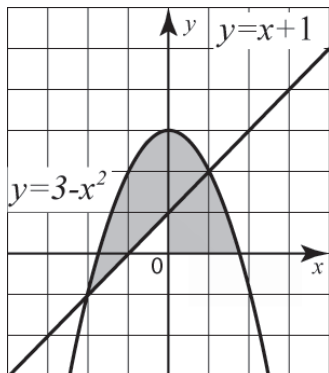


## Критерии оценивания заданий с развёрнутым ответом

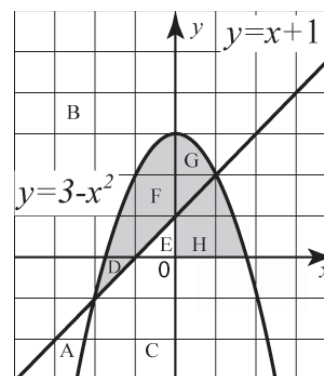
C1



Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости ( $x, y$  – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.

Бейсик	<pre> INPUT x, y IF y&lt;=3-x*x THEN   IF y&gt;=0 THEN     IF y&gt;=x+1 THEN       PRINT "принадлежит"     ELSE       PRINT "не принадлежит"     END IF   END IF END IF END </pre>
Паскаль	<pre> var x,y: real; begin   readln(x,y);   if y&lt;=3-x*x then     if y&gt;=0 then       if y&gt;=x+1 then         write('принадлежит')       else         write('не принадлежит')       end if     end if   end if end. </pre>

Си	<pre> void main(void) {   float x,y;   scanf("%f %f",&amp;x,&amp;y);   if (y&lt;=3-x*x)     if (y&gt;=0)       if (y&gt;=x+1)         printf("принадлежит");       else         printf("не принадлежит");     } } </pre>
Алгоритмический язык	<pre> алг нач   вещ x,y   ввод x,y   если y&lt;=3-x*x то     если y&gt;=0 то       если y&gt;=x+1 то         вывод 'принадлежит'       иначе         вывод 'не принадлежит'       все     все   все кон </pre>



Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям A, B, C, D, E, F, G и H (каждая область ограничена осями координат и нарисованными на рисунке графиками). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ( $y \leq 3 - x * x$ )	Условие 2 ( $y \geq 0$ )	Условие 3 ( $y \geq x + 1$ )	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					
H					

В столбцах условий укажите "да", если условие выполнится, "нет", если условие не выполнится, "-" (прочерк), если условие не будет проверяться, "не изв.", если программа ведёт себя по-разному для разных значений, принадлежащих данной области. В столбце "Программа выведет" укажите, что программа выведет на экран. Если программа ничего не выводит, напишите "-" (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите "не изв.". В последнем столбце укажите "да" или "нет".

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

Элементы ответа:

1.

Область	Условие 1 ( $y \leq 3 - x * x$ )	Условие 2 ( $y \geq 0$ )	Условие 3 ( $y \geq x + 1$ )	Программа выведет	Область обрабатывается верно
A	нет	–	–	–	нет
B	нет	–	–	–	нет
C	да	нет	–	–	нет
D	да	нет	–	–	нет
E	да	да	нет	не принадлежит	да
F	да	да	да	принадлежит	да
G	да	да	да	принадлежит	да
H	да	да	нет	не принадлежит	нет

2. Возможная доработка (Паскаль):

```
if (y <= 3 - x * x) and ((y >= x + 1) or ((x >= 0) and (y >= 0)))
  then write('принадлежит')
  else write('не принадлежит')
```

Возможны и другие способы доработки.

Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требуется выполнить три действия:</p> <ol style="list-style-type: none"> <li>1. Указать для каждой указанной в задании области, как будет работать программа, что она выведет на экран и правильно ли это (в виде таблицы).</li> <li>2. Исправить ошибку, связанную с ошибочной записью условного оператора (отсутствуют ELSE).</li> <li>3. Исправить ошибку, связанную с неполнотой набора условий (чтобы различить области E и H, нужно проверить условие <math>x &gt; 0</math>).</li> </ol> <p>Количество баллов, выставаемых за решение, – это количество успешно выполненных действий.</p> <ol style="list-style-type: none"> <li>1 Первое действие считается выполненным успешно, если не менее семи строк таблицы из восьми заполнены без ошибок.</li> <li>2 Второе действие считается успешно выполненным, если программа выдаёт одно из двух сообщений «принадлежит» или «не принадлежит» для любых чисел <math>x</math> и <math>y</math> и при этом программа не стала работать хуже,</li> </ol>	

чем раньше, то есть для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.

