

Ответы к заданиям с выбором ответа

№ задания	Ответ
A1	2
A2	2
A3	4
A4	4
A5	3
A6	4
A7	4

№ задания	Ответ
A8	3
A9	2
A10	4
A11	3
A12	1
A13	3
A14	1

Ответы к заданиям с кратким ответом

№ задания	Ответ
B1	30
B2	12122
B3	55
B4	2
B5	15
B6	27
B7	51

№ задания	Ответ
B8	7
B9	14
B10	1281
B11	510
B12	16600
B13	15
B14	2

Ответы к заданиям с выбором ответа

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

№ задания	Ответ
A8	1
A9	4
A10	2
A11	4
A12	1
A13	4
A14	3

Ответы к заданиям с кратким ответом

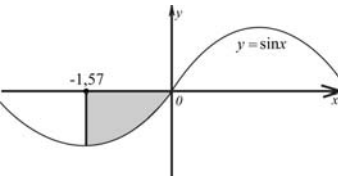
№ задания	Ответ
B1	192
B2	11122
B3	165
B4	163
B5	3
B6	-15
B7	83

№ задания	Ответ
B8	3
B9	12
B10	10241
B11	2046
B12	260
B13	8
B14	2

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 точки закрашенной области, включая ее границы. Область ограничена графиком  $y=\sin x$  и прямыми  $y=0$  и  $x=-1,57$ . 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&lt;=0 then   if x&gt;=-1.57 then   if y&gt;=sin(x) then   write     ('принадлежит')   else   write('не     принадлежит')   end .</pre>	<pre>INPUT x, y IF y&lt;=0 THEN IF x&gt;=-1.57 THEN IF y&gt;=SIN(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&lt;=0)   if (x&gt;=-1.57)   if (y&gt;=sin(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. Объясните, почему для указанных чисел программа неверно решает поставленную задачу.

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

Элементы ответа:

1. Например,  $x=0, y=1$ . Подойдет любая точка, у которой  $y>0$  или  $x<-1.57$  или  $(y<=0$  и  $y>=sin(x)$  и  $x>0)$ .

2. Возможная доработка (Паскаль):

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

Возможны и другие способы решения.

Содержание критерия	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 $(y<=0)$ и $(x>=-1.57)$ и $(y>=sin(x))$ 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, добавлены ветки “else” ко всем условиям “if”, но программа выдает “принадлежит” для точек, у которых $(y<=0)$ и $(x>=-1.57)$ и $(y>=sin(x))$ и $(x>=0)$ <pre>if (y&lt;=0) then if x&gt;=-1.57 then if y&gt;=sin(x) then write('принадлежит') else write('не принадлежит') else write('не принадлежит') else write('не принадлежит')</pre>	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
Максимальный балл	3

C2

Дан целочисленный массив из 30 элементов. Опишите на русском языке или на одном из языков программирования алгоритм подсчета суммы всех отрицательных элементов заданного целочисленного массива размером 30 элементов. Если отрицательных элементов нет, сообщите об этом.

<b>Паскаль</b>	<b>Бейсик</b>
Const N=30; var a: array [1..N] of integer; i,m,s: integer; begin for i:=1 to N do readln(a [i]); ... end.	N=30 DIM A(N) AS INTEGER DIM I, M, S AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END
<b>СИ</b>	<b>Естественный язык</b>
#include <stdio.h> #define N 30 void main(void) {int a[N]; int i, m, s; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }	Объявляем массив A из 30 эле- ментов. Объявляем целочис- ленные переменные I, m, s. В цикле от 1 до 30 вводим элементы массива A с 1-го по 30-й. ...

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).

Решение на естественном языке:

Присвоим s и m значение ноль. В цикле от первого (нулевого) до последнего (N или N-1) значения индекса совершим следующие действия:  
Сравним с нулем значение текущего элемента массива. В случае, если значение текущего элемента меньше нуля, прибавим к текущему значению переменной s значение рассматриваемого элемента массива и увеличим значение переменной m на единицу.

По завершении цикла сравним значение переменной m с нулем (проверка наличия отрицательных элементов) и в случае, если значение m больше нуля, выводим значение переменной s, в ином случае выводим «Отрицательных элементов нет».  
Замечание: можно обойтись без переменной m. В этом случае сообщение «Отрицательных элементов нет» выводится в случае, если значение s равно нулю.

Примеры записи алгоритма на языках программирования:

<b>Бейсик</b>	<b>Паскаль</b>	<b>Си</b>
S=0 M=0 FOR I=1 TO N IF A(I) < 0 THEN S = S + A(I) M = M + 1 ENDIF NEXT I IF M > 0 THEN PRINT S ELSE PRINT «Отрицательных элементов нет» ENDIF END	s := 0; m := 0; for i:= 1 to N do if a[i] < 0 then begin s := s + a[i]; m := m + 1; end; if m > 0 then write (s) else write('Отрицательных элементов нет '); end.	s = 0; m = 0; for (int i=0;i<N;i++) { if (array [i]<0) { s+=array[i]; m++; } } if (m>0) printf (s); else printf ("Отрицательных элементов нет"); }

Содержание критерия	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 M и S. 2. В сравнении со значением 0 вместо знака «меньше» используется знак «меньше или равно». 3. Отсутствует вывод ответа, когда нет отрицательных элементов. 4. Используется переменная, не объявленная в разделе описания переменных. 5. Не указано или неверно указано условие завершения цикла 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
Максимальный балл	2

C3

У исполнителя Удвоитель две команды, которым присвоены номера:

- прибавь 1,
- умножь на 2.

