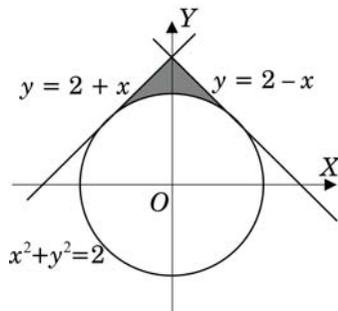


Критерии оценивания заданий с развёрнутым ответом

С1 Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Область ограничена окружностью $x^2 + y^2 = 2$ и прямыми $y = 2 + x$ и $y = 2 - x$. Окружность касается первой прямой в точке $(-1; 1)$ и второй прямой в точке $(1; 1)$. Программист торопился и написал программу неправильно.



ПРОГРАММА НА ПАСКАЛЕ	<pre>var x,y: real; begin readln(x,y); if x*x+y*y>=2 then if y<=2+x then if y<=2-x then write('принадлежит') else write('не принадлежит') end. end.</pre>
ПРОГРАММА НА БЕЙСИКЕ	<pre>INPUT x, y IF x*x+y*y>=2 THEN IF y<=2+x THEN IF y<=2-x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>
ПРОГРАММА НА СИ	<pre>void main(void) { float x,y; scanf("%f%f", &x, &y); if (x*x+y*y>=2) if (y<=2+x) if (y<=2-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: $x=0, y=0$ (Любая пара (x, y) , для которой выполняется: $x^2 + y^2 < 2$, или $y > 2 + x$, или $(y \leq 2 - x$ и $y \leq 2 + x$ и $y < 1)$)

