

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 89, май 2024 г.

011000100010101001001111100100011001001000101001001001011

ОГЭ

О заданиях 7 ОГЭ по информатике

1. Общие вопросы

В Спецификации контрольных измерительных материалов ОГЭ по информатике в качестве проверяемого результата обучения применительно к заданию 7 указывается «Знание принципов адресации в сети Интернет».

Тема в федеральном компоненте ГОС ООО – «Сохранение информационных объектов из компьютерных сетей и ссылок на них для индивидуального использования (в том числе из Интернета)».

Уровень сложности – базовый.

Максимальный балл за задание – 1.

Примерное время выполнения задания (мин.) – 3.

В демонстрационных вариантах контрольных измерительных материалов ОГЭ по информатике нескольких последних лет и в открытом банке заданий ОГЭ представлены задания, общий вид которых такой:

«Доступ к файлу *<приводится имя файла>*, находящемуся на сервере¹ *<приводится имя сервера>*, осуществляется по протоколу *<приводится имя (тип) протокола>*. Фрагменты адреса файла закодированы цифрами от 1 до 7 (*<приводится фрагменты адреса с номерами>*). Запишите в ответе последовательность этих цифр, кодирующую адрес указанного файла в сети Интернет».

Пример

Доступ к файлу rus.doc, находящемуся на сервере obr.org, осуществляется по протоколу https. Фрагменты адреса файла закодированы цифрами от 1 до 7. Запишите в ответе последовательность этих цифр, кодирующую адрес указанного файла в сети Интернет.

- 1) obr.
- 2) /
- 3) org
- 4) ://
- 5) doc
- 6) rus.
- 7) https

¹ Сервер – мощный компьютер, постоянно подключенный к сети Интернет, программное обеспечение предназначено для обслуживания других компьютеров, подключенных к сети.

2. Теоретические сведения и основные понятия

Файл – это область в памяти компьютера (на диске), в которой записана некоторая информация и которая имеет имя.

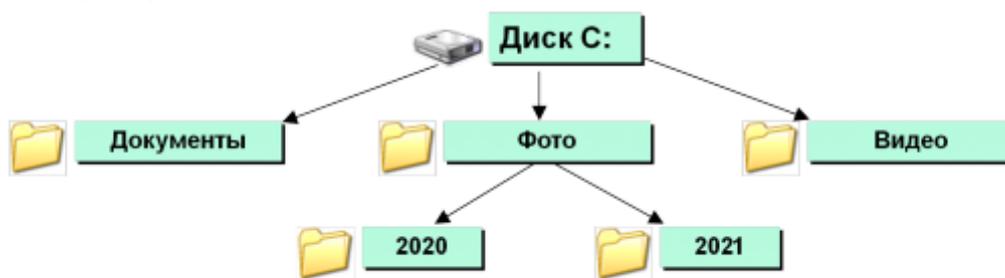
Имя файла состоит из двух частей – собственно имени и расширения, записанных через точку:

имя.расширение

Как правило, имя файлу дает его создатель, т. е. человек, который создал этот файл в определенной программе. И, как правило, имя носит содержательный характер, что позволяет понять, что именно записано в этом файле.

Расширение файла определяет, к какому типу файлов он относится или, другими словами, какая информация в нем хранится: например, текстовый файл, файл-изображение, файл электронной таблицы, видеофайл и т. п. Расширение файла либо задается программой автоматически, либо выбирается пользователем из некоторого списка, либо задается им.

Так как на диске хранится множество файлов, то для удобства поиска того или иного файла их объединяют в так называемые «*каталоги*» (или «*папки*»). Внутри каталогов (папок) с большим количеством файлов могут находиться другие каталоги (папки), таким образом, организуется сложная структура каталогов – иерархическая структура, называемая «*деревом каталогов*». Пример:



В дальнейшем мы будем использовать термин «каталог», который употребляется в заданиях демонстрационных вариантов ОГЭ.

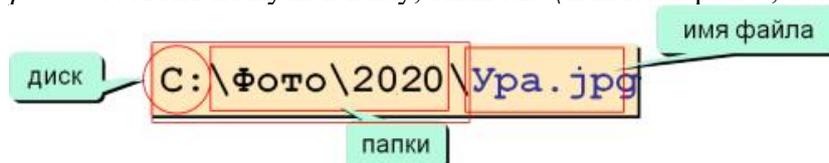
Главный каталог диска, в котором находятся все другие каталоги и отдельные файлы, называется «*корневым каталогом*» и обозначается буквой логического диска, за которой следует двоеточие и знак «\» (обратный слэш); например, C:\ – это обозначение корневого каталога диска C.

