**Система оценивания экзаменационной работы по информатике и ИКТ**

**Ответы к заданиям с кратким ответом**

За правильный ответ на каждое задание части 1 ставится 1 балл. Если указаны два и более ответов (в том числе правильный), неверный ответ или ответ отсутствует - 0 баллов.

|  |  |  |  |
| --- | --- | --- | --- |
|  | № задания | Ответ |  |
|  | А1 | 4 |  |
|  | А2 | 1 |  |
|  | A3 | 2 |  |
|  | А4 | 2 |  |
|  | А5 | 4 |  |
|  | А6 | 2 |  |
|  | А7 | 3 |  |
|  | А8 | 4 |  |
|  | А9 | 1 |  |
|  | А10 | 2 |  |
|  | All | 3 |  |
|  | А12 | 2 |  |
|  | А13 | 3 |  |
|  |  | **Часть 2** |  |
| За правильный ответ на каждое задание части 2 ставится 1 балл , за неверный ответ или его отсутствие - 0 баллов. | | | |
|  | № задания | Ответ |  |
|  | В1 | 2212 |  |
|  | В2 | 34 |  |
|  | ВЗ | 3 |  |
|  | В4 | 243 |  |
|  | В5 | 99 |  |
|  | В6 | 43 |  |
|  | В7 | 4 |  |
|  | В8 | 58 |  |
|  | В9 | 23 |  |
|  | В10 | А33 |  |
|  | В11 | BСDА |  |
|  | В12 | 410 |  |
|  | В13 | 37 |  |
|  | В14 | 2 |  |
|  | В15 | 20 |  |

**Часть 3**

**КРИТЕРИИ ОЦЕНИВАНИЯ ЗАДАНИЙ С РАЗВЁРНУТЫМ ОТВЕТОМ**

**C1**

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число N, не превосходящее 109, и выводится максимальная цифра этого числа. Программист торопился и написал программу неправильно. (Ниже для Вашего удобства программа представлена на четырёх языках программирования.)

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 423.

2. Найдите все ошибки в этой программе (их может быть одна или несколько). Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, – приведите правильный вариант строки.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.



|  |  |  |
| --- | --- | --- |
| **Содержание верного ответа и указания по оцениванию**  (допускаются иные формулировки ответа, не искажающие его смысла) | | |
| Решение использует запись программы на Паскале. Допускается использование программы на трёх других языках.  1. Программа выведет число 9.  2. Первая ошибка. Неверная инициализация ответа (переменная max\_digit).  Строка с ошибкой: max\_digit := 9;  Возможные варианты исправления: max\_digit := 0;  Возможны и другие исправления инициализации, например на отрицательное число, в том числе -maxint.  3. Вторая ошибка. Неверное условие продолжения цикла. Программа не будет рассматривать старшую цифру числа. Строка с ошибкой: while N >= 10 do  Возможные варианты исправления:  while (N >= 1) do или  while (N > 0) do  При этом замены на while (N > 1) do или while (N >= 0) do корректными не являются | | |
| **Указания по оцениванию** | **Баллы** |
| Обратите внимание! В задаче требовалось выполнить **три** действия: указать, что выведет программа при конкретном входном значении, и исправить две ошибки.  Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия (ниже указано, какое действие считается выполненным).  1. Верно указано, что именно выведет программа при указанных в условии входных данных.  2. Указана и верно исправлена ошибка инициализации (не обязательно с упоминанием этого термина).  3. Указано на неверное условие продолжения цикла, и оно исправлено на верное.  Каждый из п. 2 и 3 считается выполненным, если:  а) правильно указана строка с ошибкой;  б) указан такой новый вариант строки, что при исправлении второй ошибки получается правильная программа |  |
| Правильно выполнены все пункты задания. Программа после исправлений для всех натуральных чисел N, не превосходящих 109,верно определяет максимальную цифру.  В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения | 3 |
| 1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в п. 1 задания ответ неверный или отсутствует, или выполнен п. 1, и верно исправлена только одна ошибка). Верное указание на ошибку при её неверном исправлении при этом не засчитывается.  2. Или выполнен п. 1, а вместо указания на ошибки в программе и их исправления приведён новый верный текст решения, возможно, совершенно непохожий на исходный  3. Или правильно выполнены все действия (приведён верный ответ на вопрос 1, и исправлены обе ошибки), но в текст программы внесены и другие изменения, приводящие к её неверной работе | 2 |
| Правильно выполнено только одно действие из трёх, то есть либо только выполнен п. 1, либо он не выполнен или выполнен неверно и верно исправлена только одна ошибка программы путём её явного указания и исправления | 1 |
| Все пункты задания выполнены неверно (ответ на п. 1 не приведён или приведён неверно; ошибки не найдены или найдены, но не исправлены или исправлены неверно) | 0 |
| *Максимальный балл* | 3 |

