

Система оценивания экзаменационной работы по информатике и ИКТ**Часть 1**

За правильный ответ на задания 1–23 ставится 1 балл; за неверный ответ или его отсутствие – 0 баллов.

Часть 2**Критерии оценивания заданий с развёрнутым ответом****24**

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число x , не превосходящее 1000, и выводится количество значащих цифр в двоичной записи этого числа. Программист торопился и написал программу неправильно. (Ниже для Вашего удобства программа представлена на пяти языках программирования.)

Бейсик	Python
<pre>INPUT x cnt = 0 WHILE x > 0 cnt = cnt + x MOD 2 x = x \ 10 WEND PRINT cnt END</pre>	<pre>x = int(input()) cnt = 0 while x > 0: cnt = cnt+x % 2 x = x // 10 print(cnt)</pre>
Алгоритмический язык	Паскаль
<pre><u>алг</u> <u>нач</u> <u>цел</u> x, cnt <u>ввод</u> x cnt := 0 <u>нц пока</u> x > 0 cnt := cnt+mod(x,2) x := div(x, 10) <u>кц</u> <u>вывод</u> cnt <u>кон</u></pre>	<pre>var x,cnt: integer; begin readln(x); cnt := 0; while x > 0 do begin cnt:=cnt + x mod 2; x := x div 10 end; writeln(cnt) end.</pre>
Си	
<pre>#include<stdio.h> int main() { int x,cnt; scanf("%ld", &x); cnt = 0; while (x > 0) { cnt = cnt + x%2; x = x /10; } printf("%d", cnt); }</pre>	

Последовательно выполните следующее:

1. Напишите, что выведет эта программа при вводе числа 15.
2. Приведите пример такого числа x , что, несмотря на ошибки, программа печатает правильный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:
 - 1) выпишите строку, в которой сделана ошибка;
 - 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

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 четырех других языках.

1. Программа выведет число 2.
2. Программа напечатает правильный ответ при $x=1$.

Замечание для проверяющего: это единственный случай, когда программа работает верно.

3. **Первая ошибка.** Неверный подсчёт количества цифр. Программа не учитывает значащие нули.

Строка с ошибкой:

```
cnt:=cnt + x mod 2;
```

Возможный вариант исправления:

```
cnt:=cnt + 1;
```

4. **Вторая ошибка.** Неверная обработка числа x в цикле. Строка с ошибкой:

```
x := x div 10;
```

Возможный вариант исправления:

```
x := x div 2;
```

Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить четыре действия:</p> <p>1) указать, что выведет программа при конкретных входных данных (числе x);</p> <p>2) указать пример входных данных, при которых программа работает правильно;</p> <p>3) исправить первую ошибку;</p> <p>4) исправить вторую ошибку.</p> <p>Каждое из действий 3 и 4 считается выполненным, если:</p> <p>а) правильно указана строка с ошибкой;</p> <p>б) указан такой новый вариант строки, что при исправлении второй ошибки получается правильная программа</p>	
Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций:</p> <p>а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной;</p> <p>б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки</p>	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырёх	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

25

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от -100 до 100 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, произведение которых положительно, а сумма кратна 7. Под парой подразумевается два подряд идущих элемента массива.

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.

Бейсик	Паскаль
<pre>N = 40 DIM A(N) AS LONG DIM I, J, K AS LONG FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>const N = 40; var a: array [1..N] of longint; i, j, k: longint; begin for i := 1 to N do readln(a[i]); ... end.</pre>
Си	Алгоритмический язык
<pre>#include <stdio.h> #define N 40 void main() { long a[N]; long i, j, k; for (i = 0; i < N; i++) scanf("%ld", &a[i]); ... }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N = 40 <u>цел таб</u> a[1:N] <u>цел</u> i, j, k <u>нц для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></pre>
Естественный язык	
<p>Объявляем массив А из 40 элементов. Объявляем целочисленные переменные I, J, K. В цикле от 1 до 40 вводим элементы массива А с 1-го по 40-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 2.6) 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
<pre> к := 0; for i := 1 to N - 1 do if ((a[i] + a[i + 1]) mod 7 = 0) and (a[i] * a[i + 1] > 0) then inc(k); writeln(k); </pre>
На алгоритмическом языке
<pre> к := 0 нц для i от 1 до N - 1 если mod(a[i] + a[i + 1], 7) = 0 и a[i] * a[i + 1] > 0 то к := к + 1 все кц вывод к </pre>
На языке Бейсик
<pre> K = 0 FOR I = 1 TO N-1 IF (A(I) + A(I + 1)) MOD 7 = 0 AND A(I) * A(I + 1) > 0 THEN K = K + 1 END IF NEXT I PRINT K </pre>
На языке Си
<pre> к = 0; for (i = 0; i < N - 1; i++) if ((a[i] + a[i + 1]) % 7 == 0 && a[i] * a[i + 1] > 0) k++; printf("%ld", k); </pre>
На языке Python
<pre> к = 0 for i in range(N - 1): if (a[i] + a[i + 1]) % 7 == 0 and a[i] * a[i + 1] > 0: к += 1 print(к) </pre>
На естественном языке
<p>Записываем в переменную К начальное значение, равное 0. В цикле от первого элемента до предпоследнего находим остаток от деления суммы текущего и следующего элементов массива на 7. Если значение данного остатка равно 0 и произведение текущего и следующего элементов массива больше 0, увеличиваем переменную К на единицу.</p> <p>После завершения цикла выводим значение переменной К</p>

Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы</p>	2
<p>Не выполнены условия, позволяющие поставить 2 балла. При этом предложено в целом верное решение, содержащее не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> 1) в цикле происходит выход за границу массива (например, используется цикл от 1 до N); 2) не инициализируется или неверно инициализируется счётчик количества найденных пар; 3) неверно осуществляется проверка суммы пары элементов на кратность 7 (в том числе не учитывается приоритет операций, т.е. используется выражение вида $a[i]+a[i+1] \bmod 7$); 4) на кратность 7 проверяется не сумма значений элементов, а сумма их индексов; 5) неверно осуществляется проверка произведения пары элементов на положительность; 6) в сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции «И» и «ИЛИ», неверно расставлены скобки в логическом выражении); 7) отсутствует вывод ответа; 8) используется переменная, не объявленная в разделе описания переменных; 9) не указано или неверно указано условие завершения цикла; 10) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно; 11) неверно расставлены операторные скобки. <p>Допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы</p>	1
<p>Не выполнены условия, позволяющие поставить 1 или 2 балла. Например, ошибок, перечисленных в п. 1–11, две или больше, или алгоритм сформулирован неверно</p>	0
<i>Максимальный балл</i>	2

26

Два игрока, Паша и Вася, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу **один** или **три** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 18 или 30 камней. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 48. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 48 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 47$.

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.

Задание 1.

- а) Укажите все такие значения числа S , при которых Паша 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 Паша не может выиграть за один ход, но при любом ходе Паши Вася может выиграть своим первым ходом. опишите выигрышную стратегию Васи.

Задание 2.

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

Задание 3.

Укажите значение S , при котором:

- у Васи есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, и
- у Васи нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

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

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

Задание 1.

- а) Паша может выиграть, если $S = 24, \dots, 47$. Во всех этих случаях достаточно удвоить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 47 камней.
- б) Вася может выиграть первым ходом (как бы ни играл Паша), если исходно в куче будет $S = 23$ камня. Тогда после первого хода Паши в куче будет 24, 27 или 46 камней. Во всех случаях Вася удваивает количество камней и выигрывает в один ход.

Задание 2.

Возможные значения S : 20, 22. В этих случаях Паша, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 23 камней: в первом случае добавлением трёх камней, во втором – добавлением одного камня. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Вася), выиграть не может, а его противник (то есть Паша) следующим ходом выиграет.

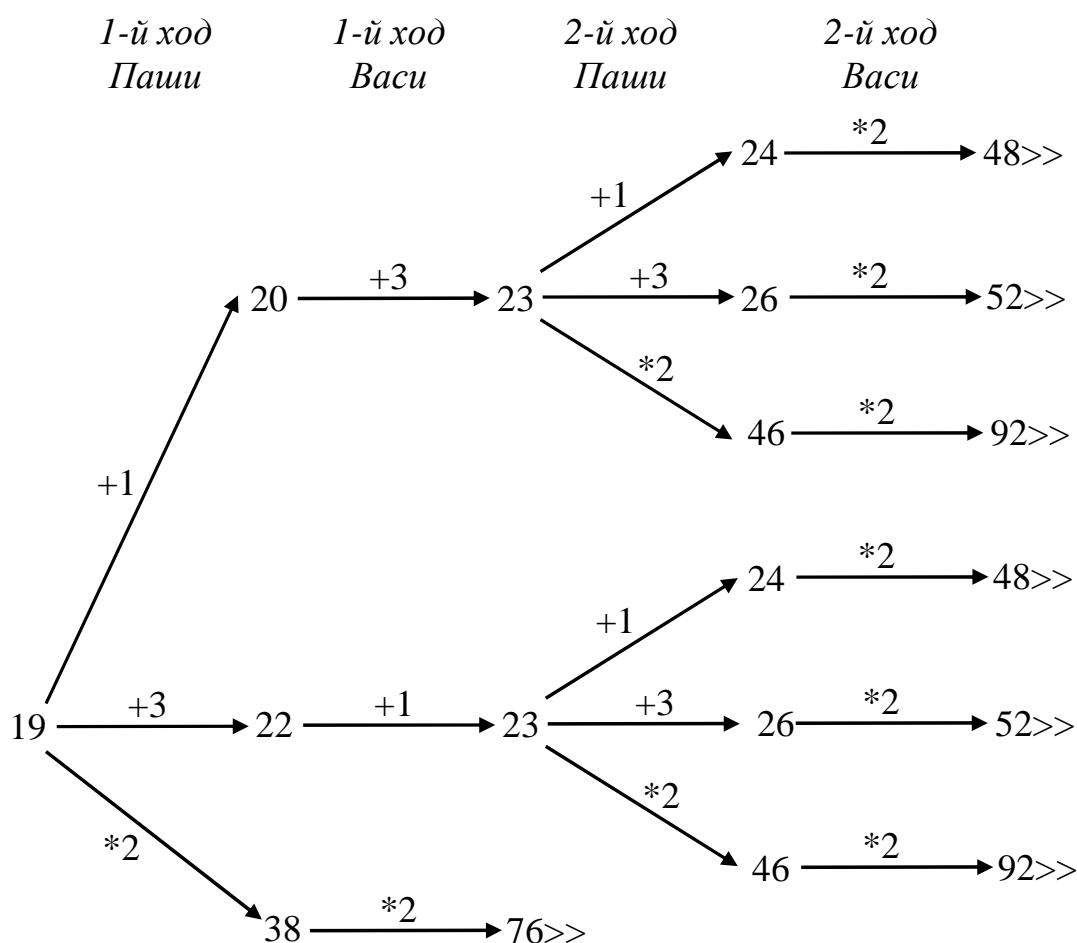
Задание 3.

Возможные значения S : 19, 21. Опишем выигрышную стратегию Васи для $S = 19$. После первого хода Паши в куче будет 20, 22 или 38 камней. Если в куче станет 38 камней, Вася удвоит количество камней и выиграет первым ходом. Ситуация, когда в куче 20 или 22 камня, разобрана в п. 2. В этих ситуациях игрок, который будет ходить (теперь это Вася), выигрывает своим вторым ходом.

