

Ассоциация научно-технических организаций "Уральский профессиональный форум"  
Автономная некоммерческая профессиональная образовательная организация  
"Современный цифровой колледж при Западно-уральском институте экономики и права"  
(АНПОО "СЦК при ЗУИЭП")

РАССМОТРЕНО  
на заседании Педагогического  
совета  
протокол от «09» февраля 2023 №  
8



**ОЦЕНОЧНЫЕ СРЕДСТВА**  
**для проведения промежуточной аттестации**  
**в форме экзамена**  
*по учебному предмету*

**ИНФОРМАТИКА**

Общеобразовательного цикла

по специальности

09.02.07 Информационные системы и программирование  
квалификация «Программист»

форма обучения: очно-заочная

Вводится с 01.09.2023

Пермь 2023

## Содержание:

1. Паспорт комплекта оценочных средств..... 3
2. Лист согласования дополнений и изменений к комплекту КИМ ..... 6
3. Комплект контрольно-измерительных материалов для оценки освоения учебного предмета. .... 8

# 1. ПАСПОРТ КОМПЛЕКТА ОЦЕНОЧНЫХ СРЕДСТВ

Предметом оценки на экзамене являются следующие результаты:

П1. владение представлениями о роли информации и связанных с ней процессов в природе, технике и обществе, понятиями "информация", "информационный процесс", "система", "компоненты системы", "системный эффект", "информационная система", "система управления";

П2. владение методами поиска информации в сети Интернет, умение критически оценивать информацию, полученную из сети Интернет;

П3. умение характеризовать большие данные, приводить примеры источников их получения и направления использования;

П4. понимание основных принципов устройства и функционирования современных стационарных и мобильных компьютеров, тенденций развития компьютерных технологий;

П5. владение навыками работы с операционными системами, основными видами программного обеспечения для решения учебных задач по выбранной специализации;

П6. соблюдение требований техники безопасности и гигиены при работе с компьютерами и другими компонентами цифрового окружения, понимание правовых основ использования компьютерных программ, баз данных и материалов, размещенных в сети Интернет;

П7. понимание основных принципов дискретизации различных видов информации, умение определять информационный объем текстовых, графических и звуковых данных при заданных параметрах дискретизации;

П8. умение строить неравномерные коды, допускающие однозначное декодирование сообщений (префиксные коды);

П9. владение теоретическим аппаратом, позволяющим осуществлять представление заданного натурального числа в различных системах счисления, выполнять преобразования логических выражений, используя законы алгебры логики;

П10. умение создавать структурированные текстовые документы и демонстрационные материалы с использованием возможностей современных программных средств и облачных сервисов;

П11. наличие представлений о компьютерных сетях и их роли в современном мире, об общих принципах разработки и функционирования интернет-приложений;

П12. понимание угроз информационной безопасности, использование методов и средств противодействия этим угрозам, соблюдение мер безопасности, предотвращающих незаконное распространение персональных данных;

П13. владение теоретическим аппаратом, позволяющим определять кратчайший путь во взвешенном графе и количество путей между вершинами ориентированного ациклического графа;

П14. умение читать и понимать программы, реализующие несложные алгоритмы обработки числовых и текстовых данных (в том числе массивов и символьных строк) на выбранном для изучения универсальном языке программирования высокого уровня (Паскаль, Python, Java, C++, C#), анализировать

алгоритмы с использованием таблиц трассировки, определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных, модифицировать готовые программы для решения новых задач, использовать их в своих программах в качестве подпрограмм (процедур, функций);

П15. умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей, нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10, вычисление обобщенных характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию), сортировку элементов массива;

П16. умение использовать табличные (реляционные) базы данных, в частности, составлять запросы к базам данных (в том числе запросы с вычисляемыми полями), выполнять сортировку и поиск записей в базе данных, наполнять разработанную базу данных, умение использовать электронные таблицы для анализа, представления и обработки данных (включая вычисление суммы, среднего арифметического, наибольшего и наименьшего значений, решение уравнений);

П17. умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования, оценивать соответствие модели моделируемому объекту или процессу, представлять результаты моделирования в наглядном виде;

П18. умение организовывать личное информационное пространство с использованием различных цифровых технологий, понимание возможностей цифровых сервисов государственных услуг, цифровых образовательных сервисов, понимание возможностей и ограничений технологий искусственного интеллекта в различных областях, наличие представлений об использовании информационных технологий в различных профессиональных сферах.

**Вид контроля:** промежуточная аттестация – экзамен.

Контроль и оценка на экзамене осуществляются с использованием стандартизированных контрольно-измерительных материалов (КИМ), представленных в форме тестовых заданий, соответствующих материалам ЕГЭ текущего учебного года.

По итогам тестирования выставляется оценка по шкале: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» в соответствии со шкалой перевода:

Процент результативности (правильных ответов)	Количество правильных ответов на вопросы теста	Оценка уровня подготовки	
		балл (отметка)	вербальный аналог
65 - 100	65 - 100	5	отлично

56 - 65	56 - 65	4	хорошо
30 - 55	40 - 55	3	удовлетворительно
Менее 30	Менее 30	2	неудовлетворительно

Оценочные средства предназначены для контроля и оценки результатов обучения студентов, освоивших программу учебного предмета «Информатика» и разработаны на основании рабочей программы предмета.



## Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ

### Инструкция по выполнению работы

Экзаменационная работа состоит из двух частей, включающих в себя 27 заданий. Часть 1 содержит 23 задания с кратким ответом. Часть 2 содержит 4 задания с развёрнутым ответом.

На выполнение экзаменационной работы по информатике и ИКТ отводится 3 часа 55 минут (235 минут).

Ответы к заданиям 1–23 записываются в виде числа, последовательности букв или цифр. Ответ запишите в поле ответа в тексте работы, а затем перенесите в бланк ответов № 1.

КИМ

Ответ: 23.

2	3											
---	---	--	--	--	--	--	--	--	--	--	--	--

Бланк

Задания 24–27 требуют развёрнутого решения. В бланке ответов № 2 укажите номер задания и запишите его полное решение.

Все бланки ЕГЭ заполняются яркими чёрными чернилами. Допускается использование гелевой или капиллярной ручки.

При выполнении заданий можно пользоваться черновиком. **Записи в черновике, а также в тексте контрольных измерительных материалов не учитываются при оценивании работы.**

Баллы, полученные Вами за выполненные задания, суммируются. Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов.

После завершения работы проверьте, чтобы ответ на каждое задание в бланках ответов № 1 и № 2 был записан под правильным номером.

*Желаем успеха!*

В экзаменационных заданиях используются следующие соглашения.

- Обозначения для логических связок (операций):
  - отрицание* (инверсия, логическое НЕ) обозначается  $\neg$  (например,  $\neg A$ );
  - конъюнкция* (логическое умножение, логическое И) обозначается  $\wedge$  (например,  $A \wedge B$ ) либо  $\&$  (например,  $A \& B$ );
  - дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается  $\vee$  (например,  $A \vee B$ ) либо  $|$  (например,  $A | B$ );
  - следование* (импликация) обозначается  $\rightarrow$  (например,  $A \rightarrow B$ );
  - тождество* обозначается  $\equiv$  (например,  $A \equiv B$ ). Выражение  $A \equiv B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают (либо они оба истинны, либо они оба ложны);
  - символ 1 используется для обозначения истины (истинного высказывания); символ 0 – для обозначения лжи (ложного высказывания).

2. Два логических выражения, содержащих переменные, называются *равносильными* (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения  $A \rightarrow B$  и  $(\neg A) \vee B$  равносильны, а  $A \vee B$  и  $A \wedge B$  неравносильны (значения выражений разные, например, при  $A = 1, B = 0$ ).

3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), тождество. Таким образом,  $\neg A \wedge B \vee C \wedge D$  означает то же, что и  $((\neg A) \wedge B) \vee (C \wedge D)$ .

Возможна запись  $A \wedge B \wedge C$  вместо  $(A \wedge B) \wedge C$ . То же относится и к дизъюнкции: возможна запись  $A \vee B \vee C$  вместо  $(A \vee B) \vee C$ .

4. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле – как обозначения единиц измерения, чьё соотношение с единицей «байт» выражается степенью двойки.

**Часть 1**

Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишете в отдельной клеточке в соответствии с приведёнными в бланке образцами.

**1** Сколько единиц в двоичной записи шестнадцатеричного числа E1F0<sub>16</sub>?

Ответ: \_\_\_\_\_.

**2** Миша заполнял таблицу истинности функции  $(x \wedge \neg y) \vee (x \equiv z) \vee \neg w$ , но успел заполнить лишь фрагмент из трёх **различных** её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ .

				$(x \wedge \neg y) \vee (x \equiv z) \vee \neg w$
0	1	1	0	<b>0</b>
0				<b>0</b>
	1	0	1	<b>0</b>

Определите, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ .

В ответе напишите буквы  $w, x, y, z$  в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишете подряд, никаких разделителей между буквами ставить не нужно.

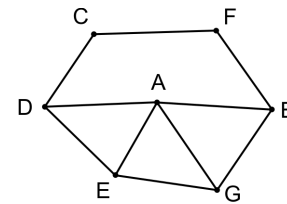
Пример. Функция задана выражением  $\neg x \vee y$ , зависящим от двух переменных, а фрагмент таблицы имеет следующий вид.

		$\neg x \vee y$
0	1	<b>0</b>

В этом случае первому столбцу соответствует переменная  $y$ , а второму столбцу – переменная  $x$ . В ответе следует написать  $yx$ .

Ответ: \_\_\_\_\_.

**3** На рисунке слева изображена схема дорог N-ского района. В таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет.



		Номер пункта						
		1	2	3	4	5	6	7
Номер пункта	1			*			*	
	2				*	*	*	*
	3	*						*
	4		*			*		*
	5		*		*		*	
	6	*	*			*		
	7		*	*	*			

Каждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам E и G на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

Ответ: \_\_\_\_\_.



- 4 Ниже представлены два фрагмента таблиц из базы данных о жителях микрорайона. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных, у скольких детей на момент их рождения матерям было меньше 27 полных лет. При вычислении ответа учитывайте только информацию из приведённых фрагментов таблиц.

Таблица 1				Таблица 2	
ID	Фамилия И.О.	Пол	Год рождения	ID_Родителя	ID_Ребёнка
14	Краснова Н.А.	Ж	1933	24	25
24	Кузьминых И.П.	М	1934	44	25
25	Кузьминых П.И.	М	1964	25	26
26	Кузьминых П.П.	М	1985	64	26
34	Красняк А.И.	Ж	1955	24	34
35	Красняк В.С.	Ж	1978	44	34
36	Красняк С.С.	М	1955	34	35
44	Воевода А.С.	Ж	1932	36	35
45	Воевода В.А.	М	1944	14	36
46	Макаренко О.С.	Ж	1980	34	46
47	Макаренко П.О.	М	2000	36	46
54	Клычко А.П.	Ж	1984	25	54
64	Крот П.А.	Ж	1955	64	54
...	...	...	...	...	...

Ответ: \_\_\_\_\_.

- 5 По каналу связи передаются сообщения, содержащие только четыре буквы: А, Б, В, Г; для передачи используется двоичный код, удовлетворяющий условию Фано. Для букв Б, В, Г используются такие кодовые слова: Б – 101; В – 110; Г – 0.
- Укажите кратчайшее кодовое слово для буквы А, при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите код с **наибольшим** числовым значением.
- Примечание.* Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Ответ: \_\_\_\_\_.

- 6 На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число следующим образом.

1) Строится двоичная запись числа  $N$ .

2) К этой записи дописываются справа ещё два разряда по следующему правилу: если  $N$  чётное, в конец числа (справа) дописываются два нуля, в противном случае справа дописываются две единицы. Например, двоичная запись 1001 числа 9 будет преобразована в 100111.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью числа – результата работы данного алгоритма.

Укажите минимальное число  $N$ , для которого результат работы алгоритма будет больше 134. В ответе это число запишите в десятичной системе счисления.

Ответ: \_\_\_\_\_.

- 7 Дан фрагмент электронной таблицы. Из ячейки А3 в ячейку С4 была скопирована формула. При копировании адреса ячеек в формуле автоматически изменились. Какова сумма числовых значений формул в ячейках А3 и С4?

	А	В	С	Д	Е
1	1	2	3	4	5
2	6	7	8	9	10
3	=С1+А\$1	12	13	14	15
4	16	17		19	20

*Примечание.* Знак \$ обозначает абсолютную адресацию.

Ответ: \_\_\_\_\_.

- 8 Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на пяти языках программирования.

Бейсик	Python
<pre>DIM S, N AS INTEGER S = 175 N = 0 WHILE S + N &lt; 325   S = S - 10   N = N + 30 WEND PRINT S</pre>	<pre>s = 175 n = 0 while s + n &lt; 325:   s = s - 10   n = n + 30 print(s)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач   цел n, s   s := 175   n := 0   нц пока s + n &lt; 325     s := s - 10     n := n + 30   кц   вывод s кон</pre>	<pre>var s, n: integer; begin   s := 175;   n := 0;   while s + n &lt; 325 do     begin       s := s - 10;       n := n + 30     end;   writeln(s) end.</pre>
C++	
<pre>#include &lt;iostream&gt; using namespace std;  int main() {   int s = 175, n = 0;   while (s + n &lt; 325) {     s = s - 10;     n = n + 30;   }   cout &lt;&lt; s &lt;&lt; endl;   return 0; }</pre>	

Ответ: \_\_\_\_\_.

- 9 Музыкальный фрагмент был записан в формате квадрo (четырёхканальная запись), оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла без учёта размера заголовка файла – 12 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате моно и оцифрован с разрешением в 2 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер в Мбайт файла, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно. Искомый объём не учитывает размера заголовка файла.

Ответ: \_\_\_\_\_.