Возможный (не единственный!) способ выполнить это действие – добавить случаи ELSE к каждому условию IF.

3. Третье действие считается успешно выполненным, если верно определена закрашенная область, то есть программа выводит сообщение «принадлежит» для всех точек закрашенной области и только для них, для точек вне закрашенной области программа выводит «не принадлежит» или не выводит ничего.

Приведённым трем ограничениям, например, не удовлетворяют точки плоскости, у которых  $y < 3 - x^2$ ,  $y < x + 1$ ,  $x > 0$ ,  $y > 0$ . Поэтому возможным способом исправления ошибки может быть разбиение области на две части и использование дизъюнкции либо отбрасывание от большей области её части.

Успешно заполнена таблица. Программа работает правильно для всех областей.	3
1. Успешно выполнены два действия из трёх (исправлены обе ошибки, но в первом пункте задания не приведена таблица (либо таблица содержит ошибки в двух и более строках), либо приведена таблица (которая содержит ошибки не более чем в одной строке), но исправлена только одна ошибка программы). При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо « $y \geq 0$ » используется « $y > 0$ ».	2
2. Или выполнены все три действия, но при этом в логическом выражении неверно учтены приоритеты логических операций (не расставлены или неправильно расставлены скобки в выражениях).	
Правильно выполнено только одно действие из трёх, то есть, либо только приведена таблица, которая содержит ошибки в не более чем одной строке, либо таблица не приведена (или приведена и содержит ошибки более чем в одной строке), но исправлена одна ошибка программы. При оценивании этого задания на 1 балл допускается не учитывать корректность работы программ на точках границ областей (вместо нестрогих неравенств в решении были использованы строгие неравенства	1

или наоборот).	
Все пункты задания выполнены неверно (таблица анализа правильности алгоритма не приведена либо содержит ошибки в двух и более строках, программа не приведена, либо ни одна из двух ошибок не исправлена).	0
Максимальный балл	3

**C2** Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество элементов массива, кратных трем, десятичная запись которых заканчивается цифрой 1.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	<pre> const   N=30; var   a: array [1..N] of integer;   i, j, k: integer; begin   for i:=1 to N do     readln(a[i]);   ... end.</pre>
Бейсик	<pre> N=30 DIM A(N) AS INTEGER DIM I, J, K AS INTEGER FOR I = 1 TO N   INPUT A(I) NEXT I ... END</pre>

СИ	<pre>#include &lt;stdio.h&gt; #define N 30 void main(void){     int a[N];     int i, j, k;     for (i=0; i&lt;N; i++)         scanf("%d", &amp;a[i]);     ... }</pre>
Алгоритмический язык	<pre><u>алг</u> <u>нач</u>     <u>цел</u> N=30     <u>целтаб</u> a[1:N]     <u>цел</u> i, j, k     <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N         <u>ввод</u> a[i]     <u>кц</u>     ... <u>кон</u></pre>
Естественный язык	<p>Объявляем массив А из 30 элементов. Объявляем целые переменные I, J, K. В цикле от 1 до 30 вводим элементы массива А с 1-го по 30-й. ...</p>

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, Free Pascal 2.4) или в виде блок-схемы. В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

<b>Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)</b>	
<b>На языке Паскаль</b>	
<pre>k:=0; for i:=1 to N do begin     if (a[i] mod 3=0) and (a[i] mod 10=1) then k:=k+1; end; writeln(k);</pre>	
<b>На языке Бейсик</b>	
<pre>K = 0 FOR I = 1 TO N     IF A(I) MOD 3=0 AND A(I) MOD 10=1 THEN         K = K + 1     END IF NEXT I PRINT K</pre>	
<b>На языке СИ</b>	
<pre>k=0; for (i=0; i&lt;N; i++) {     if (a[i]%3==0 &amp;&amp; a[i]%10==1) ++k; } printf("%d", k);</pre>	
<b>На естественном языке</b>	
<p>Записываем в переменную k начальное значение, равное 0. В цикле от первого элемента до тридцатого находим остаток от деления элемента исходного массива на 3 и на 10. Если остаток от деления на три равен нулю, а остаток от деления на десять равен 1, то увеличиваем значение k на 1. Переходим к следующему элементу массива. После завершения цикла выводим значение переменной k.</p>	

