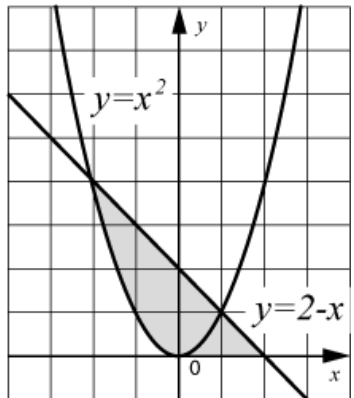


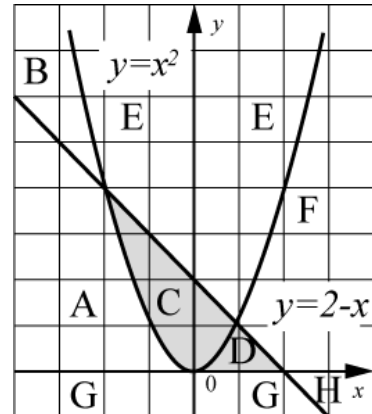
Критерии оценивания заданий с развёрнутым ответом

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	Паскаль
<pre> INPUT x, y IF y<=2-x THEN IF y>=0 THEN IF y>=x*x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END </pre>	<pre> var x,y: real; begin readln(x,y); if y<=2-x then if y>=0 then if y>=x*x then write('принадлежит') else write('не принадлежит') end end end end. </pre>
Си	Алгоритмический язык
<pre> void main(void) { float x,y; scanf("% f % f", &x,&y); if (y<=2-x) if (y>=0) if (y>=x*x) printf("принадлежит"); else printf("не принадлежит"); } </pre>	<pre> алг нач вещ x,y ввод x,y если y<=2-x то если y>=0 то если y>=x*x 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). Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ($y \leq 2-x$)	Условие 2 ($y \geq 0$)	Условие 3 ($y \geq x*x$)	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 ведет 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 ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

Элементы ответа:

1.

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

2. Возможная доработка (Паскаль):

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

Возможны и другие способы доработки.

Пример:

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

Обратите внимание! В задаче требовалось выполнить три действия: указать для каждой области, как будет работать программа, что она выведет на экран и правильно ли это (в виде таблицы), и исправить две ошибки.

Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.

1. Верное заполнение предложенной таблицы.

2. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE). Исправлением этой ошибки может быть либо добавление случая ELSE к каждому условию IF, либо объединение всех условий IF в одно при помощи конъюнкции.

В сложных случаях это действие считается выполненным, если программа выдает одно из двух сообщений «принадлежит» или «не принадлежит» для любых чисел x и y , при этом программа не стала работать хуже, чем раньше, то есть для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.

3. Приведенным трем ограничениям не удовлетворяют точки плоскости, у которых $y < x^2$, $y \leq 2-x$, $x > 0$. Исправлением этой ошибки может быть

разбиение области на две части и использование дизъюнкции либо отбрасывание от большей области ее части.

В сложных случаях это действие считается 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.

Указания по оцениванию	Баллы
Правильно выполнены оба пункта задания. Исправлены две ошибки. Программа для всех пар чисел x , y верно определяет принадлежность точки закрашенной области. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.	3
1. Правильно выполнены два действия из трех (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 трех, 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 или наоборот).	1
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
<i>Максимальный балл</i>	3

Информатика. 11 класс. Вариант 1

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 сумму элементов массива, кратных тринадцати. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого делится на тринадцать.

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

C2

Паскаль	Алгоритмический язык
<pre>const N=30; var a: array [1..N] of integer; i, j, s: integer; begin for I := 1 to N do readln(a[i]); ... end.</pre>	<pre>алг нач цел N=30 целтаб a[1:N] цел i, j, s нц для i от 1 до N ввод a[i] кц ... кон</pre>
Бейсик	СИ
<pre>N=30 DIM A(N) AS INTEGER DIM I, J, S AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>#include <stdio.h> #define N 30 void main(void){ int a[N]; int i, j, s; for (i=0; i<N; i++) scanf("%d", &a[i]); ... }</pre>
Естественный язык	
<p>Объявляем массив A из 30 элементов. Объявляем целочисленные переменные I, J, S. В цикле от 1 до 30 вводим элементы массива A 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

Информатика. 11 класс. Вариант 1

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)