- 10 Вася составляет 5-буквенные слова, в которых есть только буквы В, О, Л, К, причём буква В используется в каждом слове ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

Ответ: \_\_\_\_\_.

**11** Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик	Python
<pre>SUB F(n)   IF n &gt; 0 THEN     PRINT n,     F(n - 3)     F(n \ 2)   END IF END SUB</pre>	<pre>def F(n):   if n &gt; 0:     print(n)     F(n - 3)     F(n // 2)</pre>
Алгоритмический язык	Паскаль
<pre>алг F(цел n) нач   если n &gt; 0 то     вывод n     F(n - 3)     F(div(n, 2))   все кон</pre>	<pre>procedure F(n: integer); begin   if n &gt; 0 then     begin       write(n);       F(n - 3);       F(n div 2)     end end;</pre>
C++	
<pre>void F(int n){   if (n &gt; 0){     std::cout &lt;&lt; n;     F(n - 3);     F(n / 2);   } }</pre>	

Запишите подряд без пробелов и разделителей все числа, которые будут выведены на экран при выполнении вызова F(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

Ответ: \_\_\_\_\_.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, – в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда – нули. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 117.191.176.37 адрес сети равен 117.191.160.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Ответ: \_\_\_\_\_.

**13** При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 25 символов и содержащий только символы из 7-символьного набора: С, Д, А, М, Е, Г, Э. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 50 пользователях потребовалось 1200 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число – количество байт.

Ответ: \_\_\_\_\_.

**14** Исполнитель Редактор получает на вход строку символов и преобразовывает её.

Редактор может выполнять две команды, в обеих командах  $v$  и  $w$  обозначают цепочки символов.

**А) заменить** ( $v, w$ ).

Эта команда заменяет в строке первое слева вхождение цепочки  $v$  на цепочку  $w$ .

Например, выполнение команды

**заменить** (111, 27)

преобразует строку 05111150 в строку 0527150.

Если в строке нет вхождений цепочки  $v$ , то выполнение команды **заменить** ( $v, w$ ) не меняет эту строку.

**Б) нашлось** ( $v$ ).

Эта команда проверяет, встречается ли цепочка  $v$  в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА *условие*

*последовательность команд*

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*

*ТО команда1*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно).

В конструкции

ЕСЛИ *условие*

*ТО команда1*

*ИНАЧЕ команда2*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно) или *команда2* (если условие ложно).

На вход приведённой ниже программе поступает строка, начинающаяся с символа «>», а затем содержащая 10 цифр 1, 20 цифр 2 и 30 цифр 3, расположенных в произвольном порядке.

Определите сумму числовых значений цифр строки, получившейся в результате выполнения программы.

Так, например, если результат работы программы представлял бы собой строку, состоящую из 50 цифр 4, то верным ответом было бы число 200.

НАЧАЛО

ПОКА **нашлось** (>1) ИЛИ **нашлось** (>2) ИЛИ **нашлось** (>3)

ЕСЛИ **нашлось** (>1)

**ТО заменить** (>1, 22>)

КОНЕЦ ЕСЛИ

ЕСЛИ **нашлось** (>2)

**ТО заменить** (>2, 2>)

КОНЕЦ ЕСЛИ

ЕСЛИ **нашлось** (>3)

**ТО заменить** (>3, 1>)

КОНЕЦ ЕСЛИ

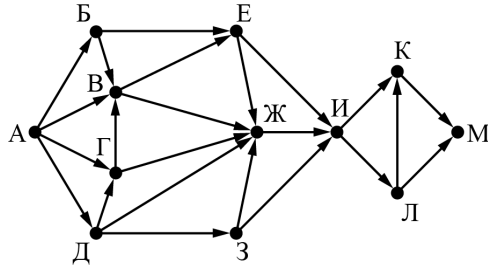
КОНЕЦ ПОКА

КОНЕЦ

Ответ: \_\_\_\_\_.

- 15 На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Какова длина самого длинного пути из города А в город М? Длиной пути считать количество дорог, составляющих этот путь.



Ответ: \_\_\_\_\_.

- 16 Значение арифметического выражения:  $9^{11} \times 3^{20} - 3^9 - 27$  – записали в системе счисления с основанием 3. Сколько цифр 2 содержится в этой записи?

Ответ: \_\_\_\_\_.

- 17 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в сотнях тысяч)
<i>Физика</i>	46
<i>Квант</i>	34
<i>Ньютон</i>	34
<i>Ньютон   Физика   Квант</i>	90
<i>Ньютон &amp; Физика</i>	12
<i>Ньютон &amp; Квант</i>	0

Какое количество страниц (в сотнях тысяч) будет найдено по запросу *Физика & Квант*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: \_\_\_\_\_.

- 18 Для какого наименьшего целого неотрицательного числа  $A$  выражение

$$(x + 2y < A) \vee (y > x) \vee (x > 20)$$

тождественно истинно, т.е. принимает значение 1 при любых целых неотрицательных  $x$  и  $y$ ?

Ответ: \_\_\_\_\_.

- 19 В программе используется одномерный целочисленный массив  $A$  с индексами от 0 до 11. Значения элементов массива  $A[i]$  приведены в таблице.

$i$	0	1	2	3	4	5	6	7	8	9	10	11
$A[i]$	14	13	15	8	4	12	30	21	22	16	5	9

Определите значение переменной  $s$  после выполнения следующего фрагмента этой программы (записанного ниже на пяти языках программирования).

Бейсик	Python
<pre>s = 0 n = 1 FOR i = 0 TO 11   IF A(i) &gt; A(n) THEN     s = s + A(i) + i   ELSE     A(n) = A(i)   END IF NEXT i</pre>	<pre>s = 0 n = 1 for i in range(0, 12):   if A[i] &gt; A[n]:     s += A[i] + i   else:     A[n] = A[i]</pre>
Алгоритмический язык	Паскаль
<pre>s := 0; n := 1; нц для i от 0 до 11   если A[i] &gt; A[n] то     s := s + A[i] + i   иначе     A[n] := A[i] все кц</pre>	<pre>s := 0; n := 1; for i := 0 to 11 do   if A[i] &gt; A[n] then     s := s + A[i] + i   else     A[n] := A[i]; end do</pre>
C++	
<pre>s = 0; n = 1; for (int i = 0; i &lt; 12; i++) {   if (A[i] &gt; A[n])     s += A[i] + i;   else     A[n] = A[i]; }</pre>	

Ответ: \_\_\_\_\_.

- 20 Ниже на пяти языках программирования записан алгоритм. Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите **наибольшее** число  $x$ , при вводе которого алгоритм выводит сначала 2, а потом 3.

Бейсик	Python
<pre>DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 WHILE X &gt; 0   M = M + 1   IF X MOD 2 &lt;&gt; 0 THEN     L = L + X MOD 8   END IF   X = X \ 8 WEND PRINT L PRINT M</pre>	<pre>x = int(input()) L = 0 M = 0 while x &gt; 0:   M = M + 1   if x % 2 != 0:     L = L + x % 8   x = x // 8 print(L) print(M)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач   цел x, L, M   ввод x   L := 0   M := 0   нц пока x &gt; 0     M := M + 1     если mod(x, 2) &lt;&gt; 0       то         L := L + mod(x, 8)     все     x := div(x, 8)   кц   вывод L, M кон</pre>	<pre>var x, L, M: integer; begin   readln(x);   L := 0;   M := 0;   while x &gt; 0 do     begin       M := M + 1;       if x mod 2 &lt;&gt; 0 then         L := L + x mod 8;       x := x div 8;     end;     writeln(L);     writeln(M);   end.</pre>

```

C++
#include <iostream>
using namespace std;

int main(){
    int x, L, M;
    cin >> x;
    L = 0;
    M = 0;
    while (x > 0) {
        M = M + 1;
        if(x % 2 != 0) {
            L = L + x % 8;
        }
        x = x / 8;
    }
    cout << L << endl << M << endl;
    return 0;
}

```

Ответ: \_\_\_\_\_.

- 21** Определите **наибольшее** значение входной переменной  $k$ , при котором программа выдаёт тот же ответ, что и при входном значении  $k=27$ . Для Вашего удобства программа приведена на пяти языках программирования.

<b>Бейсик</b>	<pre> DIM K, I AS LONG INPUT K I = 1 WHILE F(I) &lt; G(K)     I = I + 1 WEND PRINT I  FUNCTION F(N)     F = N * N * N END FUNCTION  FUNCTION G(N)     G = 2 * N + 2 END FUNCTION </pre>
---------------	---

<b>Python</b>	<pre> def F(n):     return n*n*n  def G(n):     return 2 * n + 2  k = int(input()) i = 1 while F(i) &lt; G(k):     i+=1 print (i) </pre>
<b>Алгоритмический язык</b>	<pre> алг нач     цел i, k     ввод k     i := 1     нц пока f(i) &lt; g(k)         i := i + 1     кц     вывод i кон  алг цел f(цел n) нач     знач := n * n * n кон  алг цел g(цел n) нач     знач := 2 * n + 2 кон </pre>
<b>Паскаль</b>	<pre> var     k, i : longint;  function F(n: longint): longint; begin     F := n * n * n; end;  function G(n: longint): longint; begin     G := 2 * n + 2; end;  begin     readln(k);     i := 1;     while F(i) &lt; G(k) do         i := i + 1;     writeln(i) end. </pre>

```

C++
#include <iostream>
using namespace std;

long F(long n) {
    return n * n * n;
}

long G(long n) {
    return 2 * n + 2;
}

int main()
{
    long k, i;
    cin >> k;
    i = 1;
    while(F(i) < G(k))
        i++;
    cout << i;
    return 0;
}

```

Ответ: \_\_\_\_\_.

22

Исполнитель Вычислитель преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

**1. Прибавить 1**

**2. Умножить на 2**

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для Вычислителя – это последовательность команд.

Сколько существует программ, для которых при исходном **числе 1** результатом является **число 21** и при этом траектория вычислений содержит **число 10** и не содержит **числа 18**?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы **121** при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

Ответ: \_\_\_\_\_.

23

Сколько существует различных наборов значений логических переменных  $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$ , которые удовлетворяют всем перечисленным ниже условиям?

$$x_1 \rightarrow y_1 = 1$$

$$(x_2 \rightarrow (x_1 \wedge y_2)) \wedge (y_2 \rightarrow y_1) = 1$$

$$(x_3 \rightarrow (x_2 \wedge y_3)) \wedge (y_3 \rightarrow y_2) = 1$$

...

$$(x_7 \rightarrow (x_6 \wedge y_7)) \wedge (y_7 \rightarrow y_6) = 1$$

В ответе **не нужно** перечислять все различные наборы значений переменных  $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$ , при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ: \_\_\_\_\_.



*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы. Проверьте, чтобы каждый ответ был записан в строке с номером соответствующего задания.*



**Часть 2**

Для записи ответов на задания этой части (24–27) используйте **БЛАНК ОТВЕТОВ № 2**. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.

- 24** Требовалось написать программу, которая получает на вход натуральное число  $N$ , не превосходящее  $10^9$ , и выводит число, равное количеству цифр 4 в десятичной записи числа  $N$ . Программист написал программу неправильно. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre>DIM N AS LONG DIM R, d AS INTEGER INPUT N R = 0 WHILE N &gt; 0   d = N MOD 10   IF d &lt;&gt; 4 THEN     R = R + d   END IF   N = N \ 10 WEND PRINT R END</pre>	<pre>N = int(input()) R = 0 while N &gt; 0:   d = N % 10   if d != 4:     R = R + d   N = N // 10 print(R)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач   цел N, R, d   ввод N   R := 0   нц пока N &gt; 0     d := mod(N, 10)     если d &lt;&gt; 4 то       R := R + d     все     N := div(N, 10)   кц   вывод R кон</pre>	<pre>var N: longint;     R, d: integer; begin   readln(N);   R := 0;   while N &gt; 0 do begin     d := N mod 10;     if d &lt;&gt; 4 then       R := R + d;     N := N div 10;   end;   writeln(R); end.</pre>

**C++**

```
#include <iostream>
using namespace std;

int main()
{
  long int N;
  int R, d;
  cin >> N;
  R = 0;
  while (N > 0) {
    d = N % 10;
    if (d != 4) {
      R = R + d;
    }
    N = N / 10;
  }
  cout << R << endl;
  return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 241.
2. Приведите пример входного числа  $N$ , при котором приведённая программа, несмотря на ошибки, выдаёт верный ответ.
3. Найдите допущенные программистом ошибки и исправьте их. Исправление ошибки должно затрагивать только строку, в которой находится ошибка. Для каждой ошибки:

- 1) выпишите строку, в которой сделана ошибка;
- 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Известно, что в тексте программы нужно исправить не более двух строк так, чтобы она стала работать правильно.

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

Обратите внимание на то, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

**25** Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит количество элементов массива, больших 100 и при этом не кратных 4, а затем заменяет каждый такой элемент на число, равное найденному количеству. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести изменённый массив, каждый элемент выводится с новой строки.

Например, для исходного массива из шести элементов:

141  
256  
92  
148  
511  
4

программа должна вывести следующий массив:

2  
256  
92  
148  
2  
4

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

Бейсик	Python
<pre>CONST N AS INTEGER = 30 DIM A (1 TO N) AS LONG DIM I AS LONG,       J AS LONG,       K AS LONG  FOR I = 1 TO N     INPUT A(I) NEXT I ...</pre>	<pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 30 for i in range(0, n):     a.append(int(input())) ...</pre>

Алгоритмический язык	Паскаль
<pre>алг нач     цел N = 30     целтаб a[1:N]     цел i, j, k     нц для i от 1 до N         ввод a[i]     кц     ... кон</pre>	<pre>const     N = 30; var     a: array [1..N] of longint;     i, j, k: longint; begin     for i := 1 to N do         readln(a[i]);     ... end.</pre>
<b>C++</b> <pre>#include &lt;iostream&gt; using namespace std; const int N = 30; int main() {     long a[N];     long i, j, k;     for (i = 0; i &lt; N; i++)         cin &gt;&gt; a[i];     ...     return 0; }</pre>	

В качестве ответа Вам необходимо привести фрагмент программы, 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). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на Алгоритмическом языке).

26

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 **четыре** камня либо увеличить количество камней в куче **в пять раз**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 19 или 75 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 63.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 63 или больше камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 62$ .

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. В описание выигрышной стратегии **не следует** включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

#### Задание 1

- Укажите все такие значения числа  $S$ , при которых Петя может выиграть за один ход.
- 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.

#### Задание 2

Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для каждого указанного значения  $S$  опишите выигрышную стратегию Пети.

#### Задание 3

Укажите значение  $S$ , при котором одновременно выполняются два условия:

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

Дерево не должно содержать партии, невозможные при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не является верным ответом на это задание.

27

Дана последовательность  $N$  целых положительных чисел. Рассматриваются все пары элементов последовательности, разность которых чётна, и в этих парах, по крайней мере, одно из чисел пары делится на 17. Порядок элементов в паре неважен. Среди всех таких пар нужно найти и вывести пару с максимальной суммой элементов. Если одинаковую максимальную сумму имеет несколько пар, можно вывести любую из них. Если подходящих пар в последовательности нет, нужно вывести два нуля.

#### Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел  $N$  ( $2 \leq N \leq 10\,000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000.

*Пример входных данных:*