Указания по оцениванию	Баллы
Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы	2
в любом варианте решения может присутствовать не более одной ошибки из числа следующих: 1) не инициализируется или неверно инициализируется переменная k; 2) неверно осуществляется проверка на делимость на три; 3) неверно проверяется последняя цифра числа; 4) вместо значения элемента проверяется его индекс; 5) неверно осуществляется подсчёт количества элементов (например, вместо количества считается сумма); 6) отсутствует вывод ответа; 7) используется переменная, не объявленная в разделе описания переменных; 8) не указано или неверно указано условие завершения цикла; 9) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно; 10) неверно расставлены операторные скобки	1
Ошибок, перечисленных в п. 1–10, две или больше, или алгоритм сформулирован неверно	0
Максимальный балл	2

**СЗ** У исполнителя Увеличитель две команды, которым присвоены номера:

1. прибавь 2,
2. умножь на 3.

Первая из них увеличивает число на экране на 2, вторая – умножает его на 3.

Программа для Увеличителя – это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 62?

Ответ обоснуйте.

**Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)**

Обозначим через  $R(n)$  – количество программ, которые преобразуют число 2 в число  $n$ . Очевидно, если  $n$  – нечетное, то  $R(n) = 0$ . Поэтому ниже будем рассматривать только чётные числа  $n$ . Обозначим через  $t(n)$  наибольшее чётное число, кратное трём, не превосходящее  $n$ .

Обе команды исполнителя увеличивают исходное число, поэтому общее количество команд в программе не может превосходить 30.

Верны следующие соотношения:

1. Если  $n$  не делится на 3, то тогда  $R(n) = R(t(n))$ , так как существует единственный способ получения  $n$  из  $t(n)$  – прибавлением двоек.
2. Пусть  $n$  делится на 3.

Тогда  $R(n) = R(n/3) + R(n-2) = R(n/3) + R(n-6)$  (если  $n > 6$ ).

При  $n=6$  выполнено:  $R(n) = 2$  (два способа: прибавлением двух двоек или однократным умножением на 3).

Поэтому достаточно по индукции вычислить значения  $R(n)$  для всех чётных чисел, кратных трем и не превосходящих 62.

Имеем:

$$R(2) = R(4) = 1$$

$$R(6) = 2 = R(8) = R(10)$$

$$R(12) = R(4) + R(6) = 1 + 2 = 3 = R(14) = R(16)$$

$$R(18) = R(6) + R(12) = 2 + 3 = 5 = R(20) = R(22)$$

$$R(24) = R(8) + R(18) = 2 + 5 = 7 = R(26) = R(28)$$

$$R(30) = R(10) + R(24) = 2 + 7 = 9 = R(32) = R(34)$$

$$R(36) = R(12) + R(30) = 3 + 9 = 12 = R(38) = R(40)$$

$$R(42) = R(14) + R(36) = 3 + 12 = 15 = R(44) = R(46)$$

$$R(48) = R(16) + R(42) = 3 + 15 = 18 = R(50) = R(52)$$

$$R(54) = R(18) + R(48) = 5 + 18 = 23 = R(56) = R(58)$$

$$R(60) = R(20) + R(54) = 5 + 23 = 28 = R(62)$$

Ответ: 28

### Другой способ решения

Количество программ, которые преобразуют число 2 в число  $n$ , будем обозначать через  $R(n)$ . Очевидно, если  $n$  – нечетное, то  $R(n) = 0$ . Поэтому ниже будем рассматривать только чётные числа  $n$ .