Ещё два понятия.

Путь к файлу – это перечисление всех каталогов, в которые нужно войти (открыть), чтобы попасть в каталог с данным файлом (начиная с корневого каталога диска); например, если файл находится в каталоге Информатика, то путь к нему такой:

C:\user\Митя\Школа\Информатика

Полное имя файла состоит из пути к нему, символа \ и имени файла, например:



Все сказанное выше относилось к каталогам и файлам, находящимся на дисках настольных компьютеров, ноутбуков, планшетов и т. п. Адресация файлов, расположенных на серверах в Интернете, имеет особенности.

Адрес файла на сервере в Интернете имеет такой общий вид (пробелы записаны для наглядности²):

протокол :// имя_сервера / полное_имя_файла

где протокол – набор правил, по которым нужно обращаться к файлу, имя_сервера – так называемое «доменное имя» компьютера-сервера, на котором расположен файл, полное_имя_файла – имя файла с указанием всех каталогов, в которые нужно войти, чтобы попасть в каталог с данным файлом.

Пример адреса файла index.html, расположенного в каталоге business на сервере www.comp.ru, к которому нужно обращаться по протоколу передачи файлов FTP:

ftp://www.comp.ru/business/index.html

Некоторые другие возможные протоколы: http, https, telnet.

В простейшем случае, когда файл находится в корневом каталоге сервера, его адрес имеет вид (также с условными пробелами):

протокол :// имя_сервера / имя_файла

Пример адреса файла inf.doc, записанного в корневом каталоге сервера obr.org:

https://obr.org/inf.doc

3. Выполнение задания из примера

Выполним задание из примера, приведённого в пункте 1.

Согласно правилам записи адресов файлов на серверах в Интернете:

- 1) сначала записывается протокол: http (номер фрагмента 7);
- 2) затем идет обязательный знак: :// (4);
- 3) потом имя сервера: obr.org (1 и 3);
- 4) затем опять знак: / (2);
- 5) последним идет имя файла: rus.doc (6 и 5).

Ответ: 7413265. (На экзамене точку приводить не следует.)

4. Задания для самостоятельной работы

1. Доступ к файлу muzeum.htm, находящемуся на сервере mart.ru, осуществляется по протоколу http. Фрагменты адреса файла закодированы цифрами от 1 до 7. Запишите последовательность этих цифр, кодирующую адрес указанного файла в сети Интернет.

- 1) muzeum
- 2) /
- 3) mart
- 4) .htm
- 5) .ru
- 6) ://
- 7) http

2. На сервере net.ru имеется файл org.txt, доступ к которому осуществляется по протоколу http. Фрагменты адреса файла закодированы цифрами от 1 до 7. Запишите последовательность этих цифр, кодирующую адрес указанного файла в сети Интернет.

- 1) org
- 2) .ru
- 3) /
- 4) ://
- 5) http
- 6) net
- 7) .txt

² Обратите внимание на вид символов «/».

3. Доступ к файлу tasks.rar, находящемуся на сервере contest.ru, осуществляется по протоколу http. Фрагменты адреса файла закодированы цифрами от 1 до 7. Запишите последовательность этих цифр, кодирующую адрес указанного файла в сети Интернет.

- 1) tasks
- 2) .rar
- 3) ://
- 4) contest
- 5) http
- 6) .ru
- 7) /

4 (Задание из демонстрационного варианта ОГЭ 2024 года). Доступ к файлу edu.txt, находящемуся на сервере htm.ru, осуществляется по протоколу http. Фрагменты адреса файла закодированы цифрами от 1 до 7. Запишите последовательность этих цифр, кодирующую адрес указанного файла в сети Интернет.

- 1) htm
- 2) ://
- 3) /
- 4) http
- 5) .ru
- 6) .txt
- 7) .edu

О заданиях 15 ОГЭ по информатике (вариант 15.2)

1. Общие вопросы

В Спецификации контрольных измерительных материалов ОГЭ в качестве проверяемого результата обучения применительно к варианту 15.2 задания 15 указывается «Умение создавать и выполнять программы на универсальном языке программирования».