Возможная доработка (Паскаль):

```
if (x*x+y*y>=2) and (y<=2+x) and (y<=2-x) and (y>=1) 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 $(x^2 + y^2 \geq 2)$, и $(y <= 2 + x)$, и $(y <= 2 - x)$, и $(y < 1)$.	
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). Например, если дан правильный ответ на вопрос № 1, добавлены ветки “else” ко всем условиям “if”, но программа выдает “принадлежит” для точек, у которых $(x^2 + y^2 \geq 2)$, и $(y <= 2 + x)$, и $(y <= 2 - x)$, и $(y < 1)$: if x*x+y*y>=2 then if y<=2+x then if y<=2-x then write('принадлежит') else write('не принадлежит') else write('не принадлежит') else write('не принадлежит')	2
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 0 до 100 – баллы учащихся выпускного класса за итоговый тест по информатике. Учащийся получает за работу «зачет», если он набрал за тест не менее 50 баллов.

Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет определить и вывести на экран средний балл за тест для тех учащихся, которые получили «зачет». Гарантируется, что хотя бы один из учащихся в классе получил «зачет».

Паскаль	Бейсик
<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 DIM S AS REAL FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
СИ	Естественный язык
<pre>#include <stdio.h> #define N 30 void main(void) {int a[N]; int i, j; float s; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>	<p>Объявляем массив A из 30 элементов. Объявляем целочисленные переменные I, J и переменную S вещественного типа. 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; s:=0; for i:=1 to N do if A[i]>=50 then begin j:=j+1; s:=s+A[i] end; writeln(s/j)</pre>	<pre>J=0 S=0 FOR I = 1 TO N IF A(I)>=50 THEN J=J+1 S=S+A(i) ENDIF NEXT I PRINT S/J</pre>
СИ	Естественный язык
<pre>j=0; s=0; for(i=0;i<N;i++) if(a[i]>=50) { j=j+1; s+=a[i]; } printf("%f",s/j);</pre>	<p>Подсчитаем и запишем в переменную J количество учащихся, получивших «зачет», а в переменную S – сумму баллов всех учащихся, получивших «зачет». Для этого записываем в переменную J начальное значение, равное 0? и в переменную S также значение 0. В цикле от первого элемента до тридцатого сравниваем значение текущего элемента с числом 50. Если значение текущего элемента больше или равно 50, то увеличиваем значение переменной J на 1, а значение переменной S увеличиваем на значение текущего элемента массива. После окончания цикла выводим частное от деления S на J.</p>

Указания по оцениванию	Баллы
Предложен правильный алгоритм, всегда выдающий верный ответ. 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. Текст сообщений, выводимых программой, может отличаться от указанных в условии, если это не искажает их смысла.	2
В любом варианте решения может присутствовать не более одной ошибки из числа следующих: 1) Не инициализируются или неверно инициализируются переменные J, S. 2) В сравнении со значением 50 вместо знака «больше или равно» используется знак «больше». 3) Используется переменная, не объявленная в разделе описания переменных. 4) Не указано или неверно указано условие завершения цикла. 5) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно. 6) Неверно расставлены операторные скобки. 7) Ошибка преобразования типов при использовании операции деления. 8) Отсутствует вывод ответа. 9) По окончании алгоритма производится деление S/N, а не S/J.	1
Ошибок, перечисленных в п. 1–9, 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 (0, 0). Игроки ходят по очереди. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трех точек: (x+2, y+1), (x+1, y+3), (x+3, y+4). Игра заканчивается, как только расстояние по прямой от фишки до начала координат станет не меньше 13. 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-й ход I игрок (все варианты хода)	2-й ход II игрок (выигрышный ход)	3-й ход I игрок (все варианты хода)	4-й ход II игрок (выигрышный ход)
2,1	5,5	7,6 6,8 8,9	10,10 7,11 или 9,12 Любой ход
1,3	2,6	4,7 3,9 5,10	7,11 6,13 Любой ход
3,4	5,5	Аналогичные варианты 3–4-го хода	

Таблица содержит **все возможные** 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 сотрудников организации 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 фамилию и имя самого молодого сотрудника, празднующего свой день рождения в течение ближайших семи дней от текущей даты, но в текущем календарном году.

На вход программе в первой строке подается текущая дата, заданная в формате 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.

Во второй строке подается количество людей в списке N. Значение N может быть велико, например, может быть больше 10.000. 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 в формате дд.мм.гггг. Известно, что у всех сотрудников даты рождения различаются.

Пример входных данных:

30.01.2010

3

Иванов Сергей 02.02.1974

Сергеев Петр 31.01.1965

Петров Иван 15.02.1975

Программа должна вывести фамилию и имя самого молодого сотрудника, празднующего день рождения в течение семи дней от текущей даты (включая текущую дату). Если при этом часть дней в течение семи дней от текущей даты приходится на следующий календарный год, то сотрудников, отмечающих ближайший день рождения в следующем календарном году, учитывать не надо.

Пример выходных данных для данного примера входных данных:

Иванов Сергей

Если в течение семи дней ни один сотрудник не празднует день рождения, необходимо вывести сообщение: «В ближайшую неделю дней рождения нет».

Количество дней в месяцах года (для справки): январь – 31, февраль – 28, март – 31, апрель – 30, май – 31, июнь – 30, июль – 31, август – 31, сентябрь – 30, октябрь – 31, ноябрь – 30, декабрь – 31. При решении задания не нужно учитывать наличие високосных годов.

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла.)

Создадим переменные для хранения следующей информации: текущая дата (номер дня, номер месяца), дата рождения самого молодого сотрудника, празднующего день рождения в течение ближайших семи дней (номер дня, номер месяца, номер года), его фамилия и имя.

Программа читает все входные данные один раз. Во время чтения данных определяется дата рождения каждого человека. Определяется, приходится ли день рождения этого человека на ближайшие семь дней от текущей даты (но в рамках текущего календарного года), в этом случае определяется, не является ли данный сотрудник моложе, чем ранее известный самый молодой сотрудник, и при необходимости обновляется информация о самом молодом сотруднике.

После окончания программы выводится имя самого молодого сотрудника или сообщение о том, что сотрудников, празднующих день рождения в ближайшие семь дней, нет.

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

```

Procedure StrToDate(Date:String; var Day: integer; var Month: integer; var
Year: integer);
Var code: integer;
Begin
  val (Copy (Date, 1, 2), Day, Code);
  val (Copy (Date, 4, 2), Month, Code);
  val (Copy (Date, 7, 4), Year, Code)
End;

Const
Days : array [1..12] of integer =
  (31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);
Var
N, i, Day, Month, Year, CurrDay, CurrMonth, YoungestDay, YoungestMonth,
YoungestYear: integer;
Name, YoungestName, Date: string;
Begin
  YoungestYear:=0;
  {Считываем текущую дату}
  ReadLn (Date);
  {Извлекаем из текущей даты номер дня, месяца, года рождения}
  StrToDate (Date, CurrDay, CurrMonth, Year);
  {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}
    StrToDate (Copy (Name, Length (Name) - 9, 10), Day, Month, Year);
    {Если дата приходится на ближайшие семь дней}
    if (Month=CurrMonth) and (Day>=CurrDay) and (Day<CurrDay+7) or
      (Month=CurrMonth+1) and (Day<CurrDay+7-Days[CurrMonth])
    then
      {Сравниваем дату рождения с датой рождения самого молодого
сотрудника}
      if (Year>YoungestYear) or (Year=YoungestYear) and (Month >
YoungestMonth) or
        (Year=YoungestYear) and (Month=YoungestMonth) and (Day>YoungestDay)
      then
        begin
          YoungestDay:=Day;
          YoungestMonth:=Month;
          YoungestYear:=Year;
          YoungestName:=copy (Name, 0, length (Name) - 11);
        end
      end;
    if YoungestYear>0
    then
      WriteLn (YoungestName)
    else
      WriteLn ('В ближайшую неделю дней рождения нет')
  end
end;

```

end.

Решение на языке C++