На языке Паскаль
<pre>s:=0; for i:=1 to N do if (a[i] mod 13=0) then s:=s+a[i]; writeln(s);</pre>
На алгоритмическом языке
<pre>s:=0 нц для i от 1 до N если mod(a[i],13)=0 то s:=s+a[i] все кц вывод s</pre>
На языке Бейсик
<pre>S = 0 FOR I = 1 TO N IF A(I) MOD 13=0 THEN S = S + A(I) ENDIF NEXT I PRINT S</pre>
На языке СИ
<pre>s=0; for (i=0; i<N; i++) if (a[i]%13=0) s+=a[i]; printf("%d", s);</pre>
На естественном языке
<p>Записываем в переменную S начальное значение, равное 0. В цикле от первого элемента до тридцатого находим остаток от деления элемента исходного массива на тринадцать. Если остаток от деления на тринадцать равен нулю, то считаем сумму текущего элемента массива и значения переменной S. Результат сложения сохраняем в переменную S. Переходим к следующему элементу массива. После завершения цикла выводим значение переменной S.</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 S. 2) Неверно осуществляется проверка на делимость на тринадцать. 3) На делимость на 13 проверяется не значение элемента, а его индекс. 4) Неверно осуществляется накопление суммы в цикле (например, s:=a[i]) 5) Отсутствует вывод ответа. 6) Используется переменная, не объявленная в разделе описания переменных. 7) Не указано или неверно указано условие завершения цикла. 8) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно. Неверно расставлены операторные скобки.	1
Ошибок, перечисленных в п. 1-9, две или больше, или алгоритм сформулирован неверно.	0
<i>Максимальный балл</i>	2

С3

У исполнителя Увеличитель две команды, которым присвоены номера:

1. прибавь 1,**2. умножь на 4.**

Первая из них увеличивает число на экране на 1, вторая – умножает его на 4.

Программа для Увеличителя – это последовательность команд.

Сколько есть программ, которые число 1 преобразуют в число 32?

Ответ обоснуйте.

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

Обозначим $R(n)$ – количество программ, которые преобразуют число 1 в число n . Обозначим $t(n)$ наибольшее кратное четырем, не превосходящее n .

Обе команды исполнителя увеличивают исходное число, поэтому общее количество команд в программе не может превосходить 32.

Верны следующие соотношения:

- Если n не делится на 4, то тогда $R(n) = R(t(n))$, так как существует единственный способ получения n из $t(n)$ – прибавлением единиц.

- Пусть n делится на 4.

Тогда $R(n) = R(n/4) + R(n-1) = R(n/4) + R(n-4)$ (если $n > 4$).

При $n=4$ $R(n) = 2$ (два способа: прибавлением трех единиц или однократным умножением на 4).

Поэтому достаточно по индукции вычислить значения $R(n)$ для всех чисел, кратных четырем и не превосходящих 32.

Имеем:

$$R(1) = R(2) = R(3) = 1$$

$$R(4) = 2 = R(5) = R(6) = R(7)$$

$$R(8) = R(2) + R(7) = 1 + 2 = 3 = R(9) = R(10) = R(11)$$

$$R(12) = R(3) + R(11) = 1 + 3 = 4 = R(13) = R(14) = R(15)$$

$$R(16) = R(4) + R(15) = 2 + 4 = 6 = R(17) = R(18) = R(19)$$

$$R(20) = R(5) + R(19) = 2 + 6 = 8 = R(21) = R(22) = R(23)$$

$$R(24) = R(6) + R(23) = 2 + 8 = 10 = R(25) = R(26) = R(27)$$

$$R(28) = R(7) + R(27) = 2 + 10 = 12 = R(29) = R(30) = R(31)$$

$$R(32) = R(8) + R(31) = 3 + 12 = 15$$

Ответ: 15

Другой способ решения

Будем решать поставленную задачу последовательно для чисел 1, 2, 3, ..., 32 (то есть для каждого из чисел определим, сколько программ исполнителя существует для его получения). Количество программ, которые преобразуют число 1 в число n , будем обозначать через $R(n)$. Число 1 у нас уже есть, значит, его можно получить с помощью “пустой” программы. Любая