В таблице приведены возможные партии при описанной стратегии Васи. Заключительные позиции (в них выигрывает Вася) подчёркнуты. На рисунке дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Исходная позиция	Положения после очередных ходов			
	1-й ход Паши (разобраны все ходы)	1-й ход Васи (только ход по стратегии)	2-й ход Паши (разобраны все ходы)	2-й ход Васи (только ход по стратегии)
19	$19+1=20$	$20+3=23$	$23+1=24$	<u>$24*2=48$</u>
			$23+3=26$	<u>$26*2=52$</u>
			$23*2=46$	<u>$46*2=92$</u>
	$19+3=22$	$22+1=23$	$23+1=24$	<u>$24*2=48$</u>
			$23+3=26$	<u>$26*2=52$</u>
			$23*2=46$	<u>$46*2=92$</u>
	$19*2=38$	<u>$38*2=76$</u>		

Рис. 1. Дерево всех партий, возможных при Васиной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается



Указания по оцениванию

Баллы

В задаче от ученика требуется выполнить три задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).

Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, 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 Паша выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным полностью, если правильно указана позиция, в которой Вася выигрывает первым ходом, и описана стратегия Васи, то есть показано, как Вася может получить кучу, в которой содержится нужное количество камней, при любом ходе Паши.

Первое задание считается выполненным частично, если:
а) правильно указаны все позиции, в которых Паша выигрывает

<p>первым ходом; б) правильно указана позиция, в которой Вася выигрывает первым ходом, и явно сказано, что при любом ходе Паши Вася может получить кучу, которая содержит нужное для выигрыша количество камней. Отличие от выполненного полностью задания состоит в том, что не указаны явно ходы, которыми выиграет Паша или Вася.</p> <p>Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Паши, и описана соответствующая стратегия Паши – так, как это написано в примере решения, или другим способом, например, с помощью дерева всех партий, возможных при выбранной стратегии Паши.</p> <p>Третье задание выполнено, если правильно указана хотя бы одна позиция, выигрышная для Васи, и для этой позиции построено дерево всех партий, возможных при Васиной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Паша, разобраны все возможные ходы, а для позиций, где должен ходить Вася, – только ход, соответствующий стратегии, которую выбрал Вася.</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения или другим способом</p>	
<p>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)</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Третье задание выполнено полностью. 2. Первое и второе задания выполнены полностью. 3. Первое задание выполнено полностью или частично, для второго и третьего заданий указаны правильные значения S 	2
<p>Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Первое или второе задание выполнено полностью. 2. Во втором задании правильно указано одно из двух возможных значений S, и для этого значения указана и обоснована выигрышная стратегия Паши. 3. Первое задание выполнено частично, и для одного из остальных заданий правильно указаны значения S. 4. Для второго и третьего заданий правильно указаны значения S 	1
<p>Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл</p>	0
<p><i>Максимальный балл</i></p>	3

27

Для заданной последовательности неотрицательных целых чисел необходимо найти максимальное произведение двух её элементов, номера которых различаются не менее чем на 8. Значение каждого элемента последовательности не превышает 1000. Количество элементов последовательности не превышает 10000.

Вам предлагаются два задания, связанные с этой задачей: задание А и задание Б. Вы можете решать оба задания А и Б или одно из них по своему выбору.

*Итоговая оценка выставляется как **максимальная** из оценок за задания А и Б. Если решение одного из заданий не представлено, то считается, что оценка за это задание составляет 0 баллов.*

Задание Б является усложненным вариантом задания А, оно содержит дополнительные требования к программе.

А. Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Перед программой укажите версию языка программирования.

Обязательно укажите, что программа является решением **задания А**.

Максимальная оценка за выполнение задания А – 2 балла.

Б. Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству элементов последовательности N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

Обязательно укажите, что программа является решением **задания Б**.

Максимальная оценка за правильную программу, эффективную по времени и по памяти – 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, – 3 балла.

Напоминаем! Не забудьте указать, к какому заданию относится каждая из представленных Вами программ.

Входные данные представлены следующим образом. В первой строке задаётся число N – общее количество элементов последовательности. Гарантируется, что $N > 8$. В каждой из следующих N строк задаётся одно неотрицательное целое число – очередной элемент последовательности.

Пример входных данных:

10
100
45
55
245
35
25
10
10
10
26

Программа должна вывести одно число – описанное в условии произведение.

Пример выходных данных для приведённого выше примера входных данных:

2600

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Задание Б (решение для задания А приведено ниже, см. программу 3). Для каждого элемента с номером k (нумерацию начинаем с 1), начиная с $k = 9$, рассмотрим все допустимые по условиям задачи пары, в которых данный элемент является вторым.

Максимальное произведение из всех этих пар будет получено, если первым в паре будет взят максимальный элемент среди всех, от первого и до элемента с номером $k-8$. Для получения эффективного по времени решения нужно по мере ввода данных помнить максимальное текущее значение, каждое вновь введенное значение умножать на максимум, имевшийся на 8 элементов ранее, и выбрать максимальное из всех таких произведений.

Поскольку каждое текущее максимальное значение используется после ввода ещё 8 элементов и после этого становится ненужным, достаточно хранить только 8 последних максимумов. Для этого можно использовать буферный массив из 8 элементов и циклически заполнять его по мере ввода данных.

Размер этого массива не зависит от общего количества введенных элементов, поэтому такое решение будет эффективным не только по времени, но и по памяти.