**C2**

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 до 10000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести максимальное значение среди трёхзначных элементов массива, не делящихся на 9. Если в исходном массиве нет элемента, значение которого является трёхзначным числом и при этом не кратно 9, то выведите сообщение «Не найдено».

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

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).





|  |
| --- |
| **Содержание верного ответа и указания по оцениванию**  (допускаются иные формулировки ответа, не искажающие его смысла) |
| **На языке Паскаль** |
| max := 99;  for i := 1 to N do  if (a[i]>=100) and (a[i]<=998) and (a[i] mod 9<>0) and (a[i]>max) then  max := a[i];  if max > 99 then writeln(max) else writeln(Не найдено); |
| **На Алгоритмическом языке** |
| max := 99  нц для i от 1 до N  если a[i]>=100 и a[i]<=998 и mod(a[i],9)<>0 и a[i]>max  то  max := a[i]  все  кц  если max > 99  то  вывод max  иначе  вывод "Не найдено"  все |
| **На языке Бейсик** |
| MAX = 99  FOR I = 1 TO N  IF A(I)>=100 AND A(I)<=999 AND A(I) MOD 9<>0 AND A(I)>MAX THEN  MAX = A(I)  END IF  NEXT I  IF MAX > 99 THEN  PRINT MAX  ELSE  PRINT "Не найдено"  END IF |
| **На языке СИ** |
| max = 99;  for (i = 0; i<N; i++)  if (a[i]>99 && a[i]<999 && a[i]%9!=0 && a[i]>max)  max = a[i];  if (max>99)  printf("% d", max);  else  printf("Не найдено"); |
| **На естественном языке** |
| Записываем в переменную MAX начальное значение, равное 99. В цикле от 1-ого элемента до 20-ого находим остаток от деления элемента исходного массива на 9. Если значение данного остатка не равно 0 и значение текущего элемента массива больше 99 и меньше 1000, то сравниваем значение текущего элемента массива со значением переменной MAX. Если текущий элемент массива больше MAX, то записываем в MAX значение этого элемента массива.  Переходим к следующему элементу. После завершения цикла проверяем значение переменной MAX. Если оно больше 99, то выводим его, иначе выводим сообщение «Не найдено» |

|  |  |
| --- | --- |
| **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 MAX (например, присваивается начальное значение, большее или равное 100);  2) неточно определяется принадлежность диапазону трёхзначных чисел (например, в сравнении со 100 вместо знака «больше или равно» используется знак «больше»; также обратите внимание, что с учётом ограничения на кратность 9 допустима проверка на принадлежность диапазону от 100 до 998);  3) неверно осуществляется проверка делимости на 9;  4) на делимость на 9 проверяется не значение элемента, а его индекс;  5) в сложном условии вместо логической операции «И» используется логическая операция «ИЛИ»;  6) отсутствует вывод ответа, или при выводе ответа не учитывается случай, когда требуемого числа в массиве может не быть;  7) используется переменная, не объявленная в разделе описания переменных;  8) не указано или неверно указано условие завершения цикла;  9) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно;  10) неверно расставлены операторные скобки | 1 |
| Ошибок, перечисленных в п. 1–10, две или больше, или алгоритм сформулирован неверно. | 0 |
| *Максимальный балл* | 2 |

**C3**

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 27. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 27 или больше камней.