непустая программа увеличит исходное число, т.е. даст число, больше 1. Значит, $R(1) = 1$. Для каждого следующего числа рассмотрим, из какого числа оно может быть получено за одну команду исполнителя. Если число не делится на четыре, то оно может быть получено только из предыдущего с помощью команды **прибавь 1**. Значит, количество искомых программ для такого числа равно количеству программ для предыдущего числа: $R(i) = R(i-1)$. Если число на 4 делится, то вариантов последней команды два: **прибавь 1** и **умножь на 4**, тогда $R(i) = R(i-1) + R(i/4)$. Заполним соответствующую таблицу по приведенным формулам слева направо:

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

При этом ячейки, относящиеся к числам, которые не делятся на 4, можно в решении и опустить (за исключением первого и последнего чисел):

1	4	8	12	16	20	24	28	32
1	2	3	4	6	8	10	12	15

Указания по оцениванию	Баллы
Правильное указание количества возможных программ со строгим доказательством правильности (одним из приведенных выше способов или любым другим)	3
1. Два балла ставятся в одном из двух случаев: 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 приведенных. 2. Приведены правильные и строгие рассуждения, доведенные до конца, но в вычислениях допущена арифметическая ошибка, в результате чего получен неверный ответ	2
Представленное решение обладает одним из свойств 1. Указано, что нужно рассматривать значения n , меньшие, чем 32, 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 его обоснования	1
Не выполнено ни одно из перечисленных выше условий	0
<i>Максимальный балл</i>	3

С4

По каналу связи передается последовательность положительных целых чисел, все числа не превышают 1000, их количество заранее неизвестно. Каждое число передается отдельно. Признаком конца передаваемой последовательности является число 0. После числа 0 передается контрольное значение – наибольшее число R , удовлетворяющее следующим условиям:

- 1) R – произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел, произведения различных, но равных по величине элементов допускаются);
- 2) R делится на 6

Напишите эффективную программу, которая получает последовательность чисел и следующие за ней признак конца и контрольное значение, а также проверяет правильность контрольного значения. Программа должна напечатать отчет по следующей форме:

Получено .. чисел

Полученное контрольное значение:

Вычисленное контрольное значение: ...

Контроль не пройден (или – контроль не пройден)

Размер памяти, которую использует Ваша программа, не должен зависеть от длины переданной последовательности чисел.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

Пример входных данных:

60
17
3
7
9
60
0
3600

Пример выходных данных для приведенного выше примера входных данных:

Получено 6 чисел

Полученное контрольное значение: 3600

Вычисленное контрольное значение: 3600

Контроль пройден.

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

Произведение двух чисел делится на 6, если

- один из сомножителей делится на 6 (второй может быть любым) либо
- ни один из сомножителей не делится на 6, причем один из сомножителей делится на 2, а другой – на 3..

Поэтому программа, вычисляющая кодовое число может работать так.

Программа читает все входные данные один раз, не запоминая все данные в массиве. Программа для прочитанного фрагмента входной последовательности хранит значения четырех величин:

- M2 – самое большое четное число;
- M3 – самое большое число кратное трем;
- M6 – самое большое число кратное шести;
- MAX – самое большое число, среди всех элементов последовательности (если число M6 встретилось более одного раза или оно не является максимальным среди всех прочитанных чисел) или самое большое число, отличное от M6 (в противном случае).

После того, как все данные прочитаны, искомое кодовое слово вычисляется, как максимум из произведений M6*MAX и M2*M3.

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

Пример правильной и эффективной программы на языке Паскаль:

```
program C4;

var
M2,M3,M6,R,count,MAX,dat,res: integer;

begin
M2:=0;
M3:=0;
M6:=0;
MAX:=0;
count:=0;

readln(dat);

while dat>0 do
begin
if ((dat mod 2)=0) and (dat>M2) then
```

```
M2:=dat
else
if ((dat mod 3)=0)and (dat>M3) then
M3:=dat;

if (dat mod 6=0) and (dat>M6) then
begin
if m6>=max then max:=M6;
M6:=dat
end
else
if dat>max then
max:=dat;
count:=count+1;
readln(dat);

end;

readln(R);

if (m2*m3<m6*max) then
res:=m6*max
else
res:=m2*m3;

writeln('Получено чисел: ',count);
writeln('Полученное контрольное значение: ',R);
writeln('Вычисленное контрольное значение: ',res);

if R=res then
writeln('Контроль пройден.')
else
writeln('Контроль не пройден.');
```