Темы в федеральном компоненте ГОС ООО – «Алгоритм, свойства алгоритмов, способы записи алгоритмов. Блок-схемы. Представление о программировании. Алгоритмические конструкции. Логические значения, операции, выражения. Разбиение задачи на подзадачи, вспомогательный алгоритм».

Уровень сложности – высокий.

Максимальный балл за задание – 2.

Примерное время выполнения задания (мин.) – 45.

В демонстрационных вариантах контрольных измерительных материалов ОГЭ по информатике нескольких последних лет и в открытом банке заданий ОГЭ представлены задания на разработку программ, связанных с определением количественных характеристик заданной последовательности натуральных чисел (нахождение их суммы, среднего арифметического, количества чисел, удовлетворяющих некоторому условию, и др.). В общем виде задания имеют вид:

«Напишите программу, которая в последовательности натуральных чисел определяет *<указывается, что следует определить>*. Программа получает на вход количество чисел в последовательности, а затем сами числа. Количество чисел не превышает *<указывается значение>*. Введённые числа не превышают *<указывается значение>* (или «Введённые числа по модулю не превышают *<указывается значение>*»). Программа должна вывести одно число – *<указывается значение-результат>*.

Пример работы программы:

Входные данные	Выходные данные
4	2
16	
7	
26	
77	

Возможен также вариант задания, в котором количество введённых чисел неизвестно, а известно, что последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Пример работы программы решения такой задачи:

Входные данные	Выходные данные
36	2
91	
40	
0	

2. Типовые задачи и методика их решения

Опишем методику разработки программ решения ряда типовых задач с указанными особенностями. Начнем с варианта, в котором количество чисел последовательности известно (задается первым).

Для каждой рассмотренной задачи приведён фрагмент программы её решения на школьном алгоритмическом языке.

Использованы следующие основные величины:

N – общее количество чисел в последовательности;

a – очередное число последовательности;

nom – порядковый номер очередного числа последовательности.

Смысл остальных величин можно легко определить по их именам.

При решении задач можно выделить четыре основных этапа:

1. Ввод количества N чисел в последовательности:

```
ввод N
```

2. Принятие начальных значений переменных, используемых при расчётах.

3. Ввод и обработка каждого отдельного числа последовательности. Так как количество обрабатываемых чисел N на этом этапе уже известно, то целесообразно использовать оператор цикла с параметром:

```
нц для ном от 1 до N  
  ввод а  
  | Обработка очередного числа  
кц
```

4. Вывод ответа

2.1. Типовая задача 1 «Суммирование всех чисел последовательности»

Пусть требуется найти сумму всех 10 чисел последовательности: 5, 3, 1, 5, 12, 6, 3, 3, 8, 12.

Используем в программе переменную $сум$, которая будет хранить сумму чисел среди уже введенных чисел.

Составим таблицу:

Порядковый номер числа	Значение	Сумма
1	5	5
2	3	8
...		
7		сум
8	a	?

Как определить сумму чисел после ввода некоторого очередного числа n , если информацию о такой сумме среди уже рассмотренных чисел хранит величина $сум$? Ответ – надо к «старому» значению суммы $сум$ прибавить n :

```
сум := сум + n
```

Такая запись, невозможная в математике, в программировании возможна. С учетом этого фрагмент программы, связанный с этапом 2, имеет вид:

```
сум := 0
нц для ном от 1 до N
  ввод а
  сум := сум + а
кц
```

Внимание! В программирования Бейсик и Паскаль начальное присваивание переменной нулевого значения в данном случае не является обязательным. Вместе с тем, начальное присваивание величине нулевого значения в программировании считается «правилом хорошего тона». В программах на школьном алгоритмическом языке, языке Python и ряде других такое присваивание обязательно.

2.2. Типовая задача 2 «Суммирование чисел последовательности, которые обладают некоторыми свойствами (удовлетворяют некоторому условию)»

Пусть требуется найти сумму чётных чисел в последовательности: 5, 3, 1, 5, 12, 6, 3, 3, 8, 12.