Будем решать поставленную задачу последовательно для чисел 2, 4, 6, ..., 62 (то есть для каждого из чисел определим, сколько программ исполнителя существует для его получения). Количество программ, которые преобразуют число 2 в число  $n$ , будем обозначать через  $R(n)$ . Число 2 у нас уже есть, значит, его можно получить с помощью «пустой» программы. Любая непустая программа увеличит исходное число, т.е. даст число, больше 2. Значит,  $R(2) = 1$

Для каждого следующего числа рассмотрим, из какого числа оно может быть получено за одну команду исполнителя. Если число не делится на три, то оно может быть получено только из предыдущего чётного числа с помощью команды **прибавь 2**. Значит, количество искомых программ для такого числа равно количеству программ для предыдущего чётного числа:  $R(i) = R(i-2)$ . Если число на 3 делится, то вариантов последней команды два: **прибавь 2** и **умножь на 3**, тогда  $R(i) = R(i-2) + R(i/3)$ . Заполним соответствующую таблицу по приведённым формулам слева направо

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	
1	1	2	2	2	3	3	3	5	5	5	7	7	7	9	
<b>32</b>	<b>34</b>	<b>36</b>	<b>38</b>	<b>40</b>	<b>42</b>	<b>44</b>	<b>46</b>	<b>48</b>	<b>50</b>	<b>52</b>	<b>54</b>	<b>56</b>	<b>58</b>	<b>60</b>	<b>62</b>
9	9	12	12	12	15	15	15	18	18	18	23	23	23	28	28

При этом ячейки, относящиеся к числам, которые не делятся на 3, можно в решении и опустить (за исключением первого и последнего чисел):

2	6	12	18	24	30	36	42	48	54	60	62
1	2	3	5	7	9	12	15	18	23	28	28

Ответ: 28

Указания по оцениванию	Баллы
Правильное указание количества возможных программ со строгим доказательством правильности (одним из приведённых выше способов или любым другим).	3
<p>Два балла ставятся в одном из двух случаев:</p> <ol style="list-style-type: none"> <li>Правильное указание количества возможных программ, основанное на верных рассуждениях, но доказательство правильности неполно. В частности, оценка в 2 балла выставляется в случае, если просто перечислены все правильные программы и не доказано отсутствие других программ, кроме приведённых.</li> <li>Приведены правильные и строгие рассуждения, доведённые до конца, но в вычислениях допущена арифметическая ошибка, в результате чего получен неверный ответ</li> </ol>	2
<p>Представленное решение обладает одним из свойств</p> <ol style="list-style-type: none"> <li>Указано, что нужно рассматривать значения <math>n</math>, меньшие, чем 62, и приведены правильные рекуррентные соотношения (см. выше), возможно, неполные.</li> <li>Правильно выписаны и обоснованы значения <math>R(n)</math> для небольших <math>n</math>.</li> </ol> <p>Правильно написан ответ, но нет его обоснования.</p>	1
Не выполнено ни одно из перечисленных выше условий	0
Максимальный балл	3

**С4** На вход программе сначала подается значение  $N$  – количество учеников школы, следующие  $N$  строк содержат список дней рождения учеников этой школы. Каждый день рождения занимает отдельную строку, содержащую дату, записанную по правилам русского языка строчными буквами, например, «двадцать третье февраля». Год не указывается, слова отделяются друг от друга одним пробелом, в начале и в конце строки дополнительных пробелов нет. Напишите эффективную программу, которая определяет количество дней рождения в каждом календарном месяце и выводит названия месяцев и соответствующие количества в порядке убывания количества дней рождения в месяце. Если у двух и более месяцев количество дней рождения совпадает, эти месяцы следует выводить в том порядке, в каком они следуют в календаре. Если в каком-то месяце дней рождения нет, данные об этом месяце выводить не надо.



Размер памяти, которую использует Ваша программа, не должен зависеть от длины исходного списка.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

*Пример входных данных:*

```
5
первое мая
восьмое марта
девятое мая
двадцать третье февраля
девятое мая
```

*Пример выходных данных для приведенного выше примера входных данных:*

```
май 3
февраль 1
март 1
```

**Содержание верного ответа и указания по оцениванию  
(допускаются иные формулировки ответа, не искажающие его  
смысла)**

Программа создаёт массив из 12 счётчиков для подсчета количества дней рождения в каждом месяце. Входные данные читаются один раз, без запоминания в массиве. Для каждой прочитанной строки определяется месяц (это всегда последнее слово в строке) и увеличивается соответствующий счётчик. После завершения ввода массив счётчиков сортируется. При сортировке необходимо сохранить соответствие счётчиков и названий месяцев и правильно (в календарном порядке) расположить месяцы с равным количеством дней рождения. Поскольку размер массива счётчиков невелик, допускается вместо сортировки использовать многократный выбор максимального элемента и его удаление. Разрешается вместо программирования сортировки обращаться к стандартным средствам языка (для языков, в которых есть такие средства).