Пример правильной и эффективной программы на Алгоритмическом языке:

```
алг C4
нач
цел M2,M3,M6,R,count,MAX
цел число
M2:=0
M3:=0
M6:=0
```

```

МАХ:=0
count:=0
ввод число
нц пока число>0
  если mod(число,2)=0 и число>M2
    то M2:=число
    иначе
      если mod(число,3)=0 и число>M3
        то M3:=число все
      все
    если mod(число,6)=0 и число>M6
      то
        если M6>=МАХ то МАХ:=M6 все
        M6:=число
      иначе
        если число>МАХ
          то МАХ:=число все
    все
  ввод число
  count:=count+1
кц
цел отв
ввод R
если M2*M3<M6*МАХ то
  отв:= M6*МАХ
иначе
  отв:=M3*M2
все
вывод "Получено чисел: ",count, нс
вывод "Полученное контрольное значение: ", R,нс
вывод "Вычисленное контрольное значение: ",
отв,нс
если отв=R то
  вывод " Контроль пройден.
иначе
  вывод "Контроль не пройден."
все
кон

```

Пример правильной и эффективной программы на языке Бейсик:

```

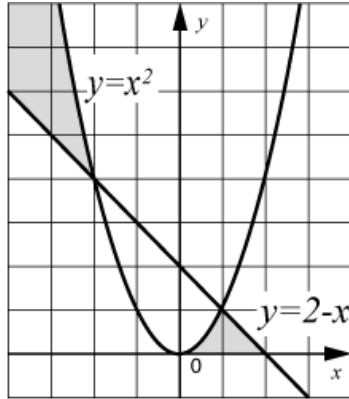
M6 = 0
M2 = 0
M3 = 0
COUNT = 0
R = 0
RES = 0
INPUT DAT
WHILE DAT > 0
  IF DAT MOD 2 = 0 AND DAT > M2 THEN
    M2 = DAT
  ELSE
    IF DAT MOD 3 = 0 AND DAT > M3 THEN
      M3 = DAT
    END IF
  END IF
  IF DAT MOD 6 = 0 AND DAT > M6 THEN
    IF M6 >=MAX THEN
      MAX = M6
    END IF
    M6 = DAT
  ELSE
    IF DAT > MAX THEN
      MAX = DAT
    END IF
  END IF
  COUNT = COUNT + 1
  INPUT "", DAT
WEND
INPUT R
IF M3 * M2 < M6 * MAX THEN
  RES = M6 * MAX
ELSE
  RES = M3 * M2
END IF
PRINT "Получено чисел:"; COUNT
PRINT "Полученное контрольное значение:"; R
PRINT "Вычисленное контрольное значение:"; RES
IF RES = R THEN
  PRINT " Контроль пройден."
ELSE
  PRINT "Контроль не пройден."
END IF

```

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).	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 при инициализации цикла анализа массива данных или обработке конца массива. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	2
Программа работает неверно, но в ней вычисляются значения M2, M3 и M6. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	1
Прочее	0
<i>Максимальный балл</i>	4

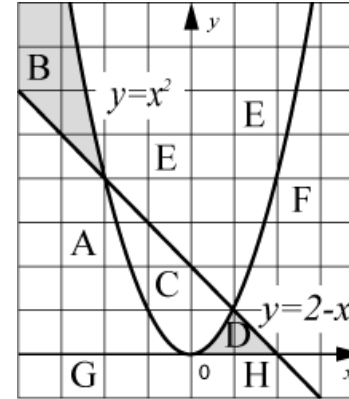
С1

Критерии оценивания заданий с развёрнутым ответом



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

Бейсик	Паскаль
<pre> INPUT x, y IF y>=0 THEN IF y>=2-x THEN IF y<=x*x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END </pre>	<pre> var x,y: real; begin readln(x,y); if y>=0 then if y>=2-x then if y<=x*x then write('принадлежит') else write('не принадлежит') end if end if end if end. </pre>
Си	Алгоритмический язык
<pre> void main(void){ float x,y; scanf("%f%f",&x,&y); if (y>=0) if (y>=2-x) if (y<=x*x) printf("принадлежит"); else printf("не принадлежит"); } } </pre>	<pre> алг нач вещ x,y ввод x,y если y>=0 то если y>=2-x то если y<=x*x 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).

Точки, лежащие на границах областей, отдельно не рассматривать.

Область	Условие 1 ($y \geq 0$)	Условие 2 ($y \geq 2-x$)	Условие 3 ($y \leq x*x$)	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 ведет 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 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)**

1.