В начальный момент в куче было *S* камней, 1 ≤ *S* ≤ 26.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока - значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S, при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S, и укажите выигрывающий ход для каждого указанного значения S.

б) Укажите такое значение S, при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два таких значения S, при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения *S* опишите выигрышную стратегию Пети.

3. Укажите значение S, при котором:

- у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

- у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах - количество камней в куче.

|  |
| --- |
| **Содержание верного ответа и указания к оцениванию**  (допускаются иные формулировки ответа, не искажающие его смысла) |
| 1. а) Петя может выиграть, если *S* = 14, … 26. Во всех этих случаях достаточно удвоить количество камней. При меньших значениях *S* за один ход нельзя получить кучу, в которой больше 26 камней.  б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче будет *S* = 13 камней. Тогда после первого хода Пети в куче будет 14, 15 или 26 камней. Во всех случаях Ваня удваивает количество камней и выигрывает в один ход.  2. Возможные значения *S*: 11, 12. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 13 камней: в первом случае добавлением двух камней, во втором добавлением одного камня. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (то есть Петя)  следующим ходом выиграет.  3. Возможное значение *S*: 10. После первого хода Пети в куче будет 11, 12 или 20 камней. Если в куче станет 20 камней, Ваня удвоит количество камней и выиграет первым ходом. Ситуация, когда в куче 11 или 12 камней, разобрана выигрывает своим вторым ходом.  В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Положения после очередных ходов | | | |
| И.п. | 1-й ход Пети (разобраны все ходы) | 1-й ход Вани (только ход по  стратегии) | 2-й ход Пети (разобраны все ходы) | 2-й ход Вани (только ход по стратегии) |
| 10 | 10+1=**11** | 11+2=13 | 13+1=**14** | 14\*2=**28** |
| 13+2=**15** | 15\*2=**30** |
| 13\*2=**26** | 26\*2=**52** |
| 10+2=12 | 12+1=**13** | 13+1=**14** | 14\*2=**28** |
|  | 13+2=**15** | 15\*2=**30** |
|  | 13\*2=**26** | 26\*2=**52** |
|  | 10\*2=20 | 20\*2=**40** |  |  |



|  |  |
| --- | --- |
| **Указания по оцениванию** | **Баллы** |
| В задаче от ученика требуется выполнить 3 задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).  Ошибка в решении, не искажающая основного замысла, например, арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.  Первое задание считается выполненным полностью, если выполнены полностью оба пункта а) и б). Пункт а) считается выполненным полностью, если правильно указаны все позиции, в которых Петя выигрывает первым ходом и указано, каким должен быть первый ход. Пункт б) считается выполненным полностью, если правильно указана позиция, в которой Ваня выигрывает первым ходом и описана стратегия Вани, т.е. показано, как Ваня может получить кучу, в которой содержится нужное количество камней при любом ходе Пети.  Первое задание считается выполненным частично, если выполнены все следующие условия: (а) правильно указаны все позиции, в которых Петя выигрывает первым ходом, (б) правильно указана позиция, в которой Ваня выигрывает первым ходом, и явно сказано, что при любом ходе Пети Ваня может получить кучу, которая содержит нужное для выигрыша количество камней. Отличие от полного решения в том, что выигрышные ходы не указаны.  Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегияПети – так, как это написано в примере решения, или другим ых при выбранной стратегии Пети.  Третье задание выполнено, если правильно указана позиция, выигрышная для Вани, и построено дерево всех партий, возможных при Ваниной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Петя, разобраны все возможные ходы, а для позиций, где должен ходить Ваня, - только ход, соответствующий стратегии, которую выбрал Ваня.  Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения или другим способом |  |
| Выполнены второе и третье задания. Первое задание выполнено полностью или частично. Здесь и далее допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу (см. выше) | 3 |
| Не выполнены условия, позволяющие поставить 3 балла, выполнено одно из следующих условии.  1. Задание 3 выполнено полностью.  2. Первое и второе задания выполнены полностью.  3. Первое задание выполнено полностью или частично; для заданий 2 и 3 указаны правильные значения S | 2 |
| Не выполнены условия, позволяющие поставить 3 или 2 балла, выполнено одно из следующих условии.  1. Первое задание выполнено полностью.  2. Во втором задании правильно указано одно из двух возможных значений S и для этого значения указана и обоснована выигрышная стратегия Пети.  3. Первое задание выполнено частично и для одного из остальных заданий правильно указано значение S.  4. Для второго и третьего задания правильно указаны значения S | 1 |
| Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл | 0 |
| *Максимальный балл* | 3 |