```
5
34
12
51
52
51
```

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

```
51 51
```

*Пояснение.* Из данных пяти чисел можно составить три различные пары, удовлетворяющие условию: (34, 12), (34, 52), (51, 51). Наибольшая сумма получается в паре (51, 51). Эта пара допустима, так как число 51 встречается в исходной последовательности дважды.

Напишите эффективную по времени и памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел  $N$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 Кбайт и не увеличивается с ростом  $N$ .

Максимальная оценка за правильную (не содержащую синтаксических ошибок) и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени или только по памяти, – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать **одну** или **две** программы решения задачи. Если Вы сдадите две программы, каждая из них будет оцениваться независимо от другой, итоговой станет **большая** из двух оценок.

Перед текстом программы кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.



**Проверьте, чтобы каждый ответ был записан рядом с номером соответствующего задания.**

## Единый государственный экзамен по ИНФОРМАТИКЕ и ИКТ

### Инструкция по выполнению работы

Экзаменационная работа состоит из двух частей, включающих в себя 27 заданий. Часть 1 содержит 23 задания с кратким ответом. Часть 2 содержит 4 задания с развёрнутым ответом.

На выполнение экзаменационной работы по информатике и ИКТ отводится 3 часа 55 минут (235 минут).

Ответы к заданиям 1–23 записываются в виде числа, последовательности букв или цифр. Ответ запишите в поле ответа в тексте работы, а затем перенесите в бланк ответов № 1.

КИМ

Ответ: 23.

2	3															
---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Бланк

Задания 24–27 требуют развёрнутого решения. В бланке ответов № 2 укажите номер задания и запишите его полное решение.

Все бланки ЕГЭ заполняются яркими чёрными чернилами. Допускается использование гелевой или капиллярной ручки.

**При выполнении заданий можно пользоваться черновиком. Записи в черновике, а также в тексте контрольных измерительных материалов не учитываются при оценивании работы.**

Баллы, полученные Вами за выполненные задания, суммируются. Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов.

После завершения работы проверьте, чтобы ответ на каждое задание в бланках ответов № 1 и № 2 был записан под правильным номером.

**Желаем успеха!**

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связок (операций):

- a) *отрицание* (инверсия, логическое НЕ) обозначается  $\neg$  (например,  $\neg A$ );
- b) *конъюнкция* (логическое умножение, логическое И) обозначается  $\wedge$  (например,  $A \wedge B$ ) либо  $\&$  (например,  $A \& B$ );
- c) *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается  $\vee$  (например,  $A \vee B$ ) либо  $|$  (например,  $A | B$ );
- d) *следование* (импликация) обозначается  $\rightarrow$  (например,  $A \rightarrow B$ );
- e) *тождество* обозначается  $\equiv$  (например,  $A \equiv B$ ). Выражение  $A \equiv B$  истинно тогда и только тогда, когда значения  $A$  и  $B$  совпадают (либо они оба истинны, либо они оба ложны);
- f) символ 1 используется для обозначения истины (истинного высказывания); символ 0 – для обозначения лжи (ложного высказывания).

2. Два логических выражения, содержащих переменные, называются *равносильными* (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения  $A \rightarrow B$  и  $(\neg A) \vee B$  равносильны, а  $A \vee B$  и  $A \wedge B$  неравносильны (значения выражений разные, например, при  $A = 1, B = 0$ ).

3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), тождество. Таким образом,  $\neg A \wedge B \vee C \wedge D$  означает то же, что и  $((\neg A) \wedge B) \vee (C \wedge D)$ .

Возможна запись  $A \wedge B \wedge C$  вместо  $(A \wedge B) \wedge C$ . То же относится и к дизъюнкции: возможна запись  $A \vee B \vee C$  вместо  $(A \vee B) \vee C$ .

4. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле – как обозначения единиц измерения, чьё соотношение с единицей «байт» выражается степенью двойки.

**Часть 1**

Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишете в отдельной клеточке в соответствии с приведёнными в бланке образцами.

**1** Сколько единиц в двоичной записи шестнадцатеричного числа E0F3<sub>16</sub>?

Ответ: \_\_\_\_\_.

**2** Миша заполнял таблицу истинности функции  $(x \wedge \neg y) \vee (x \equiv z) \vee \neg w$ , но успел заполнить лишь фрагмент из трёх **различных** её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ .

				$(x \wedge \neg y) \vee (x \equiv z) \vee \neg w$
		0	0	<b>0</b>
1	1	1	0	<b>0</b>
1	0			<b>0</b>

Определите, какому столбцу таблицы соответствует каждая из переменных  $w, x, y, z$ .

В ответе напишите буквы  $w, x, y, z$  в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

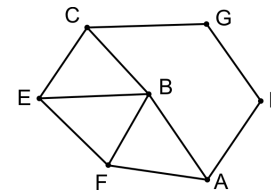
Пример. Функция задана выражением  $\neg x \vee y$ , зависящим от двух переменных, а фрагмент таблицы имеет следующий вид.

		$\neg x \vee y$
0	1	<b>0</b>

В этом случае первому столбцу соответствует переменная  $y$ , а второму столбцу – переменная  $x$ . В ответе следует написать  $yx$ .

Ответ: \_\_\_\_\_.

**3** На рисунке слева изображена схема дорог N-ского района. В таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет.



		Номер пункта						
		1	2	3	4	5	6	7
Номер пункта	1			*	*	*	*	
	2					*		*
	3	*					*	*
	4	*				*	*	
	5	*	*		*			
	6	*		*	*			
	7		*	*				

Каждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам E и F на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

Ответ: \_\_\_\_\_.

- 4 Ниже представлены два фрагмента таблиц из базы данных о жителях микрорайона. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. Определите на основании приведённых данных, у скольких детей на момент их рождения матерям было больше 29 полных лет. При вычислении ответа учитывайте только информацию из приведённых фрагментов таблиц.

Таблица 1				Таблица 2	
ID	Фамилия И.О.	Пол	Год рождения	ID_Родителя	ID_Ребёнка
14	Краснова Н.А.	Ж	1951	24	25
24	Кузьминых И.П.	М	1952	44	25
25	Кузьминых П.И.	М	1982	25	26
26	Кузьминых П.П.	М	2005	64	26
34	Красняк А.И.	Ж	1973	24	34
35	Красняк В.С.	Ж	1996	44	34
36	Красняк С.С.	М	1973	34	35
44	Воевода А.С.	Ж	1950	36	35
45	Воевода В.А.	М	1962	14	36
46	Макаренко О.С.	Ж	2001	34	46
47	Макаренко П.О.	М	2018	36	46
54	Клычко А.П.	Ж	2002	25	54
64	Крот П.А.	Ж	1974	64	54
...	...	...	...	...	...

Ответ: \_\_\_\_\_.

- 5 Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г, Д, Е, решили использовать неравномерный двоичный код, удовлетворяющий условию Фано. Для букв А, Б, В, Г использовали кодовые слова 100, 101, 00, 01 соответственно. Для двух оставшихся букв – Д и Е – коды неизвестны. Укажите кратчайшее возможное кодовое слово для буквы Д, при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите код с **наименьшим** числовым значением. *Примечание.* Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Ответ: \_\_\_\_\_.

- 6 На вход алгоритма подаётся натуральное число  $N$ . Алгоритм строит по нему новое число следующим образом.

1) Строится двоичная запись числа  $N$ .  
 2) К этой записи дописываются справа ещё два разряда по следующему правилу: если  $N$  чётное, в конец числа (справа) дописываются два нуля, в противном случае справа дописываются две единицы. Например, двоичная запись 1001 числа 9 будет преобразована в 100111. Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа  $N$ ) является двоичной записью числа – результата работы данного алгоритма. Укажите максимальное число  $N$ , для которого результат работы алгоритма будет меньше 134. В ответе это число запишите в десятичной системе счисления.

Ответ: \_\_\_\_\_.

- 7 Дан фрагмент электронной таблицы. Из ячейки В3 в ячейку D4 была скопирована формула. При копировании адреса ячеек в формуле автоматически изменились. Какова сумма числовых значений формул в ячейках В3 и D4?

	A	B	C	D	E
1	1	2	3	4	5
2	6	7	8	9	10
3	11	=\$D1+B\$1	13	14	15
4	16	17	18		20

*Примечание.* Знак \$ обозначает абсолютную адресацию.

Ответ: \_\_\_\_\_.

- 8 Запишите число, которое будет напечатано в результате выполнения следующей программы. Для Вашего удобства программа представлена на пяти языках программирования.

Бейсик	Python
<pre>DIM S, N AS INTEGER S = 175 N = 0 WHILE S + N &lt; 325   S = S - 10   N = N + 30 WEND PRINT N</pre>	<pre>s = 175 n = 0 while s + n &lt; 325:   s = s - 10   n = n + 30 print(n)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач   цел n, s   s := 175   n := 0   нц пока s + n &lt; 325     s := s - 10     n := n + 30   кц   вывод n кон</pre>	<pre>var s, n: integer; begin   s := 175;   n := 0;   while s + n &lt; 325 do     begin       s := s - 10;       n := n + 30     end;   writeln(n) end.</pre>
C++	
<pre>#include &lt;iostream&gt; using namespace std;  int main() {   int s = 175, n = 0;   while (s + n &lt; 325) {     s = s - 10;     n = n + 30;   }   cout &lt;&lt; n &lt;&lt; endl;   return 0; }</pre>	

Ответ: \_\_\_\_\_.

- 9 Музыкальный фрагмент был записан в формате стерео (двухканальная запись), оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла без учёта размера заголовка файла – 48 Мбайт. Затем тот же музыкальный фрагмент был записан повторно в формате моно и оцифрован с разрешением в 1,5 раза выше и частотой дискретизации в 3 раза меньше, чем в первый раз. Сжатие данных не производилось. Укажите размер в Мбайт файла, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно. Искомый объём не учитывает размера заголовка файла.

Ответ: \_\_\_\_\_.

- 10 Вася составляет 4-буквенные слова, в которых есть только буквы Б, Е, Л, К, А, причём буква Б используется в каждом слове ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

Ответ: \_\_\_\_\_.

**11** Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик	Python
<pre>SUB F(n)   IF n &gt; 0 THEN     F(n - 3)     PRINT n,     F(n \ 2)   END IF END SUB</pre>	<pre>def F(n):   if n &gt; 0:     F(n - 3)     print(n)     F(n // 2)</pre>
Алгоритмический язык	Паскаль
<pre>алг F(цел n) нач   если n &gt; 0 то     F(n - 3)     вывод n     F(div(n, 2))   все кон</pre>	<pre>procedure F(n: integer); begin   if n &gt; 0 then     begin       F(n - 3);       write(n);       F(n div 2)     end end;</pre>
C++	
<pre>void F(int n){   if (n &gt; 0){     F(n - 3);     std::cout &lt;&lt; n;     F(n / 2);   } }</pre>	

Запишите подряд без пробелов и разделителей все числа, которые будут выведены на экран при выполнении вызова F(7). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

Ответ: \_\_\_\_\_.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, – в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда – нули. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 111.81.224.27 адрес сети равен 111.81.192.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Ответ: \_\_\_\_\_.

**13** При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 25 символов и содержащий только символы из 7-символьного набора: С, Д, А, М, Е, Г, Э. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 100 пользователях потребовалось 2400 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число – количество байт.

Ответ: \_\_\_\_\_.



**14** Исполнитель Редактор получает на вход строку символов и преобразовывает её.

Редактор может выполнять две команды, в обеих командах  $v$  и  $w$  обозначают цепочки символов.

**А) заменить** ( $v, w$ ).

Эта команда заменяет в строке первое слева вхождение цепочки  $v$  на цепочку  $w$ . Например, выполнение команды

**заменить** (111, 27)

преобразует строку 05111150 в строку 0527150.

Если в строке нет вхождений цепочки  $v$ , то выполнение команды **заменить** ( $v, w$ ) не меняет эту строку.

**Б) нашлось** ( $v$ ).

Эта команда проверяет, встречается ли цепочка  $v$  в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА *условие*

*последовательность команд*

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*

*ТО команда1*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно).

В конструкции

ЕСЛИ *условие*

*ТО команда1*

*ИНАЧЕ команда2*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно) или *команда2* (если условие ложно).

На вход приведённой ниже программе поступает строка, начинающаяся с символа «>», а затем содержащая 11 цифр 1, 12 цифр 2 и 30 цифр 3, расположенных в произвольном порядке.

Определите сумму числовых значений цифр строки, получившейся в результате выполнения программы.

Так, например, если результат работы программы представлял бы собой строку, состоящую из 50 цифр 4, то верным ответом было бы число 200.

НАЧАЛО

ПОКА **нашлось** (>1) ИЛИ **нашлось** (>2) ИЛИ **нашлось** (>3)

ЕСЛИ **нашлось** (>1)

ТО **заменить** (>1, 22>)

КОНЕЦ ЕСЛИ

ЕСЛИ **нашлось** (>2)

ТО **заменить** (>2, 2>)

КОНЕЦ ЕСЛИ

ЕСЛИ **нашлось** (>3)

ТО **заменить** (>3, 1>)

КОНЕЦ ЕСЛИ

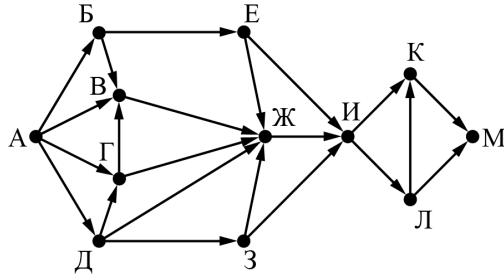
КОНЕЦ ПОКА

КОНЕЦ

Ответ: \_\_\_\_\_.

- 15 На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Какова длина самого длинного пути из города А в город М? Длиной пути считать количество дорог, составляющих этот путь.



Ответ: \_\_\_\_\_.

- 16 Значение арифметического выражения:  $9^8 \times 3^{20} - 3^{10} - 3$  – записали в системе счисления с основанием 3. Сколько цифр 2 содержится в этой записи?

Ответ: \_\_\_\_\_.

- 17 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в сотнях тысяч)
<i>Физика</i>	43
<i>Квант</i>	32
<i>Ньютон</i>	32
<i>Ньютон   Физика   Квант</i>	85
<i>Ньютон &amp; Физика</i>	11
<i>Ньютон &amp; Квант</i>	0

Какое количество страниц (в сотнях тысяч) будет найдено по запросу *Физика & Квант*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: \_\_\_\_\_.

- 18 Для какого наименьшего целого неотрицательного числа  $A$  выражение

$$(x + 2y < A) \vee (y > x) \vee (x > 30)$$

тождественно истинно, т.е. принимает значение 1 при любых целых неотрицательных  $x$  и  $y$ ?

Ответ: \_\_\_\_\_.

- 19 В программе используется одномерный целочисленный массив  $A$  с индексами от 0 до 11. Значения элементов массива  $A[i]$  приведены в таблице.

$i$	0	1	2	3	4	5	6	7	8	9	10	11
$A[i]$	14	13	15	8	4	12	30	21	22	16	5	9

Определите значение переменной  $s$  после выполнения следующего фрагмента этой программы (записанного ниже на пяти языках программирования).

Бейсик	Python
<pre>s = 0 n = 0 FOR i = 1 TO 11   IF A(i) &lt; A(n) * i THEN     s = s + A(i)   ELSE     A(n) = A(i)   END IF NEXT i</pre>	<pre>s = 0 n = 0 for i in range(1, 12):   if A[i] &lt; A[n] * i:     s += A[i]   else:     A[n] = A[i]</pre>
Алгоритмический язык	Паскаль
<pre>s := 0; n := 0; нц для i от 1 до 11   если A[i] &lt; A[n] * i то     s := s + A[i]   иначе     A[n] := A[i] все кц</pre>	<pre>s := 0; n := 0; for i := 1 to 11 do   if A[i] &lt; A[n] * i then     s := s + A[i]   else     A[n] := A[i]; end do</pre>
C++	
<pre>s = 0; n = 0; for (int i = 1; i &lt; 12; i++) {   if (A[i] &lt; A[n] * i)     s += A[i];   else     A[n] = A[i]; }</pre>	

Ответ: \_\_\_\_\_.

- 20 Ниже на пяти языках программирования записан алгоритм. Получив на вход натуральное десятичное число  $x$ , этот алгоритм печатает два числа:  $L$  и  $M$ . Укажите **наибольшее** число  $x$ , при вводе которого алгоритм выводит сначала 12, а потом 3.

Бейсик	Python
<pre>DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 WHILE X &gt; 0   M = M + 1   IF X MOD 2 = 0 THEN     L = L + X MOD 8   END IF   X = X \ 8 WEND PRINT L PRINT M</pre>	<pre>x = int(input()) L = 0 M = 0 while x &gt; 0:   M = M + 1   if x % 2 == 0:     L = L + x % 8   x = x // 8 print(L) print(M)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач   цел x, L, M   ввод x   L := 0   M := 0   нц пока x &gt; 0     M := M + 1     если mod(x, 2) = 0       то         L := L + mod(x, 8)       все     x := div(x, 8)   кц   вывод L, M кон</pre>	<pre>var x, L, M: integer; begin   readln(x);   L := 0;   M := 0;   while x &gt; 0 do     begin       M := M + 1;       if x mod 2 = 0 then         L := L + x mod 8;       x := x div 8;     end;     writeln(L);     writeln(M);   end.</pre>