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

2. Возможная доработка (Паскаль):

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

Возможны и другие способы доработки.

Пример:

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

Обратите внимание! В задаче требовалось выполнить три действия: указать для каждой области, как будет работать программа, что она выведет на экран и правильно ли это (в виде таблицы), и исправить две ошибки.

Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.

1. Верное заполнение предложенной таблицы.

2. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE). Исправлением этой ошибки может быть либо добавление случая ELSE к каждому условию IF, либо объединение всех условий IF в одно при помощи конъюнкции.

В сложных случаях это действие считается выполненным, если программа выдает одно из двух сообщений «принадлежит» или «не принадлежит» для любых чисел x и y , при этом программа не стала работать хуже, чем раньше,

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

3. Приведенным трем ограничениям не удовлетворяют точки плоскости, у которых $y < x^2$, $y \leq 2-x$, $y > 0$, $x > 0$. Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции либо отбрасывание от большей области ее части.

В сложных случаях это действие считается 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.

Указания по оцениванию	Баллы
Правильно выполнены оба пункта задания. Исправлены две ошибки. Программа для всех пар чисел x , y верно определяет принадлежность точки закрашенной области. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.	3
1. Правильно выполнены два действия из трех (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. Или выполнены все три действия, но при этом в логическом выражении неверно учтены приоритеты логических операций (не расставлены или неправильно расставлены скобки в выражениях).	2
Правильно выполнено только одно действие из трех, 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 или наоборот).	1
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
<i>Максимальный балл</i>	3

С2

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 сумму элементов массива, кратных семнадцати. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого делится на семнадцать.

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

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

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

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).

На языке Паскаль
<pre>s:=0; for i:=1 to N do if (a[i] mod 17=0) then s:=s+a[i]; writeln(s);</pre>
На алгоритмическом языке
<pre>s:=0 нц для i от 1 до N если mod(a[i],17)=0 то s:=s+a[i] все кц вывод s</pre>
На языке Бейсик
<pre>S = 0 FOR I = 1 TO N IF A(I) MOD 17=0 THEN S = S + A(I) ENDIF NEXT I PRINT S</pre>
На языке СИ
<pre>s=0; for (i=0; i<N; i++) if (a[i]%17=0) s+=a[i]; printf("%d", s);</pre>
На естественном языке
<p>Записываем в переменную S начальное значение, равное 0. В цикле от первого элемента до тридцатого находим остаток от деления элемента исходного массива на семнадцать. Если остаток от деления на семнадцать равен нулю, то считаем сумму текущего элемента массива и значения переменной S. Результат сложения сохраняем в переменную S. Переходим к следующему элементу массива. После завершения цикла выводим значение переменной S.</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 S. 2) Неверно осуществляется проверка на делимость на семнадцать. 3) На делимость на 17 проверяется не значение элемента, а его индекс. 4) Неверно осуществляется накопление суммы в цикле (например, s:=a[i]) 5) Отсутствует вывод ответа. 6) Используется переменная, не объявленная в разделе описания переменных. 7) Не указано или неверно указано условие завершения цикла. 8) Индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно. Неверно расставлены операторные скобки.	1
Ошибок, перечисленных в п. 1-9, две или больше, или алгоритм сформулирован неверно.	0
<i>Максимальный балл</i>	2

- С3** У исполнителя Увеличитель две команды, которым присвоены номера:
1. прибавь 1,
2. умножь на 4.
 Первая из них увеличивает число на экране на 1, вторая – умножает его на 4.
 Программа для Увеличителя – это последовательность команд.
 Сколько есть программ, которые число 1 преобразуют в число 29?
 Ответ обоснуйте.

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

Обозначим $R(n)$ – количество программ, которые преобразуют число 1 в число n . Обозначим $t(n)$ наибольшее кратное четырем, не превосходящее n .

Обе команды исполнителя увеличивают исходное число, поэтому общее количество команд в программе не может превосходить 28.

Верны следующие соотношения:

- Если n не делится на 4, то тогда $R(n) = R(t(n))$, так как существует единственный способ получения n из $t(n)$ – прибавлением единиц.
- Пусть n делится на 4.

Тогда $R(n) = R(n/4) + R(n-1) = R(n/4) + R(n-4)$ (если $n > 4$).

