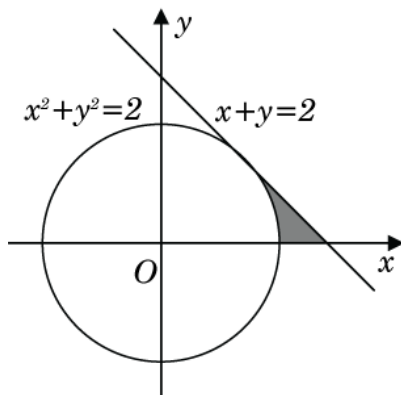


## Критерии оценивания заданий с развёрнутым ответом

С1

Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости ( $x, y$  – действительные числа) и определяется принадлежность этой точки заданной заштрихованной области (включая границы). Область ограничена осью абсцисс, окружностью  $x^2+y^2=2$  и прямой  $x+y=2$ . Заданные прямая и окружность касаются в точке  $x=1, y=1$ . Программист торопился и написал программу неправильно.



ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var x,y: real; begin readln(x,y); if x*x+y*y&gt;=2 then If y&gt;=0 then if x+y&lt;=2 then write ('принадлежит') else write('не принадлежит') end.</pre>	<pre>INPUT x, y IF x*x+y*y&gt;=2 THEN IF y&gt;=0 THEN IF x+y&lt;=2 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y; scanf("%f% f",&amp;x,&amp;y); if (x*x+y*y&gt;=2) if (y&gt;=0) if (x+y&lt;=2) printf ("принадлежит"); else printf("не принадлежит"); }</pre>

Последовательно выполните следующее:

- 1) Приведите пример таких чисел  $x, y$ , при которых программа неверно решает поставленную задачу.
- 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы).

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

Элементы ответа:

Пример:  $x=0, y=0$  (Любая пара  $(x,y)$ , для которой выполняется:  $x^2+y^2 < 2$  или  $y < 0$  или  $(x+y < 2$  и  $x < 1)$ )

Возможная доработка (Паскаль):

```
if (x*x+y*y>=2) and (y>=0) and (x+y<=2) and (x>=1) then
write('принадлежит') else
write('не принадлежит')
```

Возможны и другие способы доработки.

**Указания по оцениванию**

Обратите внимание! В задаче требовалось выполнить **три** действия: указать пример входных данных, при которых программа работает неверно и исправить две ошибки:

1. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE).
2. Приведенным трем ограничениям удовлетворяют также те точки плоскости, у которых  $(x^2+y^2 > 2)$  и  $(y > 0)$  и  $(x+y < 2)$  и  $(x < 1)$ .

Правильно выполнены оба пункта задания. Исправлены две ошибки.

В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения

**Баллы**

3

<p>Правильно выполнены 2 действия из трех (исправлены обе ошибки, но не указан/неправильно указан пример требуемых входных данных, либо правильно указан пример входных данных, программа правильно работает при большем числе случаев, чем исходная, но не при всех, и, при доработке не возникли точки, в которых исходная программа работает верно, а доработанная - нет).</p> <p>Например, если дан правильный ответ на вопрос №1, добавлены ветки "else" ко всем условиям "if", но программа выдает "принадлежит" для точек, у которых <math>(x^2+y^2 \geq 2)</math> и <math>(y &gt; 0)</math> и <math>(x+y &lt;= 2)</math> и <math>(x &lt; 1)</math>:</p> <pre>if x*x+y*y &gt;= 2 then if y &gt;= 0 then if x+y &lt;= 2 then write('принадлежит') else write('не принадлежит') else write('не принадлежит') else write('не принадлежит')</pre>	2
<p>Правильно выполнено только одно действие из трёх. То есть, только приведен пример входных данных, либо он не приведен, но имеется программа, корректно работающая при большем количестве входных данных, чем исходная, но не при всех.</p>	1
<p>Все пункты задания выполнены неверно (пример входных данных не указан или указан неверно, программа не приведена, либо приведенная программа корректно работает в не большем количестве случаев, чем исходная).</p>	0

**C2**

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 100 – баллы учащихся выпускного класса за итоговый тест по информатике. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет найти и вывести количество учащихся, чья оценка за тест выше средней оценки в классе. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