Ниже приводится пример такой программы

Программа 1. Пример правильной и эффективной программы на языке Паскаль:

```
program N_27;
const d = 8;
var
  N: integer;
  a: array[0..d-1] of integer; {буфер}
  {k-е введенное число записываем в ячейку a[k mod d]}
  x: integer;
  mx: integer; {максимальное введенное число}
  {(не считая 8 последних)}
  m: integer; { максимальное значение произведения}
  i: integer;
begin
  readln(N);
  {Ввод первых d чисел}
  for i:=1 to d do
  begin
    readln(x);
    a[i mod d] := x
  end;
  { Ввод остальных элементов, поиск максимального
произведения}
  mx := 0; m := 0;
  for i := d + 1 to N do
  begin
    readln(x);
    if a[i mod d] > mx then mx := a[i mod d];
    if x * mx > m then m := x * mx;
    a[i mod d] := x
  end;
  writeln(m)
end.
```

Работа с буферным массивом может быть организована и без использования операции `mod`, например, циклическим перезаписыванием элементов со сдвигом. Этот способ также является эффективным.

Если вместо небольшого массива фиксированного размера (циклического или со сдвигами) хранятся все исходные данные (или все текущие максимумы), программа сохраняет эффективность по времени, но становится неэффективной по памяти, так как требуемая память растёт пропорционально N . Ниже приводится пример такой программы на языке Паскаль. Подобная (и аналогичные по сути) программы оцениваются не выше 3 баллов.

Программа 2. Пример правильной программы на языке Паскаль, эффективной по времени, но неэффективной по памяти:

```
const d = 8;
var
N: integer;
a: array[1..10000] of integer; {хранение всех
элементов последовательности}
mn: integer; {максимальное введенное число}
{не считая d последних}
m: integer; {максимальное значение произведения}
i: integer;
begin
  readln(N); {Ввод всех элементов последовательности}
  for i:=1 to N do readln(a[i]);
  mn := 0;
  m := 0;
  for i := d + 1 to N do
  begin
    if a[i-d] > mn then mn := a[i-d];
    if a[i] * mn > m then m := a[i] * mn
  end;
  writeln(m)
end.
```

Возможно также переборное решение, в котором находятся произведения всех допустимых пар и из них выбирается максимальное. Ниже приведён пример подобного решения. Это (и аналогичные ему) решение не эффективно ни по времени, ни по памяти. Оно является решением задачи А, но не является решением задания Б. Оценка за такое решение – 2 балла.

Программа 3. Пример правильной программы на языке Паскаль, не эффективной ни по времени, ни по памяти:

```
Const d = 8;
var
N: integer;
a: array[1..10000] of integer;
{хранение всех элементов}
m: integer; {максимальное значение произведения}
i, j: integer;
begin
  readln(N);
  {Ввод значений элементов}
  for i:=1 to N do readln(a[i]);
  m := 0;
  for i := 1 to N-d do begin
    for j := i+d to N do begin
      if a[i] * a[j] > m then m := a[i] * a[j];
    end;
  end;
  writeln(m)
end.
```

Указания по оцениванию	Баллы
<p><i>Пояснения для проверяющих.</i></p> <p>1. Задание Б является усложнением задания А. Если в качестве решения задания Б представлено решение задания А, то согласно приведённым ниже критериям, его оценка будет такой же, как если бы это решение было представлено в качестве решения задания А.</p> <p>2. Два задания и, соответственно, возможность для экзаменуемого представить две программы дают ученику возможность (при его желании) сначала написать менее сложное и менее эффективное решение (задание А), которое даёт ему право получить два балла, а затем приступить к поиску более эффективного решения.</p> <p>3. Каждая из программ, представленных экзаменуемым (но не более двух), оценивается по нижеследующим критериям. В качестве итоговой оценки берётся наибольшая из двух оценок.</p>	

<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.</p> <p>Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:</p> <ul style="list-style-type: none"> – пропущен или неверно указан знак пунктуации; – неверно написано или пропущено зарезервированное слово языка программирования; – не описана или неверно описана переменная; – применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку) <p>Эти критерии соответствуют заданию Б.</p>	4
<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа правильно работает для любых соответствующих условию входных данных, время работы пропорционально количеству входных элементов. Размер используемой памяти не имеет значения и может зависеть от объёма входных данных. В частности, допускается использование одного или нескольких массивов размера N (как в первых двух приведённых выше решениях).</p> <p>Программа может содержать не более пяти синтаксических и приравненных к ним ошибок, описанных в критериях на 4 балла.</p> <p>Кроме того, допускается наличие не более одной «содержательной» ошибки из числа следующих:</p> <ul style="list-style-type: none"> – неверная инициализация при поиске максимального значения; – неверная обработка начальных элементов данных, которая может, например, привести к получению ошибочного ответа при $8 < N < 16$; – неточное определение границ массива, выход за границу массива (например, описан массив с границами от 1 до 8, а реально используется от 0 до 7 или наоборот); – вычисленный индекс элемента массива на 1 отличается от верного; – используется знак “<” вместо “<=”, “or” вместо “and” и т.п. <p>Эти критерии соответствуют заданию Б.</p>	3

<p>Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Например, допускается переборное решение, аналогичное вышеприведенной программе 3. Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии на 4 балла). Допускается до двух содержательных ошибок, описанных в критериях на 3 балла Эти критерии соответствуют заданиям А и Б.</p>	2
<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. Из общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. Эти критерии соответствуют заданиям А и Б.</p>	1
<p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<i>Максимальный балл</i>	<i>4</i>

Система оценивания экзаменационной работы по информатике и ИКТ**Часть 2****Критерии оценивания заданий с развёрнутым ответом****24**

Требовалось написать программу, при выполнении которой с клавиатуры считывается натуральное число x , не превосходящее 1000, и выводится количество единиц в двоичной записи этого числа. Программист торопился и написал программу неправильно. (Ниже для Вашего удобства программа представлена на пяти языках программирования.)