При $n=4$ $R(n) = 2$ (два способа: прибавлением трех единиц или однократным умножением на 4).

Поэтому достаточно по индукции вычислить значения $R(n)$ для всех чисел, кратных четырем и не превосходящих 29.

Имеем:

$$R(1) = R(2) = R(3) = 1$$

$$R(4) = 2 = R(5) = R(6) = R(7)$$

$$R(8) = R(2) + R(7) = 1 + 2 = 3 = R(9) = R(10) = R(11)$$

$$R(12) = R(3) + R(11) = 1 + 3 = 4 = R(13) = R(14) = R(15)$$

$$R(16) = R(4) + R(15) = 2 + 4 = 6 = R(17) = R(18) = R(19)$$

$$R(20) = R(5) + R(19) = 2 + 6 = 8 = R(21) = R(22) = R(23)$$

$$R(24) = R(6) + R(23) = 2 + 8 = 10 = R(25) = R(26) = R(27)$$

$$R(28) = R(7) + R(27) = 2 + 10 = 12 = R(29)$$

Ответ: 12

Другой способ решения

Будем решать поставленную задачу последовательно для чисел 1, 2, 3, ..., 29 (то есть для каждого из чисел определим, сколько программ исполнителя существует для его получения). Количество программ, которые преобразуют число 1 в число n , будем обозначать через $R(n)$. Число 1 у нас уже есть, значит, его можно получить с помощью “пустой” программы. Любая непустая программа увеличит исходное число, т.е. даст число, больше 1. Значит, $R(1) = 1$. Для каждого следующего числа рассмотрим, из какого числа оно может быть получено за одну команду исполнителя. Если число не делится на четыре, то оно может быть получено только из предыдущего с помощью команды **прибавь 1**. Значит, количество искомых программ для такого числа равно количеству программ для предыдущего числа: $R(i) = R(i-1)$. Если число на 4 делится, то вариантов последней команды два:

прибавь 1 и умножь на 4, тогда $R(i) = R(i-1) + R(i/4)$. Заполним соответствующую таблицу по приведенным формулам слева направо:

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

При этом ячейки, относящиеся к числам, которые не делятся на 4, можно в решении и опустить (за исключением первого и последнего чисел):

1	4	8	12	16	20	24	28,29
1	2	3	4	6	8	10	12

Указания по оцениванию	Баллы
Правильное указание количества возможных программ со строгим доказательством правильности (одним из приведенных выше способов или любым другим).	3
<p>Два балла ставятся в одном из двух случаев:</p> <ol style="list-style-type: none"> 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 приведенных. Приведены правильные и строгие рассуждения, доведенные до конца, но в вычислениях допущена арифметическая ошибка, в результате чего получен неверный ответ. 	2
<p>Представленное решение обладает одним из свойств</p> <ol style="list-style-type: none"> Указано, что нужно рассматривать значения n, меньшие, чем 29, и приведены правильные рекуррентные соотношения (см. выше), возможно, неполные. Правильно выписаны и обоснованы значения $R(n)$ для небольших n. Правильно написан ответ, но нет его обоснования. 	1
Не выполнено ни одно из перечисленных выше условий	0
<i>Максимальный балл</i>	3

C4 По каналу связи передается последовательность положительных целых чисел, все числа не превышают 1000, их количество заранее неизвестно. Каждое число передается отдельно. Признаком конца передаваемой последовательности является число 0. После числа 0 передается контрольное значение – наибольшее число R , удовлетворяющее следующим условиям:

1) $R - R$ – произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел, произведения различных, но равных по величине элементов допускаются);

2) R делится на 10

Напишите эффективную программу, которая получает последовательность чисел и следующие за ней признак конца и контрольное значение, а также проверяет правильность контрольного значения. Программа должна напечатать отчет по следующей форме:

Получено .. чисел

Полученное контрольное значение:

Вычисленное контрольное значение: ...

Контроль не пройден (или – контроль не пройден).

Размер памяти, которую использует Ваша программа, не должен зависеть от длины переданной последовательности чисел.

Перед текстом программы кратко опишите используемый вами алгоритм решения задачи.

Пример входных данных:

100
16
3
7
25
100
0
400

Пример выходных данных для приведенного выше примера входных данных:

Получено 6 чисел

Полученное контрольное значение: 400

Вычисленное контрольное значение: 10000

Контроль не пройден.