<b>Паскаль</b>	<b>Бейсик</b>
<pre>Const N=30; var a: array [1..N] of integer; i, j: integer; s: real; begin for i:=1 to N do readln(a [i]); ... end.</pre>	<pre>N=30 DIM A(N) AS INTEGER DIM I, J AS INTEGER S AS SINGLE FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
<b>СИ</b>	<b>Естественный язык</b>
<pre>#include &lt;stdio.h&gt; #define N 30 void main(void) {int a[N]; int i, j; float s; for (i=0; i&lt;N; i++) scanf("%d", &amp;a[i]); ... }</pre>	<p>Объявляем массив А из 30 элементов. Объявляем целочисленные переменные I, J. Объявляем действительную переменную S. 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, Borland Pascal 7.0) 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</b> (допускаются иные формулировки ответа, не искажающие его смысла)		
<b>Паскаль</b>	<b>Бейсик</b>	
<pre>s:=0; for i:=1 to N do S:=S+A[i]; j:=0; for i:=1 to N do   if A[i]*N&gt;s then j:=j+1; WriteLn(j);</pre>	<pre>S=0 FOR I = 1 TO N S=S+A(i) NEXT I J=0 FOR I = 1 to N IF A(I)*N&gt;S THEN J=J+1 NEXT I PRINT J</pre>	
<b>СИ</b>	<b>Естественный язык</b>	
<pre>s=0; for (i=0;i&lt;N;i++)   s=s+a[i]; j=0; for (i=0;i&lt;N;i++)   if (a[i]*N&gt;s)     j=j+1; printf("%d", j);</pre>	<p>Записываем в переменную S начальное значение, равное 0. В цикле от первого элемента до тридцатого прибавляем значение текущего элемента массива к значению переменной S. После завершения цикла делим значение S на 30. Записываем в переменную J начальное значение, равное 0. В цикле от первого элемента до тридцатого сравниваем значение текущего элемента со значением переменной S. Если значение текущего элемента больше значения S, то увеличиваем значение переменной J на 1. После завершения цикла выводим значение переменной J.</p>	
<b>Указания по оцениванию</b>		<b>Баллы</b>
<p>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.</p>		2

<p>В любом варианте решения может присутствовать не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> <li>1. Не инициализируется или неверно инициализируется переменные S или J.</li> <li>2. В сравнении с переменной S вместо знака «больше» используется знак «больше или равно».</li> <li>3. Отсутствует вывод ответа</li> <li>4. Используется переменная, не объявленная в разделе описания переменных.</li> <li>5. Не указано или неверно указано условие завершения цикла</li> <li>6. Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно.</li> <li>7. Неверно расставлены операторные скобки.</li> <li>8. Не выполняется (неверно выполняется) деление на N при подсчете среднего балла.</li> </ol>	1
<p>Ошибок, перечисленных в п. 1–8, две или больше, или алгоритм сформулирован неверно.</p>	0

**С3**

Два игрока играют в следующую игру. На координатной плоскости стоит фишка. В начале игры фишка находится в точке с координатами  $(-3, 2)$ . Игроки ходят по очереди. Ход состоит в том, что игрок перемещает фишку из точки с координатами  $(x, y)$  в одну из трех точек:  $(x-1, y+3)$ ,  $(x+3, y-1)$ ,  $(x+2, y+2)$ . Игра заканчивается, как только расстояние от фишки до начала координат превысит число 8. Выигрывает игрок, который сделал последний ход. Кто выигрывает при безошибочной игре – игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)			
Выигрывает второй игрок. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке координаты фишки на каждом этапе игры.			
1 ход	2 ход	3 ход	4 ход
I-й игрок (все варианты хода)	II-й игрок (выигрышный ход)	I-й игрок (все варианты хода)	II-й игрок (выигрышный ход)
-4,5	-5,8	Второй игрок выигрывает вторым ходом	
0,1	2,3	1,6	0,9 или 3,8
		5,2	8,1 или 7,4
		4,5	Любой ход
-1,4	2,3	Те же варианты 3-4 хода	
Таблица содержит <b>все возможные</b> варианты ходов первого игрока. Из неё видно, что при любом ходе первого игрока у второго имеется ход, приводящий к победе.			
Указания по оцениванию			Баллы
Правильное указание выигрывающего игрока и его ходов со строгим доказательством правильности (с помощью или без помощи дерева игры).			3
Правильное указание выигрывающего игрока, стратегии игры, приводящей к победе, но при отсутствии доказательства ее правильности.			2
При наличии в представленном решении одного из пунктов: 1. Правильно указан выигрывающий игрок и его первый ход, рассмотрены все возможные ответы второго игрока, но неверно определены дальнейшие действия. 2. Правильно указан выигрывающий игрок и его первый ход, но описание выигрышной стратегии неполно и рассмотрены несколько (больше одного, но не все) вариантов ответов второго игрока.			1
Задание не выполнено или в представленном решении полностью отсутствует описание элементов выигрышной стратегии, и отсутствует анализ вариантов первого-второго ходов играющих (даже при наличии правильного указания выигрывающего игрока).			0

