin
readln(N);
for i:=1 to N do read(a[i]);
for i:=1 to N do a[i] := a[i] mod 10 + a[i] div 10 mod 10 + a[i] div 100 mod 10;
for i:=1 to N-1 do
for j:=1 to N-i do
if a[j] < a[j+1] then begin
k := a[j];

```

Пример решения задачи А на языке Паскаль.

```

var
a: array[1..10000] of integer; {исходные данные}
min: integer; {минимальное нечётное число}
N: integer; {количество чисел}
sum: integer; {максимальная сумма}
i: integer;
begin
readln(N);
min := 1000000001;
sum := 0;
for i := 1 to N do readln(a[i]);
for i := 1 to N do begin
sum := sum + a[i];
if (a[i] mod 2 <> 0) and (a[i] mod 2 < min) then min := a[i];
end;
if sum mod 2 = 0 then
for i := 1 to N do
if (a[i] < min) and (a[i]<>0) then write(i, ' ');
if sum mod 2 <> 0 then
for i := 1 to N do
if (a[i]<>0) then write(i, ' ');
end.

```

		<pre>a[j] := a[j+1]; a[j+1] := k; end; max_lenght := N+1; lenght := 0; val := a[1]; for i := 1 to N do   if a[i]=a[i+1] then     lenght := lenght + 1   else begin     if lenght       max_lenght := lenght;       val := a[i];     end;     lenght := 0;   end;   writeln(val); end.</pre>	
--	--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--