Бейсик	Python
<pre>INPUT x cnt = 0 WHILE x > 0 cnt = cnt + 1 x = x MOD 10 WEND PRINT cnt END</pre>	<pre>x = int(input()) cnt = 0 while x > 0: cnt = cnt+1 x = x % 2 print(cnt)</pre>
Алгоритмический язык	Паскаль
<pre><u>алг</u> <u>нач</u> <u>цел</u> x, cnt <u>ввод</u> x cnt := 0 <u>нц пока</u> x > 0 cnt := cnt+1 x := mod(x, 10) <u>кц</u> <u>вывод</u> cnt <u>кон</u></pre>	<pre>var x,cnt: integer; begin readln(x); cnt := 0; while x > 0 do begin cnt:=cnt + 1; x := x mod 10 end; writeln(cnt) end.</pre>
Си	
<pre>#include<stdio.h> int main() { int x,cnt; scanf("%ld", &x); cnt = 0; while (x > 0) { cnt = cnt + 1; x = x % 10; } printf("%d", cnt); }</pre>	

Последовательно выполните следующее:

1. Напишите, что выведет эта программа при вводе числа 6.
2. Приведите пример такого числа x , что, несмотря на ошибки, программа печатает правильный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:
 - 1) выпишите строку, в которой сделана ошибка;
 - 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

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 четырех других языках.

1. Программа выведет число 1.

2. *Замечание для проверяющего:* Программа напечатает правильный ответ при x , равном натуральной степени двойки. Т.е. при $x = 2; 4; 8; 16; 32; 64; 128; 512$. Любое из этих значений является правильным ответом на п.2.

3. **Первая ошибка.** Неверный подсчет количества цифр. Программа считает все значащие цифры, а не только единицы.

Строка с ошибкой:

```
cnt:=cnt + 1;
```

Возможный вариант исправления:

```
cnt:=cnt + x mod 2;
```

4. **Вторая ошибка.** Неверная обработка числа x в цикле. Строка с ошибкой:

```
x := x mod 2;
```

Возможный вариант исправления:

```
x := x div 2;
```

Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить четыре действия:</p> <p>1) указать, что выведет программа при конкретных входных данных (числе x);</p> <p>2) указать пример входных данных, при которых программа работает правильно;</p> <p>3) исправить первую ошибку;</p> <p>4) исправить вторую ошибку.</p> <p>Каждое из действий 3 и 4 считается выполненным, если:</p> <p>а) правильно указана строка с ошибкой;</p> <p>б) указан такой новый вариант строки, что при исправлении второй ошибки получается правильная программа</p>	
Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
<p>Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций:</p> <p>а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной;</p> <p>б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки</p>	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырёх	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

25

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 100 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, сумма которых не кратна 6, а произведение меньше 1000. Под парой подразумевается два подряд идущих элемента массива.

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.

Бейсик	Паскаль
<pre>N = 40 DIM A(N) AS LONG DIM I, J, K AS LONG FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>const N = 40; var a: array [1..N] of longint; i, j, k: longint; begin for i := 1 to N do readln(a[i]); ... end.</pre>
Си	Алгоритмический язык
<pre>#include <stdio.h> #define N 40 void main() { long a[N]; long i, j, k; for (i = 0; i < N; i++) scanf("%ld", &a[i]); ... }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N = 40 <u>цел таб</u> a[1:N] <u>цел</u> i, j, k <u>нц для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u></pre>
Естественный язык	
<p>Объявляем массив А из 40 элементов. Объявляем целочисленные переменные I, J, K. В цикле от 1 до 40 вводим элементы массива А с 1-го по 40-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 2.6) 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
<pre> к := 0; for i := 1 to N - 1 do if ((a[i] + a[i + 1]) mod 6 <> 0) and (a[i] * a[i + 1] < 1000) then inc(k); writeln(k); </pre>
На алгоритмическом языке
<pre> к := 0 нц для i от 1 до N - 1 если mod(a[i] + a[i + 1], 6) <> 0 и a[i] * a[i + 1] < 1000 то к := к + 1 все кц Вывод к </pre>
На языке Бейсик
<pre> K = 0 FOR I = 1 TO N-1 IF (A(I) + A(I + 1)) MOD 6 <> 0 AND A(I) * A(I + 1) < 1000 THEN K = K + 1 END IF NEXT I PRINT K </pre>
На языке Си
<pre> к = 0; for (i = 0; i < N - 1; i++) if ((a[i] + a[i + 1]) % 6 != 0 && a[i] * a[i + 1] < 1000) k++; printf("%ld", k); </pre>
На языке Python
<pre> к = 0 for i in range(N - 1): if (a[i] + a[i + 1]) % 6 != 0 and a[i] * a[i + 1] < 1000: к += 1 print(k) </pre>
На естественном языке
<p>Записываем в переменную К начальное значение, равное 0. В цикле от первого элемента до предпоследнего находим остаток от деления суммы текущего и следующего элементов массива на 6. Если значение данного остатка не равно 0 и произведение текущего и следующего элементов массива меньше 1000, увеличиваем переменную К на единицу. После завершения цикла выводим значение переменной К</p>