С4

Имеется список людей с указанием их фамилии, имени и даты рождения. Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет определять самого старшего человека из этого списка и выводить его фамилию, имя и дату рождения, а если имеется несколько самых старших людей с одинаковой датой рождения, то определять их количество.

На вход программе в первой строке подается количество людей в списке N. В каждой из последующих N строк находится информация в следующем формате:

<Фамилия> <Имя> <Дата рождения>

где <Фамилия> – строка, состоящая не более, чем из 20 символов без пробелов,

<Имя> – строка, состоящая не более, чем из 20 символов без пробелов,

<Дата рождения> – строка, имеющая вид ДД.ММ.ГГГГ, где ДД – двузначное число от 01 до 31, ММ – двузначное число от 01 до 12, ГГГГ – четырехзначное число от 1800 до 2100.

Пример входной строки:

Иванов Сергей 27.03.1993

Программа должна вывести фамилию и имя самого старшего человека в списке.

Пример выходных данных:

Иванов Сергей

Если таких людей, несколько, то программа должна вывести их количество. Пример вывода в этом случае:

3

#### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Программа читает все входные данные один раз, не запоминая их в массиве. Во время чтения данных определяется дата рождения каждого человека, которая сравнивается с наименьшей из известных дат рождения среди всех уже просмотренных людей из списка. Также запоминается количество людей, имеющих наименьшую дату рождения, фамилия и имя человека, имеющего наименьшую дату рождения.

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая (например, когда только один человек имеет наименьшую дату рождения).

Ниже приведены примеры решения задания на языках Паскаль и C++. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования.

Решение на языке Паскаль

```

Var
N, i, Day, Month, Year, MinDay, MinMonth, MinYear, MinCount,
Code: integer;
Name, MinName: string;
Begin
  MinYear:=9999;
  {Считываем количество людей}
  ReadLn(N);
  for i:=1 to N do
  begin
    {Считываем данные об одном человеке}
    ReadLn(Name);
    {Извлекаем из строки значения дня, месяца, года рождения}
    val (Copy (Name, Length (Name) -9, 2) , Day, Code) ;
    val (Copy (Name, Length (Name) -6, 2) , Month, Code) ;
    val (Copy (Name, Length (Name) -3, 4) , Year, Code) ;
    {Если дата рождения совпадает с наименьшей известной
датой, то увеличиваем счетчик числа людей, имеющих минимальную
дату рождения}
    if (Year=MinYear) and (Month=MinMonth) and (Day=MinDay)
    then
      MinCount:=MinCount+1;
    {Если дата рождения меньше минимальное известной, то
сохраняем его дату рождения, как минимальное, счетчик числа
людей, имеющих такую дату рождения, устанавливаем в 1, также
запоминаем имя человека}
    if (Year<MinYear) or (Year=MinYear) and (Month<MinMonth)
or (Year=MinYear) and (Month=MinMonth) and (Day<MinDay)
    then
      begin
        MinYear:=Year;
        MinMonth:=Month;
        MinDay:=Day;
        MinCount:=1;
        MinName:=Copy (Name, 1, length (Name) -11)
      end
    end;
    {Если счетчик числа людей, имеющих минимальную дату рождения
больше 1, то выводим значение счетчика, иначе выводим имя
человека}
    if MinCount>1
    then
      WriteLn (MinCount)
    else

```

```

  WriteLn (MinName)
end.