**C4**

По каналу связи передаётся последовательность положительных целых чисел, все числа не превышают 1000. Количество чисел известно, но может быть очень велико. Затем передаётся контрольное значение последовательности – наибольшее число *R*, удовлетворяющее следующим условиям:

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

2) *R* делится на 21.

Если такого числа *R* нет, то контрольное значение полагается равным 0.

В результате помех при передаче как сами числа, так и контрольное значение могут быть искажены.

Напишите эффективную, в том числе по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland

Pascal 7.0), которая будет проверять правильность контрольного значения.

Программа должна напечатать отчёт по следующей форме:

*Вычисленное контрольное значение: …*

*Контроль пройден (или – Контроль не пройден)*

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

На вход программе в первой строке подаётся количество чисел *N*. В каждой из последующих *N* строк записано одно натуральное число, не превышающее 1000. В последней строке записано контрольное значение.

*Пример входных данных:*

*6*

*70*

*21*

*997*

*7*

*9*

*300*

*21000*

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

*Вычисленное контрольное значение: 21000*

*Контроль пройден*

|  |
| --- |
| **Содержание верного ответа и указания по оцениванию**  (допускаются иные формулировки ответа, не искажающие его смысла) |
| Произведение двух чисел делится на 21, если:   один из сомножителей делится на 21 (второй может быть любым) либо   ни один из сомножителей не делится на 21, причём один из сомножителей делится на 7, 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 четырёх величин:  М7 – самое большое число, кратное 7, но не кратное 3;  M3 – самое большое число, кратное 3, но не кратное 7;  M21 – самое большое число, кратное 21;  МAX – самое большое число среди всех элементов последовательности, отличное от М21 (если число М21 встретилось более одного раза и оно же является максимальным, то MAX = M21).  После того как все данные прочитаны, искомое контрольное значение вычисляется как максимум из произведений М21\*MAX и М7\*М3.  Ниже приведён пример программы на языке Паскаль, которая реализует описанный алгоритм.  Кроме того, приведён пример программы на языке Бейсик, которая правильно решает задачу, но использует алгоритм, немного отличающийся от описанного выше. Возможны и другие правильные алгоритмы.  Допускаются решения, записанные на других языках программирования |
| **Пример правильной и эффективной программы на языке Бейсик:** |
| var M7,M3,M21,R,MAX,dat,res,i,N: longint;  begin  M7 := 0;  M3 := 0;  M21 := 0;  MAX := 0;  readln(N);  for i := 1 to N do  begin  readln(dat);  if ((dat mod 7) = 0) and ((dat mod 3) > 0) and (dat > M7) then  M7 := dat;  if ((dat mod 3) = 0) and ((dat mod 7) > 0) and (dat > M3) then  M3 := dat;  if (dat mod 21 = 0) and (dat > M21) then  begin  if M21 > MAX then MAX := M21;  M21 := dat  end  else  if dat > MAX then  MAX := dat;  end;  readln(R);  if (M7\*M3 < M21\*MAX) then  res := M21\*MAX  else  res := M7\*M3;  writeln('Вычисленное контрольное значение: ',res);  if R = res then writeln('Контроль пройден')  else writeln('Контроль не пройден');  M21 = 0  M7 = 0  M3 = 0  MAX = 0  INPUT N  FOR I = 1 TO N  INPUT DAT  IF DAT MOD 7 = 0 AND DAT > M7 THEN  M7 = DAT  ELSE  IF DAT MOD 3 = 0 AND DAT > M3 THEN  M3 = DAT  END IF  END IF  IF DAT MOD 21 = 0 AND DAT > M21 THEN  IF M21 > MAX THEN  MAX = M21  END IF  M21 = DAT  ELSE  IF DAT > MAX THEN  MAX = DAT  END IF  END IF  NEXT I  INPUT R  IF M3 \* M7 < M21 \* MAX THEN  RES = M21 \* MAX  ELSE  RES = M3 \* M7  END IF  PRINT "Вычисленное контрольное значение:"; RES  IF RES = R THEN  PRINT "Контроль пройден"  ELSE  PRINT "Контроль не пройден"  END IF  END |