Указания по оцениванию	Баллы
<p>Предложен правильный алгоритм, выдающий верное значение. Допускается запись алгоритма на другом языке, использующая аналогичные переменные. В случае, если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования, при этом количество переменных и их идентификаторы должны соответствовать условию задачи. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы</p>	2
<p>Не выполнены условия, позволяющие поставить 2 балла. При этом предложено в целом верное решение, содержащее не более одной ошибки из числа следующих:</p> <ol style="list-style-type: none"> 1) в цикле происходит выход за границу массива (например, используется цикл от 1 до N); 2) не инициализируется или неверно инициализируется счётчик количества найденных пар; 3) неверно осуществляется проверка суммы пары элементов на кратность 6 (в том числе не учитывается приоритет операций, т.е. используется выражение вида $a[i]+a[i+1] \bmod 6$); 4) на кратность 6 проверяется не сумма значений элементов, а сумма их индексов; 5) неверно осуществляется сравнение произведения пары элементов с 1000; 6) в сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции «И» и «ИЛИ», неверно расставлены скобки в логическом выражении); 7) отсутствует вывод ответа; 8) используется переменная, не объявленная в разделе описания переменных; 9) не указано или неверно указано условие завершения цикла; 10) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно; 11) неверно расставлены операторные скобки. <p>Допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы</p>	1
<p>Не выполнены условия, позволяющие поставить 1 или 2 балла. Например, ошибок, перечисленных в п. 1–11, две или больше, или алгоритм сформулирован неверно</p>	0
<i>Максимальный балл</i>	2

26

Два игрока, Паша и Вася, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу **один** или **два** камня или увеличить количество камней в куче в **три** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 45 камней. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 51. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 51 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 50$.

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.

Задание 1.

- а) Укажите все такие значения числа S , при которых Паша 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 Паша не может выиграть за один ход, но при любом ходе Паши Вася может выиграть своим первым ходом. опишите выигрышную стратегию Васи.

Задание 2.

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

Задание 3.

Укажите значение S , при котором:

- у Васи есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Паши, и
- у Васи нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

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

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

Задание 1.

- а) Паша может выиграть, если $S = 17, \dots, 50$. Во всех этих случаях достаточно утроить количество камней. При меньших значениях S за один ход нельзя получить кучу, в которой больше 50 камней.
- б) Вася может выиграть первым ходом (как бы ни играл Паша), если исходно в куче будет $S = 16$ камня. Тогда после первого хода Паши в куче будет 17, 18 или 48 камней. Во всех случаях Вася утраивает количество камней и выигрывает в один ход.

Задание 2.

Возможные значения S : 14, 15. В этих случаях Паша, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 16 камней: в первом случае добавлением одного камня, во втором – добавлением двух камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Вася), выиграть не может, а его противник (то есть Паша) следующим ходом выиграет.

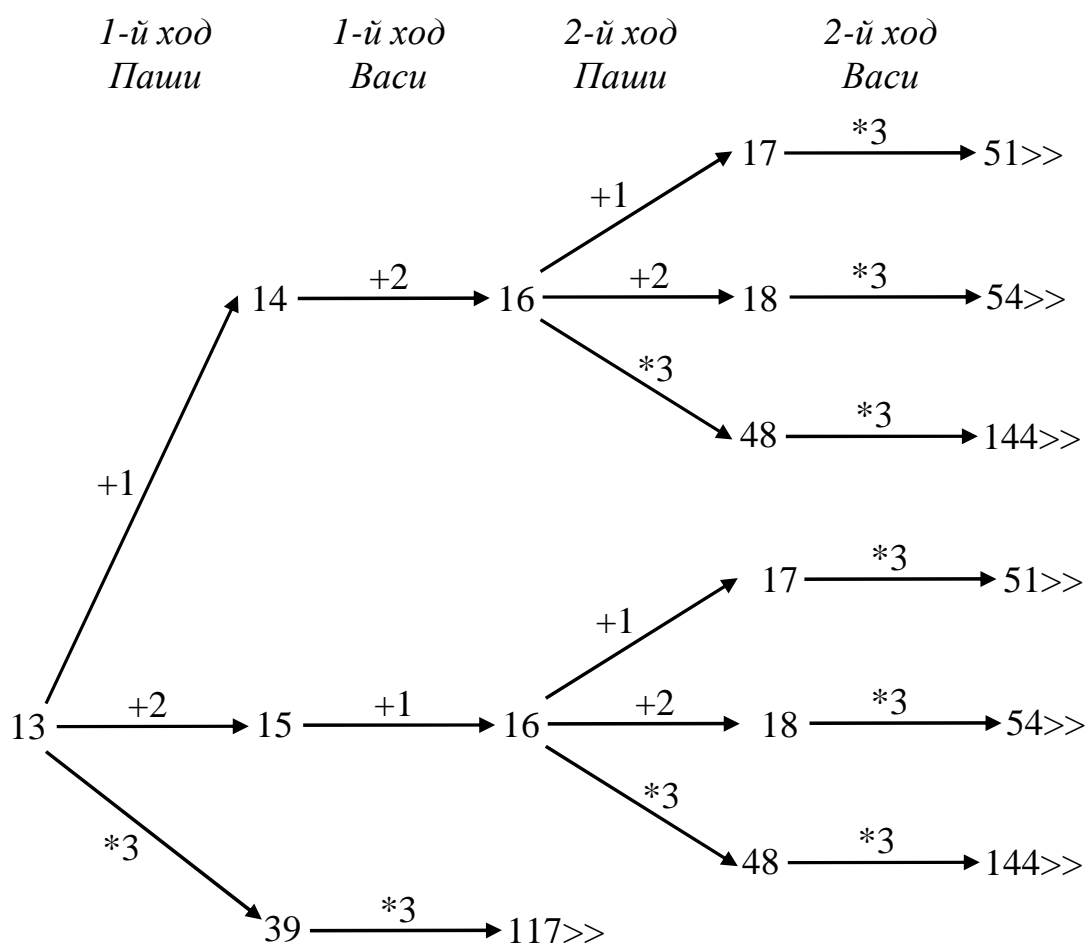
Задание 3.

Возможное значение S : 13. После первого хода Паши в куче будет 14, 15 или 39 камней. Если в куче станет 39 камней, Вася утроит количество камней и выиграет первым ходом. Ситуация, когда в куче 14 или 15 камней, разобрана в п. 2. В этих ситуациях игрок, который будет ходить (теперь это Вася), выигрывает своим вторым ходом.