```

Решение на языке C++

```

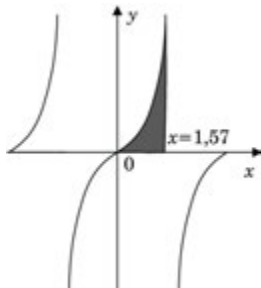
#include<iostream>
using namespace std;
int main()
{
  int n, i, day, month, year, minday, minmonth, minyear,
mincount;
  string name, minname;
  cin >> n;
  getline(cin, name);
  minyear=9999;
  for(i=0;i<n;++i)
  {
    getline(cin, name);
    day=(name[name.length()-10]-'0')*10 + name[name.length()-
9]-'0';
    month=(name[name.length()-7]-'0')*10 + name[name.length()-
9]-'6';
    year=((name[name.length()-4]-'0')*10 + name[name.length
()-3]-'0') * 10 + name[name.length()-2]-'0')*10 + name
[name.length()-1]-'0';
    if (year==minyear && month==minmonth && day==minday)
      ++mincount;
    if (year<minyear || year==minyear && month<minmonth ||
year==minyear && month==minmonth && day<minday)
      {
        minyear=year;
        minmonth=month;
        minday=day;
        mincount=1;
        name.resize(name.length()-11);
        minname=name;
      }
  }
  if(mincount>1)
    cout<<mincount<<endl;
  else
    cout<<minname<<endl;
  return 0;
}

```

Указания по оцениванию	Баллы
<p>Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке) или диапазону возможных дней рождений. Программа просматривает входные данные один раз, используя для нахождения ответа набор из нескольких переменных. 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).</p>	4
<p>Программа работает верно, но входные данные запоминаются в массиве (двух массивах), или входные данные считываются несколько раз. Возможно, вместо алгоритма поиска минимума используется сортировка всех дат рождения. Допускается наличие от одной до трех синтаксических ошибок: Возможно, в принципиально верно организованном вводе данных есть одна ошибка. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна ошибка, в результате которой программа работает не верно на некоторых (не типичных) наборах входных данных (например, все даты рождения совпадают).</p>	3
<p>Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения минимума, выход за границу массива, перевод символов в числа, используется знак "&lt;" вместо "&lt;=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных. Возможно неверно считается количество людей, имеющих минимальную дату рождения или, наоборот, программа не выводит имя человека, имеющего минимальную дату рождения, если такой человек один. При использовании сортировки допущены ошибки в ее реализации. Допускается до 4 различных ошибок в реализации алгоритма, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок, описанных выше.</p>	1
<p>Задание не выполнено или выполнено неверно.</p>	0

## Критерии оценивания заданий с развёрнутым ответом

**С1** Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости ( $x, y$  – действительные числа) и определяется принадлежность этой точки заданной заштрихованной области (включая границы). Область ограничена осями координат, графиком функции  $y = \operatorname{tg} x$  и прямой  $x = 1,57$ . Программист торопился и написал программу неправильно.



ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var x,y: real; begin readln(x,y); if y&gt;=0 then if x&lt;=1.57 then if y&lt;=tan(x) then write ('принадлежит') else write('не принадлежит') end. end.</pre>	<pre>INPUT x, y IF y&gt;=0 THEN IF x&lt;=1.57 THEN IF y&lt;=tan(x) THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void) { float x,y; scanf("%f% f",&amp;x,&amp;y); if (y&gt;=0) if (x&lt;=1.57) if (y&lt;=tan(x)) printf ("принадлежит"); else printf("не принадлежит"); }</pre>

Последовательно выполните следующее:

- 1) Приведите пример таких чисел  $x, y$ , при которых программа неверно решает поставленную задачу.
- 2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы).

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

Элементы ответа:

Пример:  $x = 0, y = -1$  (Любая пара  $(x, y)$ , для которой выполняется:  $y < 0$  или  $x > 1.57$  или  $(y \leq \operatorname{tg} x$  и  $x < 0)$ )

Возможная доработка (Паскаль):

```
if (y>=0) and (x<=1.57) and (y<=tan(x)) and (x>=0) then
write('принадлежит') else
write('не принадлежит')
```

Возможны и другие способы доработки.