Отличие задач данного типа от рассмотренных в предыдущем разделе в том, что учитывать в значении переменной $сум$ надо не все числа последовательности, а только чётные. Значит, должен быть использован неполный условный оператор (команда **если**):

```
ввод N
сум := 0
нц для ном от 1 до N
  ввод а
  если <число а - чётное>
    то
      сум := сум + а
  все
кц
```

Как определить, что целое число a является чётным, решите самостоятельно или читайте комментарий перед заданиями для самостоятельной работы.

В общем случае вид программы такой:

```
ввод N
сум := 0
нц для ном от 1 до N
  ввод а
  если <условие>
```

```

то
    сум := сум + а
все
кц
| Вывод результата
вывод нс, сум

```

Прежде чем идти дальше, заметим, что условие для учета очередного числа *a* может быть сложным – состоящим из двух или нескольких простых условий³, соединенных служебными словами (логическими операторами) **and** (**и**), **or** (**или**) или **not** (**не**).

Результат проверки сложного условия определяется по известным читателю таблицам истинности. Для сложного условия с двумя простыми условиями *У1* и *У2* эти таблицы приведены ниже.

1) **и (and)**

У1	У2	У1 и У2
Ложь	Ложь	Ложь
Ложь	Истина	Ложь
Истина	Ложь	Ложь
Истина	Истина	Истина

2) **или (or)**

Таблица 3.3

У1	У2	У1 или У2
Ложь	Ложь	Ложь
Ложь	Истина	Истина
Истина	Ложь	Истина
Истина	Истина	Истина

Результат выполнения логической операции отрицания (**not/не**) над условием *У* определяется так:

У	не У
Ложь	Истина
Истина	Ложь

2.3. Типовая задача 3 «Подсчет количества чисел последовательности, которые обладают некоторыми свойствами»

Пусть требуется найти количество чётных чисел последовательности: 5, 2, 7, 8, 3, 1, 5, 12, 6, 3, 3, 8, 12.

Используем в программе переменную *кол*, которая будет хранить количество чётных чисел среди уже введенных чисел.

Составим таблицу:

Порядковый номер числа	Значение	Количество чётных чисел
1	5	0
2	2	1
...		
...		<i>кол</i>
8	12	?

³ «Простым» условимся называть условие, к котором используется один из шести знаков сравнения («=», «>», «>=» и т. д.).

Как определить количество чётных чисел после ввода некоторого очередного чётного числа a , если информацию о таком количестве среди уже рассмотренных чисел хранит величина $кол_чет$? Ответ – надо к «старому» значению количества $кол$ прибавить 1:

```
кол_чет := кол_чет + 1
```

Напомним, что в программировании такая запись также возможна. Это значит, что фрагмент программы для подсчета искомого количества такой:

```
кол_чет := 0
нц для ном от 1 до к
  ввод а
  если <число а - чётное>
    то
      кол_чет := кол_чет + 1
  все
кц
```

В общем случае соответствующая программа:

```
ввод N
кол := 0
нц для ном от 1 до N
  ввод а
  если <условие>
    то
      кол := кол + 1
  все
кц
вывод нс, кол
```

2.4. Типовая задача 4 «Определение среднего арифметического тех чисел последовательности, которые обладают некоторыми свойствами»

Для нахождения среднего арифметического чисел последовательности, которые удовлетворяют некоторому условию (например, положительных, чётных или т. п.), надо знать сумму и количество таких чисел. Решать соответствующие задачи мы уже умеем. Поэтому можем так оформить программу:

```
ввод N
сум := 0
кол := 0
нц для ном от 1 до N
  ввод а
  если <условие>
    то
      сум := сум + а
      кол := кол + 1
  все
кц
сред_арифм := сум/кол
вывод нс, сред_арифм
```

Обращаем внимание на то, что многократно определять значение $сред_арифм$ в «теле» условного оператора:

```
нц для ном от 1 до к
  ввод а
  если <условие>
    то
      сум := сум + а
```

```

    кол := кол + 1
    | Определяем результат
    сред_арифм := сум/кол
все
кц
| Выводим окончательный результат
вывод нс, сред_арифм

```

необходимости нет. Это можно сделать один раз после окончания работы оператора цикла.

Обратим внимание на то, что приведенная программа будет работать без ошибок, если числа, удовлетворяющие заданному условию, в последовательности имеются (иначе будет иметь место деление на ноль). Именно такое уточнение, как правило, дается в демонстрационных вариантах экзамена.