```

C++
#include <iostream>
using namespace std;

int main(){
    int x, L, M;
    cin >> x;
    L = 0;
    M = 0;
    while (x > 0) {
        M = M + 1;
        if(x % 2 == 0) {
            L = L + x % 8;
        }
        x = x / 8;
    }
    cout << L << endl << M << endl;
    return 0;
}

```

Ответ: \_\_\_\_\_.

- 21** Определите **наибольшее** значение входной переменной  $k$ , при котором программа выдаёт тот же ответ, что и при входном значении  $k=45$ . Для Вашего удобства программа приведена на пяти языках программирования.

<b>Бейсик</b>	<pre> DIM K, I AS LONG INPUT K I = 1 WHILE F(I) &lt; G(K)     I = I + 1 WEND PRINT I  FUNCTION F(N)     F = N * N * N END FUNCTION  FUNCTION G(N)     G = 2 * N + 3 END FUNCTION </pre>
<b>Python</b>	<pre> def F(n):     return n*n*n  def G(n):     return 2 * n + 3  k = int(input()) i = 1 while F(i) &lt; G(k):     i+=1 print (i) </pre>

<b>Алгоритмический язык</b>	<pre> алг нач     цел i, k     ввод k     i := 1     нц пока f(i) &lt; g(k)         i := i + 1     кц     вывод i кон  алг цел f(цел n) нач     знач := n * n * n кон  алг цел g(цел n) нач     знач := 2 * n + 3 кон </pre>
<b>Паскаль</b>	<pre> var     k, i : longint;  function F(n: longint): longint; begin     F := n * n * n; end;  function G(n: longint): longint; begin     G := 2 * n + 3; end;  begin     readln(k);     i := 1;     while F(i) &lt; G(k) do         i := i + 1;     writeln(i) end. </pre>

```

C++
#include <iostream>
using namespace std;

long F(long n) {
    return n * n * n;
}

long G(long n) {
    return 2 * n + 3;
}

int main()
{
    long k, i;
    cin >> k;
    i = 1;
    while(F(i) < G(k))
        i++;
    cout << i;
    return 0;
}

```

Ответ: \_\_\_\_\_.

22

Исполнитель Вычислитель преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

**1. Прибавить 1**

**2. Умножить на 2**

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для Вычислителя – это последовательность команд.

Сколько существует программ, для которых при исходном **числе 1** результатом является **число 22** и при этом траектория вычислений содержит **число 10** и не содержит **числа 15**?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы **121** при исходном числе **7** траектория будет состоять из чисел 8, 16, 17.

Ответ: \_\_\_\_\_.

23

Сколько существует различных наборов значений логических переменных  $x_1, x_2, \dots, x_9, y_1, y_2, \dots, y_9$ , которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow (x_2 \wedge y_2)) \wedge (y_1 \rightarrow y_2) = 1$$

$$(x_2 \rightarrow (x_3 \wedge y_3)) \wedge (y_2 \rightarrow y_3) = 1$$

...

$$(x_8 \rightarrow (x_9 \wedge y_9)) \wedge (y_8 \rightarrow y_9) = 1$$

В ответе **не нужно** перечислять все различные наборы значений переменных  $x_1, x_2, \dots, x_9, y_1, y_2, \dots, y_9$ , при которых выполнена данная система равенств. В качестве ответа Вам нужно указать количество таких наборов.

Ответ: \_\_\_\_\_.



*Не забудьте перенести все ответы в бланк ответов №1 в соответствии с инструкцией по выполнению работы. Проверьте, чтобы каждый ответ был записан в строке с номером соответствующего задания.*

**Часть 2**

**Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.**

- 24** Требовалось написать программу, которая получает на вход натуральное число  $N$ , не превосходящее  $10^9$ , и выводит число, равное количеству цифр 2 в десятичной записи числа  $N$ . Программист написал программу неправильно. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre>DIM N AS LONG DIM R, d AS INTEGER INPUT N R = 0 WHILE N &gt; 0   d = N MOD 10   IF d &lt;&gt; 2 THEN     R = R + 1   END IF   N = N \ 10 WEND PRINT d END</pre>	<pre>N = int(input()) R = 0 while N &gt; 0:   d = N % 10   if d != 2:     R = R + 1   N = N // 10 print(d)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач   цел N, R, d   ввод N   R := 0   нц пока N &gt; 0     d := mod(N, 10)     если d &lt;&gt; 2 то       R := R + 1     все     N := div(N, 10)   кц   вывод d кон</pre>	<pre>var N: longint;     R, d: integer; begin   readln(N);   R := 0;   while N &gt; 0 do begin     d := N mod 10;     if d &lt;&gt; 2 then       R := R + 1;     N := N div 10;   end;   writeln(d); end.</pre>

**C++**

```
#include <iostream>
using namespace std;

int main()
{
  long int N;
  int R, d;
  cin >> N;
  R = 0;
  while (N > 0) {
    d = N % 10;
    if (d != 2) {
      R = R + 1;
    }
    N = N / 10;
  }
  cout << d << endl;
  return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 324.
2. Приведите пример входного числа  $N$ , при котором приведённая программа, несмотря на ошибки, выдаёт верный ответ.
3. Найдите допущенные программистом ошибки и исправьте их. Исправление ошибки должно затрагивать только строку, в которой находится ошибка. Для каждой ошибки:

- 1) выпишите строку, в которой сделана ошибка;
- 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Известно, что в тексте программы нужно исправить не более двух строк так, чтобы она стала работать правильно.

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

Обратите внимание на то, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

**25** Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит сумму элементов массива, больших 100 и при этом не кратных 4, а затем заменяет каждый такой элемент на число, равное найденной сумме. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести изменённый массив, каждый элемент выводится с новой строки.

Например, для исходного массива из шести элементов:

101  
128  
6  
105  
4  
18

программа должна вывести следующий массив:

206  
128  
6  
206  
4  
18

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

Бейсик	Python
<pre>CONST N AS INTEGER = 30 DIM A (1 TO N) AS LONG DIM I AS LONG,       J AS LONG,       K AS LONG  FOR I = 1 TO N   INPUT A(I) NEXT I ...</pre>	<pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 30 for i in range(0, n):     a.append(int(input())) ...</pre>

Алгоритмический язык	Паскаль
<pre>алг нач   цел N = 30   целтаб a[1:N]   цел i, j, k   нц для i от 1 до N     ввод a[i]   кц   ... кон</pre>	<pre>const   N = 30; var   a: array [1..N] of longint;   i, j, k: longint; begin   for i := 1 to N do     readln(a[i]);   ... end.</pre>
C++	
<pre>#include &lt;iostream&gt; using namespace std; const int N = 30; int main() {   long a[N];   long i, j, k;   for (i = 0; i &lt; N; i++)     cin &gt;&gt; a[i];   ...   return 0; }</pre>	

В качестве ответа Вам необходимо привести фрагмент программы, 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). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на Алгоритмическом языке).

26

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 **четыре** камня либо увеличить количество камней в куче **в пять раз**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 19 или 75 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 68.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 68 или больше камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 67$ .

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. В описание выигрышной стратегии **не следует** включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

#### Задание 1

- Укажите все такие значения числа  $S$ , при которых Петя может выиграть за один ход.
- 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.

#### Задание 2

Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для каждого указанного значения  $S$  опишите выигрышную стратегию Пети.

#### Задание 3

Укажите значение  $S$ , при котором одновременно выполняются два условия:

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

Дерево не должно содержать партии, невозможные при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не является верным ответом на это задание.

27

Дана последовательность  $N$  целых положительных чисел. Рассматриваются все пары элементов последовательности, разность которых чётна, и в этих парах, по крайней мере, одно из чисел пары делится на 19. Порядок элементов в паре неважен. Среди всех таких пар нужно найти и вывести пару с максимальной суммой элементов. Если одинаковую максимальную сумму имеет несколько пар, можно вывести любую из них. Если подходящих пар в последовательности нет, нужно вывести два нуля.

#### Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел  $N$  ( $2 \leq N \leq 10\,000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000.

*Пример входных данных:*

```
5
38
12
57
16
57
```

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

```
57 57
```

*Пояснение.* Из данных пяти чисел можно составить три различные пары, удовлетворяющие условию: (38, 12), (38, 16), (57, 57). Наибольшая сумма получается в паре (57, 57). Эта пара допустима, так как число 57 встречается в исходной последовательности дважды.

Напишите эффективную по времени и памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел  $N$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 Кбайт и не увеличивается с ростом  $N$ .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени или только по памяти, – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать **одну** или **две** программы решения задачи. Если Вы сдадите две программы, каждая из них будет оцениваться независимо от другой, итоговой станет **большая** из двух оценок.

Перед текстом программы кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.