Ниже приведены примеры решения задания на языках Паскаль и Бейсик, а также на Алгоритмическом языке.

Задачи такого типа допускают более короткие решения с использованием языков, в которых есть возможность использования ассоциативных массивов (массивов, в которых индексами служат строки, а не числа). Такие решения допускаются, они оцениваются на общих основаниях с учётом правил и особенностей выбранного

учеником языка. В качестве примера ниже приведена программа решения задачи на языке Перл.

Допускаются решения, записанные на других языках программирования.

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

```
program c4_dates;
const
  month_in: array[1..12] of string = (
    'января', 'февраля', 'марта',
    'апреля', 'мая', 'июня',
    'июля', 'августа', 'сентября',
    'октября', 'ноября', 'декабря'
  );
  month_out: array[1..12] of string = (
    'январь', 'февраль', 'март',
    'апрель', 'май', 'июнь',
    'июль', 'август', 'сентябрь',
    'октябрь', 'ноябрь', 'декабрь'
  );
var
  N: integer;
  count: array[1..12] of integer;
  i,j,k: integer;
  line, month: string;
  max: integer;
  imax: integer;
begin
  readln(N);
  for j:=1 to 12 do count[j]:=0;
  for i:=1 to N do begin
    readln(line);
    k:=length(line);
    while line[k]<>' ' do k:=k-1;
    month:=copy(line,k+1,length(line)-k);
    j:=1;
    while month_in[j]<>month do j:=j+1;
    count[j]:=count[j]+1;
  end;
  repeat
    imax:=1; max:=count[1];
    for i:=2 to 12 do begin
      if count[i]>max then begin
        imax:=i; max:=count[i];
      end;
    end;
```

```

end;
if max>0 then begin
  writeln(month_out[imax], ' ', max);
  count[imax]:=0;
end;
until max=0;
end.

```

### Пример правильной и эффективной программы на Алгоритмическом языке:

алг c4dates

нач

```

литтаб месяц_род[1:12], месяц_им[1:12]
лит строка, месяц
цел N
цел i, j, k
цел mx, imx
целтаб счёт[1:12]
месяц_род[1] := "января"
месяц_род[2] := "февраля"
месяц_род[3] := "марта"
месяц_род[4] := "апреля"
месяц_род[5] := "мая"
месяц_род[6] := "июня"
месяц_род[7] := "июля"
месяц_род[8] := "августа"
месяц_род[9] := "сентября"
месяц_род[10] := "октября"
месяц_род[11] := "ноября"
месяц_род[12] := "декабря"
нц для i от 1 до 12
  месяц := месяц_род[i]
  k := длин(месяц)
  выбор
    при месяц[k] = 'а' : месяц := месяц[1:k-1]
    при месяц[k-1] = 'а' : месяц[k] := 'й'
    при месяц[k] = 'я' : месяц[k] := 'ь'
  все
  месяц_им[i] := месяц
кц
нц для i от 1 до 12
  счёт[i] := 0
кц
ввод N
нц для i от 1 до N
  ввод строка

```

```

k := длин(строка)
нц пока строка[k] <> ' '
  k := k-1
кц
месяц := строка[k+1:длин(строка)]
нц для j от 1 до 12
  если месяц = месяц_род[j]
    то счёт[j] := счёт[j] + 1
    выход
  все
кц
нц
mx := счёт[1]; imx := 1
нц для i от 2 до 12
  если счёт[i] > mx
    то mx := счёт[i]
    imx := i
  все
кц
если mx > 0
  то вывод нс, месяц_им[imx], " ", mx
  счёт[imx] := 0
  все
кц при mx = 0
кон

```

### Пример правильной и эффективной программы на языке Бейсик:

```

DATA января, февраля, марта, апреля, мая, июня
DATA июля, августа, сентября, октября, ноября, декабря
DATA январь, февраль, март, апрель, май, июнь
DATA июль, август, сентябрь, октябрь, ноябрь, декабрь
DIM monthin$(12), monthout$(12)
FOR i = 1 TO 12
  READ monthin$(i)
NEXT i
FOR i = 1 TO 12
  READ monthout$(i)
NEXT i

DIM count(12) AS INTEGER
INPUT N
FOR i = 1 TO 12
  count(j) = 0
NEXT i
FOR i = 1 TO N
  LINE INPUT line$