В таблице приведены возможные партии при описанной стратегии Васи. Заключительные позиции (в них выигрывает Вася) подчёркнуты. На рисунке дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Исходная позиция	Положения после очередных ходов			
	1-й ход Паши (разобраны все ходы)	1-й ход Васи (только ход по стратегии)	2-й ход Паши (разобраны все ходы)	2-й ход Васи (только ход по стратегии)
13	$13+1=14$	$14+2=16$	$16+1=17$	<u>$17*3=51$</u>
			$16+2=18$	<u>$18*3=54$</u>
			$16*3=48$	<u>$48*3=144$</u>
	$13+2=15$	$15+1=16$	$16+1=17$	<u>$17*3=51$</u>
			$16+2=18$	<u>$18*3=54$</u>
			$16*3=48$	<u>$48*3=144$</u>
	$13*3=39$	<u>$39*3=117$</u>		

Рис. 1. Дерево всех партий, возможных при Васиной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается



Указания по оцениванию

Баллы

В задаче от ученика требуется выполнить три задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).

Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, 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 Паша выигрывает первым ходом, и указано, каким должен быть первый ход. Пункт б) считается выполненным полностью, если правильно указана позиция, в которой Вася выигрывает первым ходом, и описана стратегия Васи, то есть показано, как Вася может получить кучу, в которой содержится нужное количество камней, при любом ходе Паши.

Первое задание считается выполненным частично, если:
а) правильно указаны все позиции, в которых Паша выигрывает

<p>первым ходом; б) правильно указана позиция, в которой Вася выигрывает первым ходом, и явно сказано, что при любом ходе Паши Вася может получить кучу, которая содержит нужное для выигрыша количество камней. Отличие от выполненного полностью задания состоит в том, что не указаны явно ходы, которыми выиграет Паша или Вася.</p> <p>Второе задание выполнено, если правильно указаны обе позиции, выигрышные для Паши, и описана соответствующая стратегия Паши – так, как это написано в примере решения, или другим способом, например, с помощью дерева всех партий, возможных при выбранной стратегии Паши.</p> <p>Третье задание выполнено, если правильно указана позиция, выигрышная для Васи, и построено дерево всех партий, возможных при Васиной стратегии. Должно быть явно сказано, что в этом дереве в каждой позиции, где должен ходить Паша, разобраны все возможные ходы, а для позиций, где должен ходить Вася, – только ход, соответствующий стратегии, которую выбрал Вася.</p> <p>Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения или другим способом</p>	
<p>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)</p>	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Третье задание выполнено полностью. 2. Первое и второе задания выполнены полностью. 3. Первое задание выполнено полностью или частично, для второго и третьего заданий указаны правильные значения S 	2
<p>Не выполнены условия, позволяющие поставить 3 или 2 балла, и выполнено одно из следующих условий.</p> <ol style="list-style-type: none"> 1. Первое или второе задание выполнено полностью. 2. Во втором задании правильно указано одно из двух возможных значений S, и для этого значения указана и обоснована выигрышная стратегия Паши. 3. Первое задание выполнено частично, и для одного из остальных заданий правильно указаны значения S. 4. Для второго и третьего заданий правильно указаны значения S 	1
<p>Не выполнено ни одно из условий, позволяющих поставить 3, 2 или 1 балл</p>	0
<p><i>Максимальный балл</i></p>	3

27

Для заданной последовательности неотрицательных целых чисел необходимо найти минимальную сумму двух её элементов, номера которых различаются не менее чем на 4. Значение каждого элемента последовательности не превышает 1000. Количество элементов последовательности не превышает 10000.

Вам предлагаются два задания, связанные с этой задачей: задание А и задание Б. Вы можете решать оба задания А и Б или одно из них по своему выбору.

*Итоговая оценка выставляется как **максимальная** из оценок за задания А и Б. Если решение одного из заданий не представлено, то считается, что оценка за это задание составляет 0 баллов.*

Задание Б является усложненным вариантом задания А, оно содержит дополнительные требования к программе.

А. Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Перед программой укажите версию языка программирования.

Обязательно укажите, что программа является решением задания А.

Максимальная оценка за выполнение задания А – 2 балла.

Б. Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству элементов последовательности N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

Обязательно укажите, что программа является решением задания Б.

Максимальная оценка за правильную программу, эффективную по времени и по памяти – 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, – 3 балла.

Напоминаем! Не забудьте указать, к какому заданию относится каждая из представленных Вами программ.

Входные данные представлены следующим образом. В первой строке задаётся число N – общее количество элементов последовательности. Гарантируется, что $N > 4$. В каждой из следующих N строк задаётся одно неотрицательное целое число – очередной элемент последовательности.

Пример входных данных:

7
10
45
55
245
35
25
10

Программа должна вывести одно число – описанную в условии сумму.

Пример выходных данных для приведённого выше примера входных данных: 20

Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Задание Б (решение для задания А приведено ниже, см. программу З). Для каждого элемента с номером k (нумерацию начинаем с 1), начиная с $k = 5$, рассмотрим все допустимые по условиям задачи пары, в которых данный элемент является вторым.

Минимальная сумма из всех этих пар будет получена, если первым в паре будет взят минимальный элемент среди всех, от первого и до элемента с номером $k-4$. Для получения эффективного по времени решения нужно по мере ввода данных помнить минимальное текущее значение, каждое вновь введенное значение складывать с минимумом, имевшимся на 4 элемента ранее, и выбрать минимальную из всех таких сумм.

Поскольку каждое текущее минимальное значение используется после ввода еще 4 элементов и после этого становится ненужным, достаточно хранить только 4 последних минимума. Для этого можно использовать буферный массив из 4 элементов и циклически заполнять его по мере ввода данных.

Размер этого массива не зависит от общего количества введенных элементов, поэтому такое решение будет эффективным не только по времени, но и по памяти.