**Проверьте, чтобы каждый ответ был записан рядом с номером соответствующего задания.**



**Ответы к заданиям**

№ задания	Ответ
1	8
2	xwzy
3	45
4	4
5	111
6	33
7	15
8	95
9	4
10	405
11	7412131
12	224
13	14
14	110
15	9
16	38
17	12
18	61
19	202
20	393
21	31
22	14
23	36

### Критерии оценивания заданий с развёрнутым ответом

- 24 Требовалось написать программу, которая получает на вход натуральное число  $N$ , не превосходящее  $10^9$ , и выводит число, равное количеству цифр 4 в десятичной записи числа  $N$ . Программист написал программу неправильно. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre>DIM N AS LONG DIM R, d AS INTEGER INPUT N R = 0 WHILE N &gt; 0   d = N MOD 10   IF d &lt;&gt; 4 THEN     R = R + d   END IF   N = N \ 10 WEND PRINT R END</pre>	<pre>N = int(input()) R = 0 while N &gt; 0:     d = N % 10     if d != 4:         R = R + d     N = N // 10 print(R)</pre>
Алгоритмический язык	Паскаль
<pre>алг нач   цел N, R, d   ввод N   R := 0   нц пока N &gt; 0     d := mod(N, 10)     если d &lt;&gt; 4 то       R := R + d     все     N := div(N, 10) кц вывод R кон</pre>	<pre>var N: longint;     R, d: integer; begin   readln(N);   R := 0;   while N &gt; 0 do begin     d := N mod 10;     if d &lt;&gt; 4 then       R := R + d;     N := N div 10;   end;   writeln(R); end.</pre>

#### C++

```
#include <iostream>
using namespace std;

int main()
{
  long int N;
  int R, d;
  cin >> N;
  R = 0;
  while (N > 0) {
    d = N % 10;
    if (d != 4) {
      R = R + d;
    }
    N = N / 10;
  }
  cout << R << endl;
  return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 241.
2. Приведите пример входного числа  $N$ , при котором приведённая программа, несмотря на ошибки, выдаёт верный ответ.
3. Найдите допущенные программистом ошибки и исправьте их. Исправление ошибки должно затрагивать только строку, в которой находится ошибка. Для каждой ошибки:
  - 1) выпишите строку, в которой сделана ошибка;
  - 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Известно, что в тексте программы нужно исправить не более двух строк так, чтобы она стала работать правильно.

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

Обратите внимание на то, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Решение использует запись программы на Паскале. Допускается использование программы на любом из четырёх других языков программирования.</p> <p>1. Программа выведет число 3.  <i>Примечание для эксперта.</i> Программа выводит сумму цифр, отличных от 4.</p> <p>2. Программа выдаёт правильный ответ, например, при <math>N = 244</math>.</p> <p>3. В программе есть две ошибки.</p> <p><b>Первая ошибка:</b> неверная проверка условия увеличения счетчика – переменной <math>R</math>.          Строка с ошибкой:  <code>if d &lt;&gt; 4 then</code>          Верное исправление:  <code>if d = 4 then</code></p> <p><b>Вторая ошибка:</b> неверно увеличивается значение переменной <math>R</math>.          Строка с ошибкой:  <code>R := R + d;</code>          Верное исправление:  <code>R := R + 1;</code></p>	
Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить <b>четыре</b> действия:</p> <ol style="list-style-type: none"> <li>1) указать, что выведет программа при конкретном входном числе;</li> <li>2) указать пример входного числа, при котором программа выдаёт верный ответ;</li> <li>3) исправить первую ошибку;</li> <li>4) исправить вторую ошибку.</li> </ol> <p>Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы.</p> <p>Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия:</p> <ol style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> <li>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа</li> </ol>	
Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3

Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций: а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной; б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два из четырёх необходимых действий	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

25

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит количество элементов массива, больших 100 и при этом не кратных 4, а затем заменяет каждый такой элемент на число, равное найденному количеству. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести изменённый массив, каждый элемент выводится с новой строки.

Например, для исходного массива из шести элементов:

141

256

92

148

511

4

программа должна вывести следующий массив:

2

256

92

148

2

4

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

Бейсик	Python
<pre>CONST N AS INTEGER = 30 DIM A (1 TO N) AS LONG DIM I AS LONG,       J AS LONG,       K AS LONG  FOR I = 1 TO N   INPUT A(I) NEXT I ... END</pre>	<pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 30 for i in range(0, n):   a.append(int(input())) ... </pre>
Алгоритмический язык	Паскаль
<pre><u>алг</u> <u>нач</u>   цел N = 30   <u>целтаб</u> a[1:N]   цел i, j, k   <u>нц для i от 1 до N</u>     <u>ввод</u> a[i]   <u>кц</u>   ... <u>кон</u></pre>	<pre>const   N = 30; var   a: array [1..N] of longint;   i, j, k: longint; begin   for i := 1 to N do     readln(a[i]);   ... end.</pre>
C++	
<pre>#include &lt;iostream&gt; using namespace std; const int N = 30; int main() {   long a[N];   long i, j, k;   for (i = 0; i &lt; N; i++)     cin &gt;&gt; a[i];   ...   return 0; }</pre>	

В качестве ответа Вам необходимо привести фрагмент программы, 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). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на Алгоритмическом языке).

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
<b>На языке Паскаль</b>
<pre>k := 0; for i := 1 to N do   if (a[i] &gt; 100) and (a[i] mod 4 &lt;&gt; 0) then     k := k + 1; for i := 1 to N do begin   if (a[i] &gt; 100) and (a[i] mod 4 &lt;&gt; 0) then     a[i] := k;   writeln(a[i]); end;</pre>
<b>На Алгоритмическом языке</b>
<pre>к := 0 нц для i от 1 до N   если a[i] &gt; 100 и mod(a[i], 4) &lt;&gt; 0     то       к := к + 1   все кц нц для i от 1 до N   если a[i] &gt; 100 и mod(a[i], 4) &lt;&gt; 0     то       a[i] := к   все вывод a[i], нс кц</pre>
<b>На языке Бейсик</b>
<pre>K = 0 FOR I = 1 TO N   IF A(I) &gt; 100 AND A(I) MOD 4 &lt;&gt; 0 THEN     K = K + 1   END IF NEXT I FOR I = 1 TO N   IF A(I) &gt; 100 AND A(I) MOD 4 &lt;&gt; 0 THEN     A(I) = K   END IF   PRINT A(I) NEXT I</pre>
<b>На языке C++</b>
<pre>k = 0; for (i = 0; i &lt; N; i++)   if (a[i] &gt; 100 &amp;&amp; a[i] % 4 != 0)     k++; for (i = 0; i &lt; N; i++) {   if (a[i] &gt; 100 &amp;&amp; a[i] % 4 != 0)     a[i] = k;   cout &lt;&lt; a[i] &lt;&lt; endl; }</pre>

<b>На языке Python</b>	
<pre>k = 0 for i in range(0, n):     if (a[i] &gt; 100 and a[i] % 4 != 0):         k = k + 1 for i in range(0, n):     if (a[i] &gt; 100 and a[i] % 4 != 0):         a[i] = k print(a[i])</pre>	
Указания по оцениванию	Баллы
<p><i>Общие указания.</i></p> <p>1. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.</p> <p>2. Эффективность алгоритма не имеет значения и не оценивается.</p> <p>3. Допускается запись алгоритма на языке программирования, отличном от языков, приведённых в условии. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на Алгоритмическом 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.</p> <p>4. Допускается формат вывода массива, отличный от указанного, например в строчку (в том числе без разделителей)</p>	
Предложен правильный алгоритм, который изменяет исходный массив и выводит в качестве результата изменённый массив	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"> <li>1) в цикле происходит выход за границу массива;</li> <li>2) не инициализируется или неверно инициализируется количество найденных элементов;</li> <li>3) неверно осуществляется проверка делимости на 4;</li> <li>4) проверяется делимость на 4 не элемента массива, а его индекса;</li> <li>5) неверно выбрана операция отношения для сравнения с границей диапазона;</li> <li>6) сравнение с границей диапазона производится для индекса элемента массива, а не для его значения;</li> <li>7) неверно составлено логическое условие (например, используется <code>or</code> вместо <code>and</code>);</li> <li>8) не вычисляется или неверно вычисляется количество найденных элементов;</li> </ol>	1

<p>9) исходный массив не изменяется;</p> <p>10) отсутствует вывод ответа, или ответ выводится не полностью (например, только один элемент массива ввиду пропущенного цикла вывода элементов или операторных скобок);</p> <p>11) используется переменная, не объявленная в разделе описания переменных;</p> <p>12) не указано или неверно указано условие завершения цикла;</p> <p>13) индексная переменная в цикле не меняется (например, в цикле <code>while</code>) или меняется неверно</p>	
Ошибок, перечисленных в п. 1–13, две или больше, или алгоритм сформулирован неверно (в том числе при отсутствии в явном или неявном виде цикла подсчёта количества нужных элементов)	0
<i>Максимальный балл</i>	2

26

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 **четыре** камня либо увеличить количество камней в куче **в пять раз**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 19 или 75 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 63.

Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 63 или больше камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 62$ .

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. В описание выигрышной стратегии **не следует** включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

#### Задание 1

- а) Укажите все такие значения числа  $S$ , при которых Петя может выиграть за один ход.
- 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.

**Задание 2**

Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для каждого указанного значения  $S$  опишите выигрышную стратегию Пети.

**Задание 3**

Укажите значение  $S$ , при котором одновременно выполняются два условия:

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

Дерево не должно содержать партии, невозможные при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не является верным ответом на это задание.

**Содержание верного ответа и указания по оцениванию**

(допускаются иные формулировки ответа, не искажающие его смысла)

**Задание 1**

- а) Петя может выиграть, если  $S = 13, \dots, 62$ .  
 б) Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче  $S = 12$  камней. Тогда после первого хода Пети в куче будет 13, 16 или 60 камней. Во всех случаях Ваня увеличивает количество камней в 5 раз и выигрывает за один ход.

*Замечание для проверяющего.* В задаче не требуется указать **все** выигрышные стратегии. Если в работе выпускника, как в приведённом примере, просто указано, что Ваня всегда увеличивает в 5 раз количество камней, – это не ошибка.

**Задание 2**

Возможные значения  $S$ : 8, 11. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 12 камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (т.е. Петя) следующим ходом выигрывает.

**Задание 3**

Возможные значения  $S$ : 7, 10.

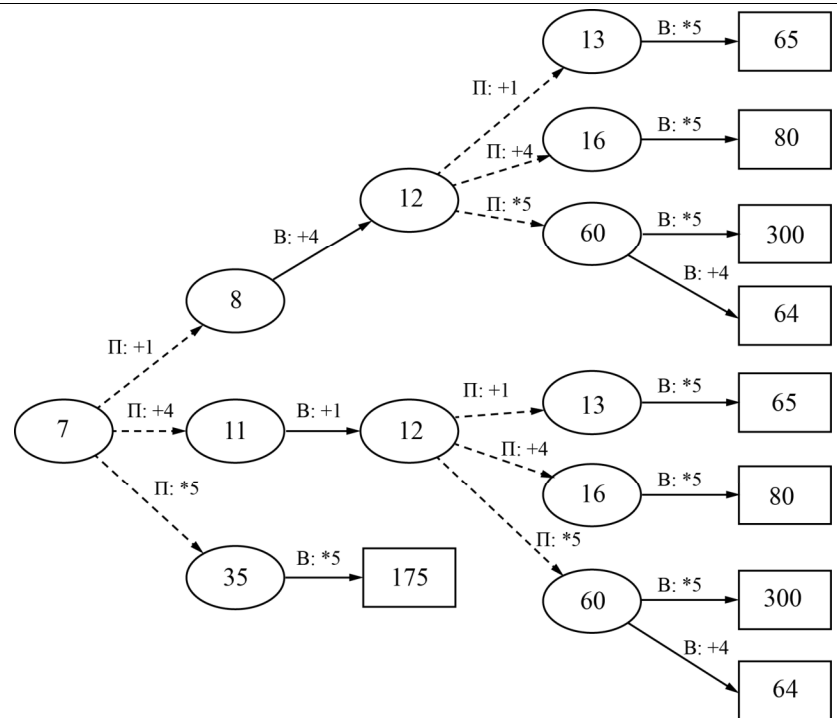
Например, для  $S = 7$  после первого хода Пети в куче будет 8, 11 или

35 камней. Если в куче станет 35 камней, Ваня увеличит количество камней в 5 раз и выигрывает первым ходом. Ситуация, когда в куче 8 или 11 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий (и только их) при описанной стратегии Вани для значения  $S = 7$ . При выбранной стратегии на последнем ходе Ваня увеличивает в 5 раз количество камней, хотя возможны и другие выигрышные стратегии. Для второго возможного значения дерева строится аналогично. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

*Примечание для проверяющего.* Здесь для полноты картины указаны два возможных значения  $S$ . По условию задачи достаточно указать одно из них

Исходное положение	Положения после очередных ходов			
	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)
7	$7 + 1 = 8$	$8 + 4 = 12$	$12 + 1 = 13$	<u><math>13 * 5 = 65</math></u>
			$12 + 4 = 16$	<u><math>16 * 5 = 80</math></u>
			$12 * 5 = 60$	<u><math>60 * 5 = 300</math></u> <u><math>60 + 4 = 64</math></u>
	$7 + 4 = 11$	$11 + 1 = 12$	$12 + 1 = 13$	<u><math>13 * 5 = 65</math></u>
			$12 + 4 = 16$	<u><math>16 * 5 = 80</math></u>
			$12 * 5 = 60$	<u><math>60 * 5 = 300</math></u> <u><math>60 + 4 = 64</math></u>
$7 * 5 = 35$	<u><math>35 * 5 = 175</math></u>			



Дерево всех партий, возможных при Ваниной стратегии.

Прямоугольником обозначены позиции, в которых партия заканчивается

*Замечание для проверяющего.* На рисунке для наглядности ходы Пети показаны пунктиром, а заключительные позиции выделены прямоугольником. Это не является обязательным для экзаменуемых. Также не является ошибкой указание только одного заключительного хода Вани в ситуации, когда у него есть два заключительных выигрышных хода

Указания по оцениванию	Баллы
В задаче от выпускника 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.	
Задание 1 выполнено, если выполнены оба пункта: а) и б). Если хотя бы один из этих пунктов не выполнен или выполнен с	

ошибкой (кроме оговоренных выше ошибок), задание считается невыполненным.	
Задание 2 выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например с помощью дерева всех возможных при выбранной стратегии Пети партий (и только их).	
Задание 3 выполнено, если правильно указана позиция, выигрышная для Вани, и построено дерево всех возможных при Ваниной стратегии партий (и только их). Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом	
Выполнены задания 1, 2 и 3	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. 1. Выполнено задание 3. 2. Выполнены задания 1 и 2	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено одно из следующих условий. 1. Выполнено задание 1. 2. Выполнено задание 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	<b>3</b>

27

Дана последовательность  $N$  целых положительных чисел. Рассматриваются все пары элементов последовательности, разность которых чётна, и в этих парах, по крайней мере, одно из чисел пары делится на 17. Порядок элементов в паре неважен. Среди всех таких пар нужно найти и вывести пару с максимальной суммой элементов. Если одинаковую максимальную сумму имеет несколько пар, можно вывести любую из них. Если подходящих пар в последовательности нет, нужно вывести два нуля.

**Описание входных и выходных данных**  
В первой строке входных данных задаётся количество чисел  $N$  ( $2 \leq N \leq 10\,000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000.

*Пример входных данных:*

5  
34  
12  
51  
52  
51

Пример выходных данных для приведённого выше примера входных данных:  
51 51

Пояснение. Из данных пяти чисел можно составить три различные пары, удовлетворяющие условию: (34, 12), (34, 52), (51, 51). Наибольшая сумма получается в паре (51, 51). Эта пара допустима, так как число 51 встречается в исходной последовательности дважды.

Напишите эффективную по времени и памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел  $N$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 Кбайт и не увеличивается с ростом  $N$ .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени или только по памяти, – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать **одну** или **две** программы решения задачи. Если Вы сдадите две программы, каждая из них будет оцениваться независимо от другой, итоговой станет **бóльшая** из двух оценок.

Перед текстом программы кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.

#### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Разность двух чисел чётна, если эти числа имеют одинаковую чётность.

Будем вводить элементы последовательности по одному и хранить четыре элемента: максимальные чётный  $m_0$  и нечётный  $m_1$  элементы всей последовательности и максимальные чётный и нечётный элементы, кратные  $p = 17$  ( $m_{p0}$  и  $m_{p1}$  соответственно). При этом у чисел, кратных  $p$ , будет «приоритет»: если вновь найденный общий максимум последовательности (с учётом чётности) кратен  $p$ , это число будет записано в качестве максимума среди кратных  $p$  ( $m_{p0}$  или  $m_{p1}$ ). В общий максимум ( $m_0$  или  $m_1$ ) оно сможет перейти, только если среди кратных  $p$  ( $m_{p0}$  или  $m_{p1}$ ) появится большее или такое же значение.

После ввода всех элементов нужно найти значения  $m_{p0} + m_0$ ,  $m_{p1} + m_1$  и выбрать пару с большей суммой. При этом необходимо убедиться, что эти максимумы действительно выбраны из последовательности, а не записаны при инициализации нулевыми или отрицательными значениями.

Ниже приведены программы на языках Pascal (использована версия PascalABC) и Алгоритмическом, реализующие этот алгоритм

#### Пример 1. Программа на языке Паскаль. Программа эффективна по времени и памяти