```



```

k = LEN(line$)
WHILE line$(k) <> " "
    k = k - 1
WEND
month$ = MID$(line$, k + 1)
j = 1
WHILE monthin$(j) <> month$
    j = j + 1
WEND
count(j) = count(j) + 1
NEXT i
DO
    imax = 1: max = count(1)
    FOR i = 2 TO 12
        IF count(i) > max THEN
            imax = i: max = count(i)
        END IF
    NEXT i
    IF max > 0 THEN
        PRINT monthout(imax); " "; max
        count(imax) = 0
    END IF
LOOP UNTIL max = 0

```

### Пример правильной и эффективной программы на языке Perl:

```

%mmnum = (
    'января' => 0,
    'февраля' => 1,
    'марта' => 2,
    'апреля' => 3,
    'мая' => 4,
    'июня' => 5,
    'июля' => 6,
    'августа' => 7,
    'сентября' => 8,
    'октября' => 9,
    'ноября' => 10,
    'декабря' => 11
);
@mmname = (
    'январь', 'февраль', 'март',
    'апрель', 'май', 'июнь',
    'июль', 'август', 'сентябрь',
    'октябрь', 'ноябрь', 'декабрь'
);

```

```

$n = <>;
while ($n--) {
    $line = <>;
    ($month) = $line =~ /\(S*)$/;
    ++$count{$month};
}
for (sort
    {$count{$b}<=>$count{$a} || $mnum{$a}<=>$mnum{$b}}
    keys %count)
{
    last if $count{$_}==0;
    print "$mname[$mnum{$_}] $count{$_}\n";
}

```

Указания по оцениванию	Баллы
Программа правильно работает для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве. Допускается наличие в тексте программы одной синтаксической ошибки: пропущен или неверно указан знак пунктуации, неверно написано или пропущено зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку). Оценка не снижается за орфографические ошибки в названиях месяцев	4
Программа работает верно, но размер используемой памяти зависит от длины используемой последовательности. Например, входные данные запоминаются в массиве или другой структуре данных (например, контейнер <code>priority_queue</code> , <code>set</code> или <code>map</code> в C++), размер которого соответствует числу N. Допускается наличие от одной до трех синтаксических ошибок. Возможно, в принципиально верно организованном вводе данных есть ошибка.	3

Программа работает в целом верно, эффективно или нет, но в реализации алгоритма содержатся ошибки. Например, при выводе результатов неверно организовано упорядочение месяцев с равным количеством дней рождения. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	2
Программа работает в отдельных частных случаях, например, обрабатывает только случаи, когда число в дате состоит из одного слова, или выводит все значения подряд без сортировки. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше	1
Прочее	0
<i>Максимальный балл</i>	4

## Ответы к заданиям с выбором ответа

№ задания	Ответ
A1	1
A2.	2
A3.	4
A4	2
A5	2
A6.	4
A7	2

№ задания	Ответ
A8.	1
A9.	3
A10.	4
A11.	2
A12.	2
A13.	1

## Ответы к заданиям с кратким ответом

№ задания	Ответ
B1.	11
B2.	12121
B3.	52
B4	CCCCO
B5.	5
B6	90
B7.	211
B8.	6

№ задания	Ответ
B9.	18
B10	606
B11.	HBEA
B12	150
B13.	7
B14	18
B15.	30

## Ответы к заданиям с выбором ответа

№ задания	Ответ
A1.	3
A2.	2
A3.	3
A4.	1
A5.	1
A6.	2
A7.	3

№ задания	Ответ
A8.	2
A9.	1
A10.	3
A11.	3
A12.	4
A13.	1

## Ответы к заданиям с кратким ответом

№ задания	Ответ
B1.	440
B2.	21121
B3.	21
B4.	CCCCH
B5.	3
B6.	120
B7.	91
B8.	5

№ задания	Ответ
B9.	27
B10.	404
B11.	EADH
B12.	440
B13.	8
B14.	2
B15.	30