|  |  |
| --- | --- |
| **Указания по оцениванию** | **Баллы** |
| 4 балла ставится за эффективную и правильно работающую программу, которая, возможно, содержит до трёх синтаксических ошибок.  3 балла ставится в случае, когда задача фактически решена, но программа содержит четыре-пять синтаксических ошибок, или если допущена одна содержательная ошибка, или если все входные данные сохраняются в массиве или иной структуре данных (программа неэффективна по памяти, но эффективна по времени работы).  2 балла ставится, если программа неэффективна по времени работы (перебираются все возможные пары элементов), или в программе две содержательные ошибки, либо шесть-семь синтаксических ошибок.  1 балл ставится, если программа написана неверно, но из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.  Далее уточняются перечисленные выше критерии |  |
| Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.  Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:   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 |
| Не выполнены условия, позволяющие поставить 4 балла.  Программа в целом работает правильно для любых входных данных произвольного размера. Время работы пропорционально количеству введённых чисел.  Количество синтаксических ошибок («описок») указанных выше видов – не более пяти.  Используемая память, возможно, зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве или другой структуре данных (контейнер priority\_queue*,* vector, set или map в С++)).  Допускается ошибка при вводе данных, неверный или неполный вывод результатов или неверная работа программы в «экзотических» ситуациях. Например, при использовании 16-битного целого (integer в BPascal или Qbasic) умножаются два числа этого типа (результат по условию может не помещаться  в 16 бит).  Кроме того, допускается наличие одной ошибки, принадлежащей к одному из следующих видов ошибок.  1. Допущена ошибка при инициализации максимумов.  2. Неверно обрабатывается ситуация, когда один или несколько максимумов не определены.  3. Неверно обрабатывается ситуация, когда максимальное произведение получается умножением одинаковых чисел (но разных элементов входной последовательности).  4. При вычислении максимумов учитываются произведения вида a[i]\*a[i].  5. Допущен выход за границу массива.  6. Допущена ошибка в реализации алгоритма сортировки.  7. Используется знак “<” вместо “<=”, “or” вместо “and” и т.п. | 3 |
| Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет, но в реализации алгоритма содержится до двух содержательны хошибок, допустимые виды ошибок перечислены в критериях на 3 балла.  Количество синтаксических «описок» не должно быть более семи.  Программа может быть неэффективна по времени. Например, все числа запоминаются в массиве (ниже n – количество прочитанных чисел), и перебираются все возможные пары элементов последовательности.  Например, так:  max := 0;  for i := 1 to n - 1 do  for j := i + 1 to n do  if ((a[i]\*a[j]) mod 21 = 0) and  ((a[i]\*a[j]) > max) then  max := a[i]\*a[j]; | 2 |
| Не выполнены условия, позволяющие поставить 2, 3 или 4 балла.  При этом выполнено одно из двух условий.  1. Из описания алгоритма и общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи.  2. Программа правильно работает в одном из важных частных случаев, например в случае, когда искомое произведение – этопроизведение наибольшего числа, которое делится на 3, и наибольшего числа, которое делится на 7.  Допускается любое количество синтаксических ошибок | 1 |
| Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла | 0 |
| *Максимальный балл* | 4 |