```
const p = 17;
var
  N: integer;      {количество чисел}
  a: integer;      {очередное число}
  d: integer;      {чётность очередного числа}

  m: array [0..1] of integer; {чётный и нечётный максимумы}
  mp: array [0..1] of integer; {чётный и нечётный максимумы, кратные p}

  x, y: integer;   {ответ - пара чисел}
  i: integer;

begin
  m[0] := 0; m[1] := 0;
  mp[0] := 0; mp[1] := 0;
  x := 0; y := 0;
  readln(N);
  for i := 1 to N do
    begin
      readln(a);
      d := a mod 2;
      if (a mod p = 0) and (a >= mp[d]) then
        {нестрогое сравнение!}
        begin
          if mp[d] > m[d] then {допустимо нестрогое сравнение}
            m[d] := mp[d];
          mp[d] := a
        end
      else if a > m[d] then {допустимо нестрогое сравнение}
        m[d] := a
      end;
    if (mp[0] > 0) and (m[0] > 0) then
      x := mp[0]; y := m[0];
    if (mp[1] > 0) and (m[1] > 0) and (mp[1] + m[1] > x + y) then
      x := mp[1]; y := m[1];
    writeln(x, ' ', y)
  end.
```



**Пример 1а. Программа на Алгоритмическом языке. Программа эффективна по времени и памяти**

```

алг задача
нач
  цел p = 17 | делитель из условия задачи
  цел N | количество чисел
  цел a | очередное число
  цел d | чётность очередного числа
  целтаб m[0:1] | чётный и нечётный максимумы
  целтаб mp[0:1] | чётный и нечётный максимумы, кратные p
  цел x, y | пара чисел для ответа

  m[0] := 0; m[1] := 0
  mp[0] := 0; mp[1] := 0
  ввод N
  нц N раз
    ввод a
    d := mod(a, 2)
    если mod(a, p) = 0 и a >= mp[d] | нужно нестрогое сравнение!
      то если mp[d] > m[d] | допустимо нестрогое сравнение
        то m[d] := mp[d]
      все
        mp[d] := a
    иначе если a > m[d] | допустимо нестрогое сравнение
      то m[d] := a
    все
  кц
  x := 0; y := 0
  если mp[0] > 0 и m[0] > 0
    то x := mp[0]; y := m[0]
  все
  если mp[1] > 0 и m[1] > 0 и mp[1] + m[1] > x + y
    то x := mp[1]; y := m[1]
  все
  вывод x, ' ', y
кон

```

В приведённом решении вместо массивов  $m$  и  $mp$  можно использовать две пары простых переменных. В этом случае программа получится более громоздкой (придётся отдельно разбирать случаи чётного и нечётного чисел в последовательности и дублировать аналогичные фрагменты программы), но останется эффективной. Ниже приводится реализующая такой алгоритм программа на языке Python 3

**Пример 2. Программа на языке Python 3. Программа эффективна по времени и памяти**

```

p = 17
m0 = m1 = mp0 = mp1 = 0
N = int(input())
for i in range(N):
    a = int(input())
    if a % 2 == 0:
        if a % p == 0 and a >= mp0:
            if mp0 > m0: m0 = mp0
            mp0 = a
        elif a > m0: m0 = a
    else:
        if a % p == 0 and a >= mp1:
            if mp1 > m1: m1 = mp1
            mp1 = a
        elif a > m1: m1 = a
x = y = 0
if mp0 > 0 and m0 > 0:
    x = mp0; y = m0
if mp1 > 0 and m1 > 0 and mp1 + m1 > x + y:
    x = mp1; y = m1
print(x, y)

```

Ещё один путь решения – записать всю последовательность в массив и анализировать её в несколько проходов. Ниже приводится реализующая такой алгоритм программа на языке C++. В этой программе массив с исходными данными обрабатывается два раза: на первом проходе находятся индексы максимального чётного и нечётного элементов, кратных  $p$ , на втором проходе – общие чётный и нечётный максимумы. При этом элементы, выделенные как кратные при первом проходе, во время второго прохода из сравнения исключаются. Такая программа эффективна по времени (несмотря на повторную обработку массива, общее время работы пропорционально  $N$ ), но неэффективна по памяти. Максимальная оценка за такую программу при отсутствии в ней синтаксических и содержательных ошибок – 3 балла

**Пример 3. Правильная программа на языке C++, эффективная только по времени**

```
#include <iostream>
using namespace std;

int main() {
    const int p = 17; // делитель
    int N; cin >> N; // количество элементов
    int a[N]; // элементы последовательности
    for (int i = 0; i < N; ++i) cin >> a[i];
    int imp0 = -1, imp1 = -1; //индексы максимумов, кратных p
    for (int i = 0; i < N; ++i) {
        if (a[i] % p == 0) {
            if (a[i] % 2 == 0) {
                if (imp0 == -1 || a[i] > a[imp0]) imp0 = i;
            }
            else {
                if (imp1 == -1 || a[i] > a[imp1]) imp1 = i;
            }
        }
    }
    int im0 = -1, im1 = -1; // индексы общих максимумов
    for (int i = 0; i < N; ++i) {
        if (i != imp0 && i != imp1) {
            if (a[i] % 2 == 0) {
                if (im0 == -1 || a[i] > a[im0]) im0 = i;
            }
            else {
                if (im1 == -1 || a[i] > a[im1]) im1 = i;
            }
        }
    }
    int x = 0, y = 0; // пара чисел для ответа
    if (imp0 != -1 && im0 != -1) {
        x = a[imp0]; y = a[im0];
    }
    if (imp1 != -1 && im1 != -1 && a[imp1] + a[im1] > x + y) {
        x = a[imp1]; y = a[im1];
    }
    cout << x << ' ' << y << endl;
    return 0;
}
```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и выберем подходящую. Такое решение не является эффективным ни по памяти (требуемая память зависит от размера исходных данных), ни по времени (количество возможных пар, а значит, количество действий и время счёта с ростом количества исходных элементов растут квадратично). Подобная программа оценивается не выше 2 баллов.

Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC)

**Пример 4. Правильная, но неэффективная программа на языке Паскаль**

```
const p = 17;
var
    N: integer;           {количество чисел}
    a: array [1..10000] of integer; {исходные данные}
    x1, x2: integer;     {ответ - пара чисел}
    i, j: integer;

begin
    readln(N);
    for i := 1 to N do readln(a[i]);
    x1 := 0; x2 := 0;
    for i := 1 to N - 1 do begin
        for j := i + 1 to N do begin
            if ((a[i] - a[j]) mod 2 = 0) and
                ((a[i] mod p = 0) or (a[j] mod p = 0)) and
                (a[i] + a[j] > x1 + x2)
            then begin
                x1 := a[i]; x2 := a[j]
            end
        end
    end;
    writeln(x1, ' ', x2)
end.
```

Указания по оцениванию	Баллы
Если в работе представлены две программы решения задачи, то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается бóльшая из двух оценок. Описание алгоритма решения без программы оценивается в 0 баллов	
Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел $N$ , время работы пропорционально этому количеству. Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов: 1) пропущен или неверно указан знак пунктуации; 2) неверно написано, пропущено или написано лишнее зарезервированное слово языка программирования; 3) не описана или неверно описана переменная; 4) применяется операция, недопустимая для соответствующего типа данных. Если одна и та же ошибка встречается несколько раз, 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 $N$ .	3

<p>Количество синтаксических ошибок («описок»), указанных в критериях на 4 балла, – не более пяти.</p> <p>Допускается наличие не более одной содержательной ошибки следующих видов:</p> <ol style="list-style-type: none"> <li>1) ошибка при вводе данных (не считается значение <math>N</math> или неверно организован ввод последовательности);</li> <li>2) ошибка при инициализации или отсутствие инициализации там, где она необходима;</li> <li>3) используется неверный тип данных;</li> <li>4) использована одна переменная (константа) вместо другой;</li> <li>5) используется один знак операции вместо другого;</li> <li>6) отсутствует вывод ответа, или выводится не то значение (например, выводится сумма вместо пары чисел);</li> <li>7) неверная работа с массивом, в том числе выход за границы массива;</li> <li>8) пропущены или неверно расставлены операторные скобки (при использовании языков с операторными скобками);</li> <li>9) учитываются пары из двух одинаковых чисел, если это число встречается в последовательности только один раз, или не учитываются пары из одинаковых чисел, встречающихся в последовательности более одного раза;</li> <li>10) используется строгое неравенство там, где необходимо нестрогое, или наоборот.</li> </ol> <p>3 балла также ставится за программу без содержательных ошибок, в которой используемая память зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве или другой аналогичной структуре данных)</p>	
<p>Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает в целом верно и эффективно по времени. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла.</p> <p style="text-align: center;">ИЛИ</p> <p>Представлено корректное переборное решение, в котором все исходные данные сохраняются в массиве (или другой аналогичной структуре) и рассматриваются все возможные пары. При этом не допускаются содержательные логические ошибки, например выход индексов за границы массива, рассмотрение сумм вида <math>a[i] + a[i]</math>.</p>	2

<p>Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа описывает в целом правильный алгоритм, содержит как минимум два элемента из следующего списка, возможно, реализованных с ошибками:</p> <ol style="list-style-type: none"> <li>1) рассматриваются пары с чётной разностью;</li> <li>2) проверяется делимость на <math>p</math>;</li> <li>3) рассматриваются пары с максимальной суммой</li> </ol>	1
<p>Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла</p>	0
<i>Максимальный балл</i>	4

**Ответы к заданиям**

№ задания	Ответ
1	9
2	wzyx
3	46
4	2
5	110
6	32
7	19
8	240
9	12
10	256
11	1421731
12	192
13	14
14	98
15	8
16	34
17	11
18	91
19	155
20	502
21	61
22	28
23	64

### Критерии оценивания заданий с развёрнутым ответом

- 24 Требовалось написать программу, которая получает на вход натуральное число  $N$ , не превосходящее  $10^9$ , и выводит число, равное количеству цифр 2 в десятичной записи числа  $N$ . Программист написал программу неправильно. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre> DIM N AS LONG DIM R, d AS INTEGER INPUT N R = 0 WHILE N &gt; 0   d = N MOD 10   IF d &lt;&gt; 2 THEN     R = R + 1   END IF   N = N \ 10 WEND PRINT d END </pre>	<pre> N = int(input()) R = 0 while N &gt; 0:     d = N % 10     if d != 2:         R = R + 1     N = N // 10 print(d) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач   цел N, R, d   ввод N   R := 0   нц пока N &gt; 0     d := mod(N, 10)     если d &lt;&gt; 2 то       R := R + 1     все     N := div(N, 10)   кц   вывод d кон </pre>	<pre> var N: longint;     R, d: integer; begin   readln(N);   R := 0;   while N &gt; 0 do begin     d := N mod 10;     if d &lt;&gt; 2 then       R := R + 1;     N := N div 10;   end;   writeln(d); end. </pre>

#### C++

```

#include <iostream>
using namespace std;

int main()
{
    long int N;
    int R, d;
    cin >> N;
    R = 0;
    while (N > 0) {
        d = N % 10;
        if (d != 2) {
            R = R + 1;
        }
        N = N / 10;
    }
    cout << d << endl;
    return 0;
}

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 324.
2. Приведите пример входного числа  $N$ , при котором приведённая программа, несмотря на ошибки, выдаёт верный ответ.
3. Найдите допущенные программистом ошибки и исправьте их. Исправление ошибки должно затрагивать только строку, в которой находится ошибка. Для каждой ошибки:
  - 1) выпишите строку, в которой сделана ошибка;
  - 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Известно, что в тексте программы нужно исправить не более двух строк так, чтобы она стала работать правильно.