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

Произведение двух чисел делится на 10, если

- один из сомножителей делится на 10 (второй может быть любым) либо
- ни один из сомножителей не делится на 10, причем один из сомножителей делится на 2, а другой – на 5.

Поэтому программа, вычисляющая кодовое число может работать так.

Программа читает все входные данные один раз, не запоминая все данные в массиве. Программа для прочитанного фрагмента входной последовательности хранит значения четырех величин:

- M2 – самое большое четное число;
- M5 – самое большое число кратное пяти;
- M10 – самое большое число кратное десяти;
- MAX – самое большое число, среди всех элементов последовательности (если число M6 встретилось более одного раза или оно не является максимальным среди всех прочитанных чисел) или самое большое число, отличное от M6 (в противном случае).

После того, как все данные прочитаны, искомое кодовое слово вычисляется, как максимум из произведений $M6 \cdot MAX$ и $M2 \cdot M3$.

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

Пример правильной и эффективной программы на языке Паскаль:

```

program C4;

var
M2,M5,M10,R,count,MAX,dat,res: integer;

begin
M2:=0;
M5:=0;
M10:=0;
MAX:=0;
count:=0;

readln(dat);

while dat>0 do
begin
if ((dat mod 2)=0) and (dat>M2) then
M2:=dat

```

```

else
if ((dat mod 5)=0)and (dat>M5) then
M5:=dat;

if (dat mod 10=0) and (dat>M10) then
begin
if m10>=max then max:=M10;
M10:=dat
end
else
if dat>max then
max:=dat;
count:=count+1;
readln(dat);

end;

readln(R);

if (m2*m5<m10*max) then
res:=m10*max
else
res:=m2*m5;

writeln('Получено чисел: ',count);
writeln('Полученное контрольное значение: ',R);
writeln('Вычисленное контрольное значение: ',res);

if R=res then
writeln('Контроль пройден.')
else
writeln('Контроль не пройден.');
```

end.

Пример правильной и эффективной программы на Алгоритмическом языке:

```

алг U4
нач
цел M2,M5,M10,R,count,MAX
цел число
M2:=0
M5:=0
M10:=0
MAX:=0

```

```

count:=0
ввод число
нц пока число>0
  если mod(число,2)=0 и число>M2
    то M2:=число
    иначе
      если mod(число,5)=0 и число>M5
        то M5:=число все
      все
    если mod(число,10)=0 и число>M10
      то
        если M10>=MAX то MAX:=M10 все
        M10:=число
      иначе
        если число>MAX
          то MAX:=число все
        все
      ввод число
      count:=count+1
    кц
  цел отв
  ввод R
  если M2*M5<M10*MAX то
    отв:= M10*MAX
  иначе
    отв:=M5*M2
  все
  вывод "Получено чисел: ",count, нс
  вывод "Полученное контрольное значение: ", R,нс
  вывод "Вычисленное контрольное значение: ",
отв,нс
  если отв=R то
    вывод "Контроль пройден."
  иначе
    вывод "Контроль не пройден."
  все
кон

```

Пример правильной и эффективной программы на языке Бейсик:

```

M10 = 0
M2 = 0
M5 = 0
COUNT = 0
R = 0
RES = 0
INPUT DAT
WHILE DAT > 0
  IF DAT MOD 2 = 0 AND DAT > M2 THEN
    M2 = DAT
  ELSE
    IF DAT MOD 5 = 0 AND DAT > M5 THEN
      M5 = DAT
    END IF
  END IF
  IF DAT MOD 10 = 0 AND DAT > M10 THEN
    IF M10 >= MAX THEN
      MAX = M10
    END IF
    M10 = DAT
  ELSE
    IF DAT > MAX THEN
      MAX = DAT
    END IF
  END IF
  COUNT = COUNT + 1
  INPUT DAT
WEND
INPUT R
IF M5 * M2 < M10 * MAX THEN
  RES = M10 * MAX
ELSE
  RES = M5 * M2
END IF
IF RES = R THEN
  PRINT "Контроль пройден."
ELSE
  PRINT "Контроль не пройден."
END IF

```

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).	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 при инициализации цикла анализа массива данных или обработке конца массива. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	2
Программа работает неверно, но в ней вычисляются значения M2, M5 и M10. Допускается наличие от одной до пяти синтаксических ошибок, описанных выше.	1
Прочее	0
<i>Максимальный балл</i>	4