```

#include<iostream>
using namespace std;

void StrToDate(string Date, int & Day, int & Month, int & Year)
{
  Day=(Date[0]-'0')*10 + Date[1]-'0';
  Month=(Date[3]-'0')*10 + Date[4]-'0';
  Year=((Date[6]-'0')*10 + Date[7]-'0') * 10 + Date[8]-'0'*10 + Date[9]-
'0';
}

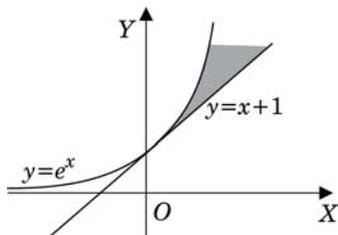
int main()
{
  int Days[13]={0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
  int N, i, Day, Month, Year, CurrDay,
  CurrMonth, YoungestDay, YoungestMonth, YoungestYear;
  string Name, YoungestName, Date;
  YoungestYear=0;
  cin>>Date;
  StrToDate (Date, CurrDay, CurrMonth, Year);
  cin >> N;
  getline (cin, Name);
  for (i=0; i<N; ++i)
  {
    getline (cin, Name);
    Date=Name.substr (Name.length () - 10);
    StrToDate (Date, Day, Month, Year);
    if ( Month==CurrMonth && Day>=CurrDay && Day<CurrDay+7 ||
      Month==CurrMonth+1 && Day < CurrDay + 7 - Days [CurrMonth])
      if (Year>YoungestYear || Year==YoungestYear && Month >
YoungestMonth ||
        Year==YoungestYear && Month==YoungestMonth && Day>YoungestDay)
      {
        YoungestDay=Day;
        YoungestMonth=Month;
        YoungestYear=Year;
        YoungestName=Name.substr (0, Name.length () - 11);
      }
  }
  if (YoungestYear>0)
    cout<<YoungestName<<endl;
  else
    cout<<"В ближайшую неделю дней рождения нет"<<endl;
  return 0;
}

```

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). Программа просматривает входные данные один раз, сохраняя только информацию о самом молодом сотруднике, празднующем день рождения в ближайшую неделю. 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 входные данные запоминаются в массиве, или входные данные считываются несколько раз. 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 (например, программа неверно сравнивает даты рождения двух сотрудников, если они родились в одном месяце одного года).	3
Программа работает в целом верно, эффективно или нет, но в реализации алгоритма содержится до двух ошибок (неверная инициализация переменных, в частности значения максимума, выход за границу массива, перевод символов в числа, 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, не выводится сообщение о том, что нет сотрудников, празднующих день рождения в течение одной недели. Возможно, неверно реализован алгоритм сравнения дат рождения сотрудников с учетом года рождения. 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

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). Область ограничена графиками функции $y = e^x$ и прямой $y = x + 1$, которые касаются в точке $(0, 1)$. Для вычисления значения e^x в языке программирования используется функция $\exp(x)$. Программист торопился и написал программу неправильно.



ПРОГРАММА НА ПАСКАЛЕ	<pre>var x, y: real; begin readln(x, y); if y >= x + 1 then if y <= exp(x) then write('принадлежит') else write('не принадлежит') end.</pre>
ПРОГРАММА НА БЕЙСИКЕ	<pre>INPUT x, y IF y >= x + 1 THEN IF y <= exp(x) THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END</pre>
ПРОГРАММА НА СИ	<pre>void main(void) { float x, y; scanf("%f%f", &x, &y); if (y >= x + 1) if (y <= exp(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: $x = 0, y = 0$ (Любая пара (x, y) , для которой выполняется: $y < x + 1$ или $(y < \exp(x)$ и $x < 0)$)

Возможная доработка (Паскаль):
 if $(y > x + 1)$ and $(y < \exp(x))$ 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 условия программа не выдавала ничего (отсутствует случай ELSE). 2. Приведенным трем ограничениям удовлетворяют также те точки плоскости, у которых $(y > x + 1)$, и $(y < \exp(x))$, 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 два 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). Например, если дан правильный ответ на вопрос № 1, добавлена ветка “else” ко всем условиям “if”, но программа выдает “принадлежит” для точек, у которых $(y > x + 1)$, и $(y < \exp(x))$, и $(x < 0)$: if $y >= x + 1$ then if $y <= \exp(x)$ then write('принадлежит') else write('не принадлежит') else write('не принадлежит')	2
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

C2 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 определить истинность утверждения о том, что температура в каждый день (за исключением первого) была не ниже, чем температура в предыдущий день (то есть среднесуточная температура на протяжении месяца не убывала). Программа должна вывести одно из двух сообщений: «Среднесуточная температура не убывала» или «Были случаи убывания среднесуточной температуры».