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

Обратите внимание на то, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения.

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)	
<p>Решение использует запись программы на Паскале. Допускается использование программы на любом из четырёх других языков программирования.</p> <p>1. Программа выведет число 3. <i>Примечание для эксперта.</i> Программа выводит значение первой слева цифры числа <math>N</math>.</p> <p>2. Программа выдаёт правильный ответ, например, при <math>N = 3222</math>.</p> <p>3. В программе есть две ошибки. <b>Первая ошибка:</b> неверная проверка условия увеличения счетчика – переменной <math>R</math>. Строка с ошибкой: <code>if d &lt;&gt; 2 then</code> Верное исправление: <code>if d = 2 then</code></p> <p><b>Вторая ошибка:</b> вместо значения переменной <math>R</math> выводится значение переменной <math>d</math>. Строка с ошибкой: <code>writeln(d);</code> Верное исправление: <code>writeln(R);</code></p>	
Указания по оцениванию	Баллы
<p>Обратите внимание! В задаче требовалось выполнить <b>четыре</b> действия:</p> <ol style="list-style-type: none"> <li>1) указать, что выведет программа при конкретном входном числе;</li> <li>2) указать пример входного числа, при котором программа выдаёт верный ответ;</li> <li>3) исправить первую ошибку;</li> <li>4) исправить вторую ошибку.</li> </ol> <p>Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы.</p> <p>Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия:</p> <ol style="list-style-type: none"> <li>а) правильно указана строка с ошибкой;</li> <li>б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа</li> </ol>	

Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций: а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной; б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два из четырёх необходимых действий	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

25

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит сумму элементов массива, больших 100 и при этом не кратных 4, а затем заменяет каждый такой элемент на число, равное найденной сумме. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести изменённый массив, каждый элемент выводится с новой строки. Например, для исходного массива из шести элементов:

101

128

6

105

4

18

программа должна вывести следующий массив:

206

128

6

206

4

18

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

Бейсик	Python
<pre>CONST N AS INTEGER = 30 DIM A (1 TO N) AS LONG DIM I AS LONG,       J AS LONG,       K AS LONG  FOR I = 1 TO N   INPUT A(I) NEXT I ...</pre>	<pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 30 for i in range(0, n):   a.append(int(input())) ...</pre>
Алгоритмический язык	Паскаль
<pre>алг нач   цел N = 30   целтаб a[1:N]   цел i, j, k   нц для i от 1 до N     ввод a[i]   кц   ... кон</pre>	<pre>const   N = 30; var   a: array [1..N] of longint;   i, j, k: longint; begin   for i := 1 to N do     readln(a[i]);   ... end.</pre>
C++	
<pre>#include &lt;iostream&gt; using namespace std; const int N = 30; int main() {   long a[N];   long i, j, k;   for (i = 0; i &lt; N; i++)     cin &gt;&gt; a[i];   ...   return 0; }</pre>	

В качестве ответа Вам необходимо привести фрагмент программы, 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). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на Алгоритмическом языке).

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
<b>На языке Паскаль</b>
<pre>k := 0; for i := 1 to N do   if (a[i] &gt; 100) and (a[i] mod 4 &lt;&gt; 0) then     k := k + a[i]; for i := 1 to N do begin   if (a[i] &gt; 100) and (a[i] mod 4 &lt;&gt; 0) then     a[i] := k;   writeln(a[i]); end;</pre>
<b>На Алгоритмическом языке</b>
<pre>k := 0 нц для i от 1 до N   если a[i] &gt; 100 и mod(a[i], 4) &lt;&gt; 0   то     k := k + a[i]   все кц нц для i от 1 до N   если a[i] &gt; 100 и mod(a[i], 4) &lt;&gt; 0   то     a[i] := k   все вывод a[i], нс кц</pre>
<b>На языке Бейсик</b>
<pre>K = 0 FOR I = 1 TO N   IF A(I) &gt; 100 AND A(I) MOD 4 &lt;&gt; 0 THEN     K = K + A(I)   END IF NEXT I FOR I = 1 TO N   IF A(I) &gt; 100 AND A(I) MOD 4 &lt;&gt; 0 THEN     A(I) = K   END IF   PRINT A(I) NEXT I</pre>

На языке C++	
<pre> k = 0; for (i = 0; i &lt; N; i++)     if (a[i] &gt; 100 &amp;&amp; a[i] % 4 != 0)         k = k + a[i]; for (i = 0; i &lt; N; i++) {     if (a[i] &gt; 100 &amp;&amp; a[i] % 4 != 0)         a[i] = k;     cout &lt;&lt; a[i] &lt;&lt; endl; } </pre>	
На языке Python	
<pre> k = 0 for i in range(0, n):     if (a[i] &gt; 100 and a[i] % 4 != 0):         k = k + a[i] for i in range(0, n):     if (a[i] &gt; 100 and a[i] % 4 != 0):         a[i] = k print(a[i]) </pre>	
Указания по оцениванию	Баллы
<p><i>Общие указания.</i></p> <p>1. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы.</p> <p>2. Эффективность алгоритма не имеет значения и не оценивается.</p> <p>3. Допускается запись алгоритма на языке программирования, отличном от языков, приведённых в условии. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на Алгоритмическом 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.</p> <p>4. Допускается формат вывода массива, отличный от указанного, например в строчку (в том числе без разделителей)</p>	

Предложен правильный алгоритм, который изменяет исходный массив и выводит в качестве результата изменённый массив	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"> <li>1) в цикле происходит выход за границу массива;</li> <li>2) не инициализируется или неверно инициализируется сумма найденных элементов;</li> <li>3) неверно осуществляется проверка делимости на 4;</li> <li>4) проверяется делимость на 4 не элемента массива, а его индекса;</li> <li>5) неверно выбрана операция отношения для сравнения с границей диапазона;</li> <li>6) сравнение с границей диапазона производится для индекса элемента массива, а не для его значения;</li> <li>7) неверно составлено логическое условие (например, используется <code>or</code> вместо <code>and</code>);</li> <li>8) не вычисляется или неверно накапливается сумма найденных элементов;</li> <li>9) исходный массив не изменяется;</li> <li>10) отсутствует вывод ответа, или ответ выводится не полностью (например, только один элемент массива ввиду пропущенного цикла вывода элементов или операторных скобок);</li> <li>11) используется переменная, не объявленная в разделе описания переменных;</li> <li>12) не указано или неверно указано условие завершения цикла;</li> <li>13) индексная переменная в цикле не меняется (например, в цикле <code>while</code>) или меняется неверно</li> </ol>	1
Ошибок, перечисленных в п. 1–13, две или больше, или алгоритм сформулирован неверно (в том числе при отсутствии в явном или неявном виде цикла подсчёта суммы нужных элементов)	0
<i>Максимальный балл</i>	2

26

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 **четыре** камня либо увеличить количество камней в куче **в пять раз**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 19 или 75 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 68.



Победителем считается игрок, сделавший последний ход, т.е. первым получивший кучу, в которой будет 68 или больше камней.

В начальный момент в куче было  $S$  камней;  $1 \leq S \leq 67$ .

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. В описание выигрышной стратегии **не следует** включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

### Задание 1

- Укажите все такие значения числа  $S$ , при которых Петя может выиграть за один ход.
- 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.

### Задание 2

Укажите два таких значения  $S$ , при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для каждого указанного значения  $S$  опишите выигрышную стратегию Пети.

### Задание 3

Укажите значение  $S$ , при котором одновременно выполняются два условия:

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

Дерево не должно содержать партии, невозможные при реализации выигрывающим игроком своей выигрышной стратегии. Например, полное дерево игры не является верным ответом на это задание.

### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

#### Задание 1

- Петя может выиграть, если  $S = 14, \dots, 67$ .
- Ваня может выиграть первым ходом (как бы ни играл Петя), если исходно в куче  $S = 13$  камней. Тогда после первого хода Пети в куче будет 14, 17 или 65 камней. Во всех случаях Ваня увеличивает количество камней в 5 раз и выигрывает за один ход.

*Замечание для проверяющего.* В задаче не требуется указать **все** выигрышные стратегии. Если в работе выпускника, как в приведённом примере, просто указано, что Ваня всегда увеличивает в 5 раз количество камней, – это не ошибка.

#### Задание 2

Возможные значения  $S$ : 9, 12. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако он может получить кучу из 13 камней. Эта позиция разобрана в п. 1б. В ней игрок, который будет ходить (теперь это Ваня), выиграть не может, а его противник (т.е. Петя) следующим ходом выигрывает.

#### Задание 3

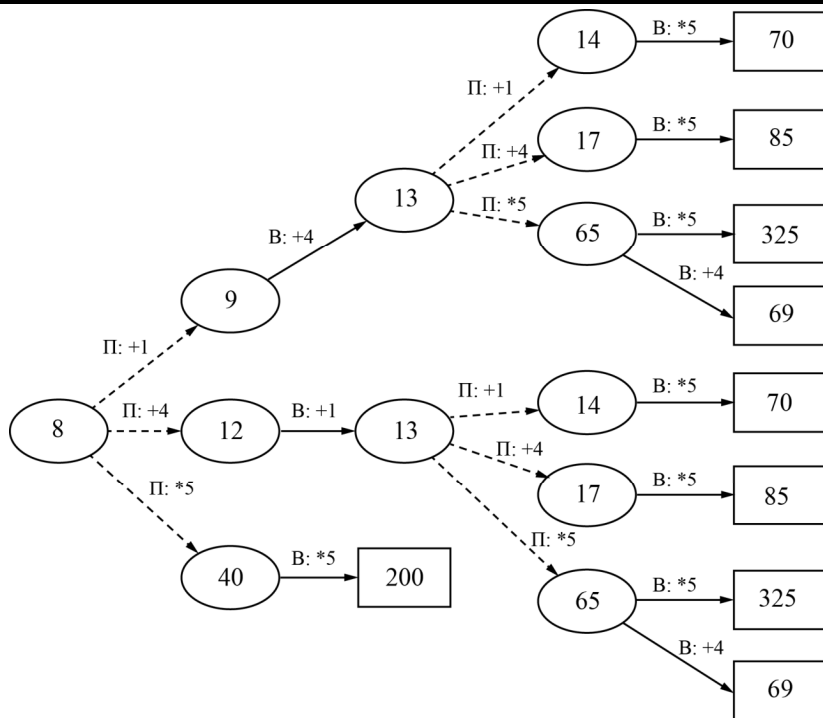
Возможные значения  $S$ : 8, 11.

Например, для  $S = 8$  после первого хода Пети в куче будет 9, 12 или 40 камней. Если в куче станет 40 камней, Ваня увеличит количество камней в 5 раз и выиграет первым ходом. Ситуация, когда в куче 9 или 12 камней, разобрана в п. 2. В этой ситуации игрок, который будет ходить (теперь это Ваня), выигрывает своим вторым ходом.

В таблице изображено дерево возможных партий (и только их) при описанной стратегии Вани для значения  $S = 8$ . При выбранной стратегии на последнем ходе Ваня увеличивает в 5 раз количество камней, хотя возможны и другие выигрышные стратегии. Для второго возможного значения дерево строится аналогично. Заключительные позиции (в них выигрывает Ваня) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

*Примечание для проверяющего.* Здесь для полноты картины указаны два возможных значения  $S$ . По условию задачи достаточно указать одно из них.