**Указания по оцениванию**

**Баллы**

Обратите внимание! В задаче требовалось выполнить **три** действия: указать пример входных данных, при которых программа работает неверно и исправить две ошибки:

1. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE).
2. Приведенным трем ограничениям удовлетворяют также те точки плоскости, у которых  $(y < \operatorname{tg} x)$  и  $(y > 0)$  и  $(x < 0)$ .

Правильно выполнены оба пункта задания. Исправлены две ошибки. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.

3

Правильно выполнены 2 действия из трех (исправлены обе ошибки, но не указан/неправильно указан пример требуемых входных данных, либо правильно указан пример входных данных, программа правильно работает при большем числе случаев, чем исходная, но не при всех, и, при доработке не возникли точки, в которых исходная программа работает верно, а доработанная - нет).

2

Например, если дан правильный ответ на вопрос №1, добавлены ветки "else" ко всем условиям "if", но программа выдает

"принадлежит" для точек, у которых  $(y < \operatorname{tg} x)$  и  $(y > 0)$  и  $(x < 0)$ :

```
if y>=0 then if x<=1.57 then if y<=tan(x) then
write('принадлежит') else
write('не принадлежит') else
write('не принадлежит') else
write('не принадлежит')
```

Правильно выполнено только одно действие из трёх. То есть, только приведен пример входных данных, либо он не приведен, но имеется программа, корректно работающая при большем количестве входных данных, чем исходная, но не при всех.

1

Все пункты задания выполнены неверно (пример входных данных не указан или указан неверно, программа не приведена, либо приведенная программа корректно работает в не большем количестве случаев, чем исходная).

0

С2

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от  $-30$  до  $+30$  – значение среднесуточной температуры для каждого из 30 дней некоторого месяца. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет найти и вывести количество дней этого месяца, температура для которых была выше, чем температура в предыдущий день этого же месяца. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	Бейсик
<pre>Const N=30; var a: array [1..N] of integer; i, j: integer; begin for i:=1 to N do readln(a[i]); ... end.</pre>	<pre>N=30 DIM A(N) AS INTEGER DIM I, J AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
СИ	Естественный язык
<pre>#include &lt;stdio.h&gt; #define N 30 void main(void) {int a[N]; int i, j; for (i=0; i&lt;N; i++) scanf("%d", &amp;a[i]); ... }</pre>	<p>Объявляем массив A из 30 элементов. Объявляем целочисленные переменные I, J. В цикле от 1 до 30 вводим элементы массива A с 1-го по 30-й.</p> <p>...</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, Borland Pascal 7.0) 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).

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

Паскаль	Бейсик	
<pre>j:=0; for i:=2 to N do if A[i]&gt;A[i-1] then j:=j+1; WriteLn(j);</pre>	<pre>J=0 FOR I = 2 to N IF A(I)&gt;A(I-1) THEN J=J+1 NEXT I PRINT J</pre>	
СИ	Естественный язык	
<pre>j=0; for (i=1; i&lt;N; i++) if (a[i]&gt;a[i-1]) j=j+1; printf("%d", j);</pre>	<p>Записываем в переменную J начальное значение, равное 0. В цикле по переменной I от значения 2 до значения 30 сравниваем значения элементов массива A[i] и A[i-1]. Если A[i]&gt;A[i-1], то увеличиваем значение переменной J на 1. После завершения цикла выводим значение переменной J.</p>	
Указания по оцениванию		Баллы
<p>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.</p>		2

В любом варианте решения может присутствовать не более одной ошибки из числа следующих: 1. Не инициализируется или неверно инициализируется переменная J. 2. В сравнении элементов массива a[i] и a[i-1] вместо знака «больше» используется знак «больше или равно». 3. Отсутствует вывод ответа 4. Используется переменная, не объявленная в разделе описания переменных. 5. Не указано или неверно указано условие завершения цикла (например, цикл начинается со значения 1, а не 2, вследствие чего происходит выход за границы массива) 6. Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно. 7. Неверно расставлены операторные скобки.	1
Ошибок, перечисленных в п. 1–7, две или больше, или алгоритм сформулирован неверно.	0