Первая из них увеличивает число на экране на 1, вторая – удваивает его.

Программа для Удвоителя – это последовательность команд.

Сколько есть программ, которые число 1 преобразуют в число 16?

Ответ обоснуйте

Обозначим  $R(n)$  – количество программ, которые преобразуют число 1 в число  $n$ . Обозначим  $t(n)$  наибольшее кратное двум, не превосходящее  $n$ . Обе команды исполнителя увеличивают исходное число, поэтому общее количество команд в программе не может превосходить 15.

Верны следующие соотношения:  
1. Если  $n$  не делится на 2, то тогда  $R(n) = R(t(n))$ , так как существует единственный способ получения  $n$  из  $t(n)$  – прибавлением единицы.  
2. Пусть  $n$  делится на 2.  
Тогда  $R(n) = R(n/2) + R(n-1) = R(n/2) + R(n-2)$  (если  $n > 2$ ).  
При  $n=2$   $R(n) = 2$  (два способа: прибавлением единицы или однократным умножением на 2).  
Поэтому достаточно по индукции вычислить значения  $R(n)$  для всех чисел, кратных двум и не превосходящих 20.  
Имеем:  
 $R(2)=2$   
 $R(3) = 2$   
 $R(4)=R(2)+R(2)=4=R(5)$   
 $R(6) = R(3)+R(4) = 2+4 = 6 = R(7)$   
 $R(8) = R(4)+R(6) = 4+6 = 10 = R(9)$   
 $R(10) = R(5)+R(8) = 4+10 = 14 = R(11)$   
 $R(12) = R(6)+R(10) = 6+14 = 20 = R(13)$   
 $R(14) = R(7)+R(12) = 6+20 = 26 = R(15)$   
 $R(16) = R(8)+R(14) = 10+26 = 36$   
**Ответ: 36**

Критерии оценивания	Баллы
Правильное указание количества возможных программ со строгим доказательством правильности (приведенным выше способом или любым другим).	3
Правильное указание количества возможных программ, основанное на верных рассуждениях, но доказательство правильности неполно. В частности, оценка в 2 балла выставляется в случае, если не доказано отсутствие других программ, кроме приведенных (в случае решения методом полного перебора).	2
Представленное решение обладает одним из свойств 1. Указано, что нужно рассматривать значения $n$ , меньшие, чем 16, и приведены правильные рекуррентные соотношения (см. выше), возможно, неполные. 2. Правильно выписаны и обоснованы значения $R(n)$ для небольших $n$ . 3. Правильно написан ответ, но нет никакого обоснования.	1
Не выполнено ни одно из перечисленных выше условий.	0
Максимальный балл	3

**C4**

На вход программе подаются сведения о номерах школ учащихся, участвовавших в олимпиаде. В первой строке сообщается количество учащихся  $N$ , каждая из следующих  $N$  строк имеет формат: <Фамилия> <Инициалы> <номер школы>, где <Фамилия> – строка, состоящая не более чем из 20 символов, <Инициалы> – строка, состоящая из 4-х символов (буква, точка, буква, точка), <номер школы> – не более чем двузначный номер. <Фамилия> и <Инициалы>, а также <Инициалы> и <номер школы> разделены одним пробелом. Пример входной строки:

Иванов П.С. 57

Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет выводить на экран информацию, из какой школы было больше всего участников (таких школ может быть несколько). Также программа должна подсчитать общее количество школ, приславших больше всего участников.

Следует учитывать, что  $N \geq 1000$ .

Программа верно читает входные данные, не запоминая их все, а сразу подсчитывая в массиве, хранящем 99 целых чисел согласно номерам школ, количество участников олимпиады из каждой школы. Затем ищется наибольший элемент в данном массиве, затем распечатываются номера соответствующих школ, одновременно подсчитывая их количество.

```
var nc:array[1..99] of integer;
  p:1..99;
  c:char;
  i, k, N, max: integer;
begin
  readln(N);
  for i:=0 to 99 do nc[i]:=0;
  for i:=1 to N do
    begin
      repeat
        read(c)
      until c=' '; {считана фамилия}
      repeat
        read(c)
      until c=' '; {считаны инициалы}
      readln(p);
      nc[p]:=nc[p]+1;
    end;
  max:=0;
  for i:=1 to 99 do
    if nc[i]>max then max:=nc[i];
  k:=0;
  for i:=1 to 99 do
```

```
  if nc[i]=max then
    begin
      writeln(i);
      k:=k+1
    end;
  writeln('Количество школ приславших наибольшее число
участников', k)
end.
```

Критерии оценивания	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 учащихся в списке). Программа просматривает входные данные один раз, подсчитывая для каждой школы количество учащихся в ней. 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, в результате которой программа работает не верно на некоторых (не типичных) наборах входных данных (например, все даты рождения совпадают).	3
Программа работает в целом верно, эффективно или нет, но, в реализации алгоритма содержатся до двух ошибок (неверная инициализация переменных, 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, выводится только один день, в который родилось наибольшее количество людей, а не несколько таких дней. 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
<i>Максимальный балл</i>	4