Ниже приводится пример такой программы

Программа 1. Пример правильной и эффективной программы на языке Паскаль:

```
program N_27;
const d = 4;
var
  N: integer;
  a: array[0..d-1] of integer; {буфер}
  {k-е введенное число записываем в ячейку a[k mod d]}
  x: integer;
  mx: integer; {минимальное введенное число}
  {(не считая 4 последних)}
  m: integer; { минимальное значение суммы}
  i: integer;
begin
  readln(N);
  {Ввод первых d чисел}
  for i:=1 to d do
  begin
    readln(x);
    a[i mod d] := x
  end;
  { Ввод остальных элементов, поиск минимальной суммы}
  mx := 1001; m := 2001;
  for i := d + 1 to N do
  begin
    readln(x);
    if a[i mod d] < mx then mx := a[i mod d];
    if x + mx < m then m := x + mx;
    a[i mod d] := x
  end;
  writeln(m)
end.
```

Работа с буферным массивом может быть организована и без использования операции `mod`, например, циклическим перезаписыванием элементов со сдвигом. Этот способ также является эффективным.

Если вместо небольшого массива фиксированного размера (циклического или со сдвигами) хранятся все исходные данные (или все текущие минимумы), программа сохраняет эффективность по времени, но становится неэффективной по памяти, так как требуемая память растет пропорционально N . Ниже приводится пример такой программы на языке Паскаль. Подобная (и аналогичные по сути) программы оцениваются не выше 3 баллов.

Программа 2. Пример правильной программы на языке Паскаль, эффективной по времени, но неэффективной по памяти:

```
const d = 4;
var
N: integer;
a: array[1..10000] of integer; {хранение всех
элементов последовательности}
mn: integer; {минимальное введенное число}
{не считая d последних}
m: integer; {минимальное значение суммы}
i: integer;
begin
  readln(N); {Ввод всех элементов последовательности}
  for i:=1 to N do readln(a[i]);
  mn := 1001;
  m := 2001;
  for i := d + 1 to N do
  begin
    if a[i-d] < mn then mn := a[i-d];
    if a[i] + mn < m then m := a[i] + mn
  end;
  writeln(m)
end.
```

Возможно также переборное решение, в котором находятся суммы всех допустимых пар и из них выбирается минимальная. Ниже приведен пример подобного решения. Это (и аналогичные ему) решение не эффективно ни по времени, ни по памяти. Оно является решением задачи А, но не является решением задания Б. Оценка за такое решение – 2 балла.

Программа 3. Пример правильной программы на языке Паскаль, не эффективной ни по времени, ни по памяти:

```
Const d = 4;
var
N: integer;
a: array[1..10000] of integer;
{хранение всех элементов}
m: integer; {минимальное значение суммы}
i, j: integer;
begin
  readln(N);
  {Ввод значений элементов}
  for i:=1 to N do readln(a[i]);
  m := 2001;
  for i := 1 to N-d do begin
    for j := i+d to N do begin
      if a[i] + a[j] < m then m := a[i] + a[j];
    end;
  end;
  writeln(m)
end.
```

Указания по оцениванию

Баллы

Пояснения для проверяющих.

1. Задание Б является усложнением задания А. Если в качестве решения задания Б представлено решение задания А, то согласно приведенным ниже критериям, его оценка будет такой же, как если бы это решение было представлено в качестве решения задания А.

2. Два задания и, соответственно, возможность для экзаменуемого представить две программы дают ученику возможность (при его желании) сначала написать менее сложное и менее эффективное решение (задание А), которое дает ему право получить два балла, а затем приступить к поиску более эффективного решения.

3. Каждая из программ, представленных экзаменуемым (но не более двух), оценивается по нижеследующим критериям. В качестве итоговой оценки берется наибольшая из двух оценок.

<p>Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел, а время работы пропорционально этому количеству.</p> <p>Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов:</p> <ul style="list-style-type: none"> – пропущен или неверно указан знак пунктуации; – неверно написано или пропущено зарезервированное слово языка программирования; – не описана или неверно описана переменная; – применяется операция, недопустимая для соответствующего типа данных (если одна и та же ошибка встречается несколько раз, то это считается за одну ошибку) <p>Эти критерии соответствуют заданию Б.</p>	4
<p>Не выполнены условия, позволяющие поставить 4 балла.</p> <p>Программа правильно работает для любых соответствующих условию входных данных, время работы пропорционально количеству входных элементов. Размер используемой памяти не имеет значения и может зависеть от объёма входных данных. В частности, допускается использование одного или нескольких массивов размера N (как в первых двух приведённых выше решениях).</p> <p>Программа может содержать не более пяти синтаксических и приравненных к ним ошибок, описанных в критериях на 4 балла.</p> <p>Кроме того, допускается наличие не более одной «содержательной» ошибки из числа следующих:</p> <ul style="list-style-type: none"> – неверная инициализация при поиске минимального значения; – неверная обработка начальных элементов данных, которая может, например, привести к получению ошибочного ответа при $4 < N < 8$; – неточное определение границ массива, выход за границу массива (например, описан массив с границами от 1 до 4, а реально используется от 0 до 3 или наоборот); – вычисленный индекс элемента массива на 1 отличается от верного; – используется знак “<” вместо “<=”, “or” вместо “and” и т.п. <p>Эти критерии соответствуют заданию Б.</p>	3

Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Например, допускается переборное решение, аналогичное вышеприведенной программе 3. Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии на 4 балла). Допускается до двух содержательных ошибок, описанных в критериях на 3 балла Эти критерии соответствуют заданиям А и Б.	2
Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. Из общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. Эти критерии соответствуют заданиям А и Б.	1
Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла	0
<i>Максимальный балл</i>	<i>4</i>