**С3** Два игрока играют в следующую игру. На координатной плоскости стоит фишка. В начале игры фишка находится в точке с координатами  $(-1, 2)$ . Игроки ходят по очереди. Ход состоит в том, что игрок перемещает фишку из точки с координатами  $(x, y)$  в одну из трех точек:  $(x-1, y+4)$ ,  $(x+3, y-1)$ ,  $(x+2, y+3)$ . Игра заканчивается, как только расстояние от фишки до начала координат превысит число 11. Выигрывает игрок, который сделал последний ход. Кто выигрывает при безошибочной игре – игрок, делающий первый ход, или игрок, делающий второй ход? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)				
<b>Выигрывает первый игрок</b> , своим первым ходом он должен поставить фишку в точке с координатами $(1, 5)$ . Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке координаты фишки на каждом этапе игры.				
<b>1 ход</b>	<b>2 ход</b>	<b>3 ход</b>	<b>4 ход</b>	<b>5 ход</b>
<b>Позиция после первого хода</b>	<b>II-й игрок (все варианты хода)</b>	<b>I-й игрок (выигрышный ход)</b>	<b>II-й игрок (все варианты хода)</b>	<b>I-й игрок (выигрышные ходы)</b>
<b>1,5</b>	0,9	<b>-1,13 или 2,12</b>	Первый игрок выигрывает на 3 ходу	
	4,4	7,3	6,7	<b>5,11 или 8,10</b>
			10,2	<b>13,1 или 12,5</b>
			9,6	<b>Любой ход</b>
3,8	<b>2,12 или 5,11</b>	Первый игрок выигрывает на 3 ходу		
Таблица содержит <b>все возможные</b> варианты ходов второго игрока. Из неё видно, что при любом ответе второго игрока у первого имеется ход, приводящий к победе.				
<b>Указания по оцениванию</b>				<b>Баллы</b>
Правильное указание выигрывающего игрока и его ходов со строгим доказательством правильности (с помощью или без помощи дерева игры).				3
Правильное указание выигрывающего игрока, стратегии игры, приводящей к победе, но при отсутствии доказательства ее правильности.				2
При наличии в представленном решении одного из пунктов: 1. Правильно указаны все варианты хода первого игрока и возможные ответы второго игрока, но неверно определены дальнейшие действия и неправильно указан победитель. 2. Правильно указан выигрывающий игрок, но описание выигрышной стратегии неполно и рассмотрены несколько (больше одного, но не все) вариантов хода первого игрока и частные случаи ответов второго игрока.				1
Задание не выполнено или в представленном решении полностью отсутствует описание элементов выигрышной стратегии, и отсутствует анализ вариантов первого-второго ходов играющих (даже при наличии правильного указания выигрывающего игрока).				0

**C4** Имеется список результатов голосования избирателей за несколько партий, в виде списка названий данных партий.

На вход программе в первой строке подается количество избирателей в списке N. В каждой из последующих N строк записано название партии, за которую проголосовал данный избиратель, в виде текстовой строки. Длина строки не превосходит 50 символов, название может содержать буквы, цифры, пробелы и прочие символы.

Пример входных данных:

```
6
Party one
Party two
Party three
Party three
Party two
Party three
```

Программа должна вывести список всех партий, встречающихся в исходном списке, в порядке убывания количества голосов, отданных за эту партию. При этом название каждой партии должно быть выведено ровно один раз, вне зависимости от того, сколько голосов было отдано за данную партию.

Пример выходных данных для приведенного выше примера входных данных:

```
Party three
Party two
Party one
```

При этом следует учитывать, что количество голосов избирателей в исходном списке может быть велико (свыше 1000), а количество различных партий в этом списке не превосходит 10.

### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Программа читает все входные данные один раз, не запоминая все входные данные в массиве, размер которого равен N, а составляя только список имеющихся политических партий и количества голосов, отданных за каждую из них. Во время чтения данных просматривается список ранее сохраненных партий, если данная партия уже есть в списке, то количество голосов, отданных за эту партию, увеличивается на 1, иначе партия добавляется в список известных партий. После окончания ввода производится сортировка массивов партий и количества голосов, отданных за них в порядке убывания количества голосов, затем выводится список партий.

Баллы начисляются только за программу, которая решает задачу хотя бы для одного частного случая.