Исходное положение	Положения после очередных ходов			
	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)
8	8 + 1 = 9	9 + 4 = 13	13 + 1 = 14	14 * 5 = 70
			13 + 4 = 17	17 * 5 = 85
			13 * 5 = 65	65 * 5 = 325 65 + 4 = 69
	8 + 4 = 12	12 + 1 = 13	13 + 1 = 14	14 * 5 = 70
			13 + 4 = 17	17 * 5 = 85
			13 * 5 = 65	65 * 5 = 325 65 + 4 = 69
8 * 5 = 40	40 * 5 = 200			



Дерево всех партий, возможных при Ваниной стратегии. Прямоугольником обозначены позиции, в которых партия заканчивается

Указания по оцениванию	Баллы
<p><i>Замечание для проверяющего.</i> На рисунке для наглядности ходы Пети показаны пунктиром, а заключительные позиции выделены прямоугольником. Это не является обязательным для экзаменуемых. Также не является ошибкой указание только одного заключительного хода Вани в ситуации, когда у него есть два заключительных выигрышных хода</p> <p>В задаче от выпускника требуется выполнить три задания. Их трудность возрастает. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).</p> <p>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.</p> <p>Задание 1 выполнено, если выполнены оба пункта: а) и б). Если хотя бы один из этих пунктов не выполнен или выполнен с ошибкой (кроме оговоренных выше ошибок), задание считается невыполненным.</p> <p>Задание 2 выполнено, если правильно указаны обе позиции, выигрышные для Пети, и описана соответствующая стратегия Пети – так, как это написано в примере решения, или другим способом, например с помощью дерева всех возможных при выбранной стратегии Пети партий (и только их).</p> <p>Задание 3 выполнено, если правильно указана позиция, выигрышная для Вани, и построено дерево всех возможных при Ваниной стратегии партий (и только их). Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом</p>	
Выполнены задания 1, 2 и 3	3
Не выполнены условия, позволяющие поставить 3 балла, и выполнено одно из следующих условий. 1. Выполнено задание 3. 2. Выполнены задания 1 и 2	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено одно из следующих условий. 1. Выполнено задание 1. 2. Выполнено задание 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
<i>Максимальный балл</i>	3

27 Дана последовательность  $N$  целых положительных чисел. Рассматриваются все пары элементов последовательности, разность которых чётна, и в этих парах, по крайней мере, одно из чисел пары делится на 19. Порядок элементов в паре неважен. Среди всех таких пар нужно найти и вывести пару с максимальной суммой элементов. Если одинаковую максимальную сумму имеет несколько пар, можно вывести любую из них. Если подходящих пар в последовательности нет, нужно вывести два нуля.

#### Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел  $N$  ( $2 \leq N \leq 10\,000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10 000.

*Пример входных данных:*

5  
38  
12  
57  
16  
57

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

57 57

*Пояснение.* Из данных пяти чисел можно составить три различные пары, удовлетворяющие условию: (38, 12), (38, 16), (57, 57). Наибольшая сумма получается в паре (57, 57). Эта пара допустима, так как число 57 встречается в исходной последовательности дважды.

Напишите эффективную по времени и памяти программу для решения этой задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел  $N$  в  $k$  раз время работы программы увеличивается не более чем в  $k$  раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 Кбайт и не увеличивается с ростом  $N$ .

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени или только по памяти, – 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, – 2 балла.

Вы можете сдать **одну** или **две** программы решения задачи. Если Вы сдадите две программы, каждая из них будет оцениваться независимо от другой, итоговой станет **бóльшая** из двух оценок.

Перед текстом программы кратко опишите алгоритм решения. Укажите использованный язык программирования и его версию.

<b>Содержание верного ответа и указания по оцениванию</b> (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Разность двух чисел чётна, если эти числа имеют одинаковую чётность.</p> <p>Будем вводить элементы последовательности по одному и хранить четыре элемента: максимальные чётный <math>m_0</math> и нечётный <math>m_1</math> элементы всей последовательности и максимальные чётный и нечётный элементы, кратные <math>p = 19</math> (<math>m_{p0}</math> и <math>m_{p1}</math> соответственно). При этом у чисел, кратных <math>p</math>, будет «приоритет»: если вновь найденный общий максимум последовательности (с учётом чётности) кратен <math>p</math>, это число будет записано в качестве максимума среди кратных <math>p</math> (<math>m_{p0}</math> или <math>m_{p1}</math>). В общий максимум (<math>m_0</math> или <math>m_1</math>) оно сможет перейти, только если среди кратных <math>p</math> (<math>m_{p0}</math> или <math>m_{p1}</math>) появится большее или такое же значение.</p> <p>После ввода всех элементов нужно найти значения <math>m_{p0} + m_0</math>, <math>m_{p1} + m_1</math> и выбрать пару с большей суммой. При этом необходимо убедиться, что эти максимумы действительно выбраны из последовательности, а не записаны при инициализации нулевыми или отрицательными значениями.</p> <p>Ниже приведены программы на языках Pascal (использована версия PascalABC) и Алгоритмическом, реализующие этот алгоритм</p>

**Пример 1. Программа на языке Паскаль. Программа эффективна по времени и памяти**

```

const p = 19;
var
  N: integer;      {количество чисел}
  a: integer;      {очередное число}
  d: integer;      {чётность очередного числа}

  m: array [0..1] of integer; {чётный и нечётный максимумы}
  mp: array [0..1] of integer; {чётный и нечётный максимумы, кратные p}
  x, y: integer;    {ответ - пара чисел}
  i: integer;

begin
  m[0] := 0; m[1] := 0;
  mp[0] := 0; mp[1] := 0;
  x := 0; y := 0;
  readln(N);
  for i := 1 to N do
    begin
      readln(a);
      d := a mod 2;
      if (a mod p = 0) and (a >= mp[d]) then
        {нестрогое сравнение!}
        begin
          if mp[d] > m[d] then {допустимо нестрогое сравнение}
            m[d] := mp[d];
          mp[d] := a
        end
      else if a > m[d] then {допустимо нестрогое сравнение}
        m[d] := a
      end;
    if (mp[0] > 0) and (m[0] > 0) then
      x := mp[0]; y := m[0];
    if (mp[1] > 0) and (m[1] > 0) and (mp[1] + m[1] > x + y) then
      x := mp[1]; y := m[1];
    writeln(x, ' ', y)
  end.

```

**Пример 1а. Программа на Алгоритмическом языке. Программа эффективна по времени и памяти**

```

алг задача
нач
  цел p = 19 | делитель из условия задачи
  цел N | количество чисел
  цел a | очередное число
  цел d | чётность очередного числа
  целтаб m[0:1] | чётный и нечётный максимумы
  целтаб mp[0:1] | чётный и нечётный максимумы, кратные p
  цел x, y | пара чисел для ответа

  m[0] := 0; m[1] := 0
  mp[0] := 0; mp[1] := 0
  ввод N
  нц N раз
    ввод a
    d := mod(a, 2)
    если mod(a, p) = 0 и a >= mp[d] | нужно нестрогое сравнение!
      то если mp[d] > m[d] | допустимо нестрогое сравнение
        то m[d] := mp[d]
      все
        mp[d] := a
    иначе если a > m[d] | допустимо нестрогое сравнение
      то m[d] := a
    все
  все
кц
  x := 0; y := 0
  если mp[0] > 0 и m[0] > 0
    то x := mp[0]; y := m[0]
  все
  если mp[1] > 0 и m[1] > 0 и mp[1] + m[1] > x + y
    то x := mp[1]; y := m[1]
  все
  вывод x, ' ', y
кон

```

В приведённом решении вместо массивов  $m$  и  $mp$  можно использовать две пары простых переменных. В этом случае программа получится более громоздкой (придётся отдельно разбирать случаи чётного и нечётного чисел в последовательности и дублировать аналогичные фрагменты программы), но останется эффективной. Ниже приводится реализующая такой алгоритм программа на языке Python 3

**Пример 2. Программа на языке Python 3. Программа эффективна по времени и памяти**

```

p = 19
m0 = m1 = mp0 = mp1 = 0
N = int(input())
for i in range(N):
    a = int(input())
    if a % 2 == 0:
        if a % p == 0 and a >= mp0:
            if mp0 > m0: m0 = mp0
            mp0 = a
        elif a > m0: m0 = a
    else:
        if a % p == 0 and a >= mp1:
            if mp1 > m1: m1 = mp1
            mp1 = a
        elif a > m1: m1 = a
x = y = 0
if mp0 > 0 and m0 > 0:
    x = mp0; y = m0
if mp1 > 0 and m1 > 0 and mp1 + m1 > x + y:
    x = mp1; y = m1
print(x, y)

```

Ещё один путь решения – записать всю последовательность в массив и анализировать её в несколько проходов. Ниже приводится реализующая такой алгоритм программа на языке C++. В этой программе массив с исходными данными обрабатывается два раза: на первом проходе находятся индексы максимального чётного и нечётного элементов, кратных  $p$ , на втором проходе – общие чётный и нечётный максимумы. При этом элементы, выделенные как кратные при первом проходе, во время второго прохода из сравнения исключаются. Такая программа эффективна по времени (несмотря на повторную обработку массива, общее время работы пропорционально  $N$ ), но неэффективна по памяти. Максимальная оценка за такую программу при отсутствии в ней синтаксических и содержательных ошибок – 3 балла

**Пример 3. Правильная программа на языке C++, эффективная только по времени**

```

#include <iostream>
using namespace std;

int main() {
    const int p = 19; // делитель
    int N; cin >> N; // количество элементов
    int a[N]; // элементы последовательности
    for (int i = 0; i < N; ++i) cin >> a[i];
    int imp0 = -1, imp1 = -1; //индексы максимумов, кратных p
    for (int i = 0; i < N; ++i) {
        if (a[i] % p == 0) {
            if (a[i] % 2 == 0) {
                if (imp0 == -1 || a[i] > a[imp0]) imp0 = i;
            }
            else {
                if (imp1 == -1 || a[i] > a[imp1]) imp1 = i;
            }
        }
    }
    int im0 = -1, im1 = -1; // индексы общих максимумов
    for (int i = 0; i < N; ++i) {
        if (i != imp0 && i != imp1) {
            if (a[i] % 2 == 0) {
                if (im0 == -1 || a[i] > a[im0]) im0 = i;
            }
            else {
                if (im1 == -1 || a[i] > a[im1]) im1 = i;
            }
        }
    }
    int x = 0, y = 0; // пара чисел для ответа
    if (imp0 != -1 && im0 != -1) {
        x = a[imp0]; y = a[im0];
    }
    if (imp1 != -1 && im1 != -1 && a[imp1] + a[im1] > x + y) {
        x = a[imp1]; y = a[im1];
    }
    cout << x << ' ' << y << endl;
    return 0;
}

```

Возможно также «лобовое» решение: запишем все исходные числа в массив, переберём все возможные пары и выберем подходящую. Такое решение не является эффективным ни по памяти (требуемая память зависит от размера исходных данных), ни по времени (количество возможных пар, а значит, количество действий и время счёта с ростом количества исходных элементов растут квадратично). Подобная программа оценивается не выше 2 баллов.

Ниже приведена реализующая описанный выше алгоритм программа на языке Паскаль (использована версия PascalABC)

**Пример 4. Правильная, но неэффективная программа на языке Паскаль**

```

const p = 19;
var
  N: integer;      {количество чисел}
  a: array [1..10000] of integer; {исходные данные}
  x1, x2: integer; {ответ - пара чисел}
  i, j: integer;

begin
  readln(N);
  for i := 1 to N do readln(a[i]);
  x1 := 0; x2 := 0;
  for i := 1 to N - 1 do begin
    for j := i + 1 to N do begin
      if ((a[i] - a[j]) mod 2 = 0) and
        ((a[i] mod p = 0) or (a[j] mod p = 0)) and
        (a[i] + a[j] > x1 + x2)
      then begin
        x1 := a[i]; x2 := a[j]
      end
    end
  end
  writeln(x1, ' ', x2)
end.

```

Указания по оцениванию	Баллы
Если в работе представлены две программы решения задачи, то каждая из них независимо оценивается по указанным ниже критериям, итоговой считается бóльшая из двух оценок. Описание алгоритма решения без программы оценивается в 0 баллов	
Программа правильно работает для любых входных данных произвольного размера. Используемая память не зависит от количества прочитанных чисел $N$ , время работы пропорционально этому количеству. Допускается наличие в тексте программы до трёх синтаксических ошибок одного из следующих видов: 1) пропущен или неверно указан знак пунктуации; 2) неверно написано, пропущено или написано лишнее зарезервированное слово языка программирования; 3) не описана или неверно описана переменная; 4) применяется операция, недопустимая для соответствующего типа данных. Если одна и та же ошибка встречается несколько раз, 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 $N$ .	3

Количество синтаксических ошибок («описок»), указанных в критериях на 4 балла, – не более пяти.  
Допускается наличие не более одной содержательной ошибки следующих видов:

- 1) ошибка при вводе данных (не считывается значение  $N$  или неверно организован ввод последовательности);
  - 2) ошибка при инициализации или отсутствие инициализации там, где она необходима;
  - 3) используется неверный тип данных;
  - 4) использована одна переменная (константа) вместо другой;
  - 5) используется один знак операции вместо другого;
  - 6) отсутствует вывод ответа, или выводится не то значение (например, выводится сумма вместо пары чисел);
  - 7) неверная работа с массивом, в том числе выход за границы массива;
  - 8) пропущены или неверно расставлены операторные скобки (при использовании языков с операторными скобками);
  - 9) учитываются пары из двух одинаковых чисел, если это число встречается в последовательности только один раз, или не учитываются пары из одинаковых чисел, встречающихся в последовательности более одного раза;
  - 10) используется строгое неравенство там, где необходимо нестрогое, или наоборот.
- 3 балла также ставится за программу без содержательных ошибок, в которой используемая память зависит от количества прочитанных чисел (например, входные данные запоминаются в массиве или другой аналогичной структуре данных).

Не выполнены условия, позволяющие поставить 3 или 4 балла, при этом программа работает в целом верно и эффективно по времени. Допускается наличие до трёх содержательных ошибок, описанных в критериях на 3 балла, и до девяти синтаксических ошибок, описанных в критериях на 4 балла.

ИЛИ

Представлено корректное переборное решение, в котором все исходные данные сохраняются в массиве (или другой аналогичной структуре) и рассматриваются все возможные пары. При этом не допускаются содержательные логические ошибки, например выход индексов за границы массива, рассмотрение сумм вида  $a[i] + a[i]$ .

2

Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. При этом программа описывает в целом правильный алгоритм, содержит как минимум два элемента из следующего списка, возможно, реализованных с ошибками: 1) рассматриваются пары с чётной разностью; 2) проверяется делимость на $p$ ; 3) рассматриваются пары с максимальной суммой	1
Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла	0
<i>Максимальный балл</i>	<i>4</i>