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 <stdio.h> #define N 30 void main(void) {int a[N]; int i, j; for (i=0; i<N; i++) scanf("%d", &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, например, допускается решение, в котором переменная J используется как признак того, что в просмотренном фрагменте массива среднесуточная температура не убывала.

Паскаль	Бейсик
<pre>i:=2; while (i<=n) and (a[i]>=a[i-1]) do i:=i+1; if i>n then writeln('Среднесуточная температура не убывала') else writeln('Были случаи убывания среднесуточной температуры')</pre>	<pre>I=2 WHILE I<=N AND A(I)>=A(I-1) I=I+1 WEND IF I>N THEN PRINT "Среднесуточная температура не убывала" ELSE PRINT "Были случаи убывания среднесуточной температуры"</pre>
СИ	Естественный язык
<pre>i=1; while (i<n && a[i]>=a[i-1]) i=i+1; if (i>n) printf("Среднесуточная температура не убывала"); else printf("Были случаи убывания среднесуточной температуры");</pre>	<p>Записываем в переменную I начальное значение, равное 2. Пока значение переменной I не превосходит 30 и значение элемента массива с индексом I не меньше значения элемента с индексом I-1, увеличиваем значение I на 1. Если после окончания цикла значение I больше 30, то выводим сообщение "Среднесуточная температура не убывала", иначе выводим сообщение "Были случаи убывания среднесуточной температуры".</p>

Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, всегда выдающий верный ответ. 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. Текст сообщений, выводимых программой, может отличаться от указанных в условии, если это не искажает их смысла.</p>	2
<p>В любом варианте решения может присутствовать не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> 1) В сравнении элементов массива $a[i]$ и $a[i-1]$ используется знак «больше» вместо «больше или равно». 2) Вместо логической операции «И» используется операция «ИЛИ». 3) Полностью или частично отсутствует вывод ответа при верном решении задачи (то есть, полностью может отсутствовать последний IF). 4) Используется переменная, не объявленная в разделе описания переменных. 5) Не указано или неверно указано условие начала или завершения цикла, вследствие чего происходит выход за границы массива или не рассматриваются некоторые элементы массива. 6) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно. 7) Неверно расставлены операторные скобки. 	1
<p>Ошибок, перечисленных в п. 1–7, 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 $(-1, -1)$. Игроки ходят по очереди. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трех точек: $(x+2, y+1)$, $(x+1, y+3)$, $(x+3, y+4)$. Игра заканчивается, как только расстояние по прямой от фишки до начала координат превысит 13. 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 первый игрок, своим первым ходом он должен поставить фишку в точке с координатами $(1, 0)$. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке координаты фишки на каждом этапе игры.

1-й ход	2-й ход	3-й ход	4-й ход	5-й ход
Позиция после первого хода	II игрок (все варианты хода)	I игрок (выигрышный ход)	II игрок (все варианты хода)	I игрок (выигрышные ходы)
1,0	3,1	6,5	8,6	11,10
			7,8	8,11 или 10,12
			9,9	Любой ход
	2,3	3,6	5,7	8,11
			4,9	7,13
			6,10	Любой ход
	4,4	6,5	Аналогичные варианты 4–5-го хода	

Таблица содержит **все возможные** варианты ходов второго игрока. Из неё видно, что при любом ответе второго игрока у первого 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 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

С4 На президентских выборах побеждает кандидат, получивший более половины от общего числа голосов избирателей, а если ни один из кандидатов не получил более половины от общего числа голосов избирателей, то во второй тур проходят два кандидата, получившие наибольшее число голосов.

Имеется список результатов голосования избирателей за кандидатов в президенты в виде фамилий кандидатов. Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет определять, кто победил на выборах или кто прошел во второй тур в случае отсутствия победителя.

На вход программе в первой строке подается количество избирателей в списке N . В каждой из последующих N строк записана фамилия кандидата, за которого отдал голос этот избиратель. Длина строки не превышает 20 символов.

Пример входных данных:

```
5
Иванов
Кузнецов
Петров
Иванов
Иванов
```

В случае, если один кандидат набрал более половины от общего числа голосов, программа должна вывести его фамилию. Если же ни один из кандидатов не набрал больше половины голосов, программа должна вывести фамилию кандидата, набравшего больше всего голосов, а затем фамилию кандидата, занявшего второе место (можно считать, что нет кандидатов, которые получили равное число голосов избирателей).

Пример выходных данных для приведенного выше примера входных данных:

```
Иванов
```

При этом следует учитывать, что количество голосов избирателей в исходном списке может быть велико (свыше 1000), а количество кандидатов 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 , составляя только список кандидатов и количества голосов, отданных за каждого из них. Во время чтения данных просматривается список ранее сохраненных фамилий кандидатов, если данный кандидат уже есть в списке, то количество голосов, отданных за него, увеличивается на 1, иначе кандидат добавляется в список известных кандидатов. После окончания ввода определяется фамилия кандидата, набравшего больше всего голосов, и выводится его фамилия. Если при этом он набрал не более $N/2$ голосов, то определяется кандидат, занявший второе место, и его фамилия выводится на экран.

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, NumCandidates, Max, i, j: integer;
Count: array[1..10] of integer;
s: string;
Names: array[1..10] of string;
Begin
    NumCandidates:=0;
```

```

{Считываем количество избирателей}
ReadLn(N);
for i:=1 to N do
begin
  {Считываем фамилию кандидата, за которого проголосовал избиратель}
  ReadLn(S);
  {Осуществляем поиск названия в списке известных кандидатов}
  j:=1;
  while (j<=NumCandidates) and (s<>Names[j])
  do
    j:=j+1;
  {Если кандидат найден в списке}
  if j<=NumCandidates
  then
    {Увеличиваем счетчик числа голосов, отданных за этого кандидата}
    Count[j]:=Count[j]+1
  else
  begin
    {Иначе добавляем фамилию кандидата в конец списка,
    увеличиваем число кандидатов на 1,
    устанавливаем счетчик голосов,
    отданных за нового кандидата, в 1}
    Names[j]:=s;
    Count[j]:=1;
    NumCandidates:=NumCandidates +1
  end
end;
{Ищем кандидата, набравшего больше половины голосов}
j:=0;
Max:=0;
for i:=1 to NumCandidates do
  if Count[i]>Max then
  begin
    Max:=Count[i];
    j:=i
  end;
{Выводим его фамилию}
WriteLn(Names[j]);
{Проверяем, нужен ли второй тур}
if Max*2<=N then
Begin
  {Обнуляем количество голосов у кандидата, занявшего первое место}
  Count[j]:=0;
  {Находим и выводим фамилию кандидата,
  набравшего следующее по величине число голосов}
  j:=0;
  Max:=0;
  for i:=1 to NumCandidates do
    if Count[i]>Max then
    begin
      Max:=Count[i];
      j:=i
    end;

```

```

  WriteLn(Names[j])
end
end.

```

Решение на языке C++

```

#include<iostream>
using namespace std;
int main()
{
  int n, Count[10], NumCandidates=0, i, j, Max;
  string Names[10], s;
  cin>>n;
  getline(cin,s);
  for(i=0;i<n;++i)
  {
    getline(cin,s);
    j=0;
    while(j<NumCandidates && s!=Names[j])
      ++j;
    if(j<NumCandidates)
      ++Count[j];
    else
    {
      Names[j]=s;
      Count[j]=1;
      ++NumCandidates;
    }
  }
  j=0;
  Max=0;
  for(i=0; i<NumCandidates; i++)
    if (Count[i]>Max)
    {
      Max=Count[i];
      j=i;
    }
  cout << Names[j] << endl;
  if (Count[j]*2 <= n)
  {
    Count[j]=0;
    j=0;
    Max=0;
    for(i=0; i<NumCandidates; i++)
      if (Count[i]>Max)
      {
        Max=Count[i];
        j=i;
      }
    cout << Names[j] << 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 голосов избирателей в списке). Программа просматривает входные данные один раз, сохраняя данные только обо всех известных кандидатах и количестве голосов, отданных за каждого 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, выход за границы массива, используется знак “>” вместо “>=” и т. д.	3
Программа работает в целом верно, эффективно или нет, но в реализации алгоритма содержится 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 работает только для одного из случаев (например, определяет только кандидата, набравшего наибольшее число голосов). 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