Ниже приведены примеры решения задания на языках Паскаль и C++. Допускаются решения, записанные на других языках программирования. При оценивании решений на других языках программирования необходимо учитывать особенности этих языков программирования.

### Решение на языке Паскаль

```
Var
n, NumParties, i, j, t: integer;
Count:array[1..10] of integer;
s: string;
Names: array[1..10] of string;
Begin
  {Общее число партий в списке}
  NumParties:=0;
  {Считываем количество избирателей}
  ReadLn(N);
  for i:=1 to N do
  begin
    {Считываем название партии, за которую проголосовал
    избиратель}

    ReadLn(S);
    {Осуществляем поиск считанного названия в списке всех известных партий}

    j:=1;
    while (j<=NumParties) and (s<>Names[j])
    do
      j:=j+1;
    {Если партия найдена в списке}
    if j<=NumParties
    then
      {Увеличиваем счетчик числа голосов, отданных за эту
      партию}
      Count[j]:=Count[j]+1
    else
      begin
        {Иначе добавляем партию в конец списка, увеличиваем
        число партий на 1, устанавливаем счетчик голосов, отданных за
        новую партию, в 1}
        Names[j]:=s;
        Count[j]:=1;
        NumParties:=NumParties+1
      end
    end;
  {Сортируем массивы Names и Count в порядке убывания
  значений массива Count}
  for i:=NumParties downto 2 do
```

```

for j:=2 to i do
  if Count[j-1]<Count[j]
  then
  begin
    t:=Count[j];
    Count[j]:=Count[j-1];
    Count[j-1]:=t;
    s:=Names[j];
    Names[j]:=Names[j-1];
    Names[j-1]:=s;
  end;
{Выводим содержимое массива Count}
for i:=1 to NumParties do
  WriteLn(Names[i])
end.

```

### Решение на языке C++

```

#include<iostream>

using namespace std;

int main()
{
  int n, Count[10], NumParties=0, i, j, t;
  string Names[10], s;
  cin>>n;
  getline(cin,s);
  for(i=0;i<n;++i)
  {
    getline(cin,s);
    j=0;
    while(j<NumParties && s!=Names[j])
      ++j;
    if(j<NumParties)
      ++Count[j];
    else
    {
      Names[j]=s;
      Count[j]=1;
      ++NumParties;
    }
  }
  for (i=NumParties-1;i>=1;--i)

```

```

for (j=1;j<=i;++j)
  if (Count[j-1]<Count[j])
  {
    t=Count[j];
    Count[j]=Count[j-1];
    Count[j-1]=t;
    s=Names[j];
    Names[j]=Names[j-1];
    Names[j-1]=s;
  }
for(i=0;i<NumParties;++i)
  cout<<Names[i]<<endl;
}

```

Указания по оцениванию	Баллы
Программа работает верно для любых входных данных произвольного размера и находит ответ, не сохраняя входные данные в массиве, размер которого соответствует числу N (количество человек в списке) или диапазону возможных дней рождений. Программа просматривает входные данные один раз, сохраняя данные только о всех известных партиях и количестве голосов, отданных за каждую из них. 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).	4
Программа работает верно, но все входные данные запоминаются в массиве или входные данные считываются несколько раз. Допускается наличие от одной до трех синтаксических ошибок. Три балла также выставляется, если в эффективной программе, удовлетворяющей критериям выставления 4 баллов, есть одна ошибка, например, использование неверной границы в алгоритме сортировке, выход за границы массива, используется знак "<" вместо "<=" и т.д.	3
Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, в частности значения минимума, выход за границу массива, используется знак "<" вместо "<=", "or" вместо "and" и т.п.). Возможно, некорректно организовано считывание входных данных. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	2

Программа, возможно, неверно работает при некоторых входных данных. Возможно, программа правильно подсчитывает количество всех партий, но не выполняется сортировка списка партий. Возможно, сортируется только массив количества голосов, а массив названий партий при этом не сортируется. Допускается до 4 различных ошибок в реализации алгоритма, в том числе описанных в критериях присвоения двух баллов. Допускается наличие от одной до семи синтаксических ошибок, описанных выше.	1
Задание не выполнено или выполнено неверно.	0