2.5. Типовая задача 5 «Определение максимального значения в последовательности чисел»

Сразу же уточним, что в демонстрационных вариантах экзамена представлены задания, в которых речь идет о последовательности натуральных чисел.

Алгоритм решения задач такого типа аналогичен алгоритму действий человека, который определяет максимальное значение в некотором наборе:

4, 2, 9, 6, 3, 16, 10, 2, 7

– сначала он запоминает первое число, а затем рассматривает второе число. Если оно больше того числа, которое помнил, то запоминает новое число и переходит к следующему, третьему числу; в противном случае просто переходит к следующему числу и делает то же самое. Иными словами, каждое число последовательности, начиная со 2-го, сравнивается с максимальным значением среди чисел, рассмотренных до него.

Это значит, что в программе следует использовать переменную величину, которая будет хранить⁴ максимальное число из уже рассмотренных чисел последовательности. Если принять имя этой переменной – макс, то сначала следует задать первое число и принять его в качестве максимального (макс). Затем задаются остальные числа a и каждое из них сравнивается со значением макс. Если $a > \text{макс}$, то в качестве нового значения макс принимается значение числа a .

Соответствующий фрагмент:

```

| Ввод первого числа a
ввод a
макс := a
нц для ном от 2 до N
    ввод a
    | Сравниваем
    если a > макс
        то
            макс := a
    все
кц
| Вывод результата или использование его в расчётах
...

```

⁴ Напомним, что переменные величины используются в программе для хранения информации.

Обсудим вопрос – можно ли не вводить отдельно первое число последовательности, а вводить и обрабатывать *все* его числа в теле оператора цикла:

```
нц для ном от 1 до N
  | Ввод очередного числа а последовательности
  ввод а
  если а > макс
    то
      макс := а
  все
кц
вывод нс, макс
```

На первый взгляд, да, можно. Однако это не так, точнее, так можно оформлять программу не на всех языках программирования и не во всех случаях.

В программе на школьном алгоритмическом языке и на языке Python при ее выполнении появится сообщение об ошибке, связанное с тем, что значение макс при первом использовании условного оператора не определено (в программе на языке Паскаль это значение будет равно нулю).

А если до оператора цикла присвоить величине макс нулевое значение?

```
макс := 0
```

В этом случае сообщения об ошибке не будет, но если, например, стоит задача определения максимальной температуры в первой декаде января и вводимые значения равны -6, -5, -4, -6, -3, -2, -5, -5, -4, то в приведенном варианте программы результат окажется неправильным (он будет равен 0⁵).

Можно сделать вывод о том, что краткий вариант программы решения задачи о максимальной температуре в первой декаде января может быть применен, если известно, что в обрабатываемом наборе чисел не все значения температуры отрицательны:

```
макс := 0
нц для ном от 1 до 10
  вывод нс, "Введите температуру очередного дня "
  ввод темп
  если темп > макс
    то
      макс := темп
  все
кц
вывод нс, "Максимальная температура за декаду: ", макс, " градусов"
```

В общем случае краткий вариант программы может быть использован, если известно число X, которое заведомо не превышает минимальное число обрабатываемой последовательности:

```
макс := X
нц для ном от 1 до N
  ввод а
  если а > макс
    то
      макс := а
  все
кц
вывод нс, макс
```

⁵ Убедитесь в этом.

В заданиях экзамена речь идет об определении максимального *натурального* числа. Это значит, что в качестве начального значения переменной *макс* можно принимать значение 0, – 1,

2.6. Типовая задача 6 «Определение порядкового номера максимального значения в последовательности чисел»

Здесь также алгоритм решения задачи аналогичен алгоритму действий человека, определяющего номер максимального значения в некотором наборе:

4, 9, 2, 6, 3, 16, 10, 2, 7

– сначала он запоминает первое число и номер 1, а затем рассматривает второе число. Если оно больше того числа, которое помнил, то запоминает новое число и номер 2 и переходит к следующему, третьему числу; в противном случае просто переходит к следующему числу и делает то же самое и т. д. Иными словами, каждое число последовательности, начиная со второго, сравнивается с максимальным числом среди рассмотренных до него.

В программе сначала следует задать первое число и принять его в качестве максимального (*макс*), а искомый порядковый номер (*номер_макс*) принять равным 1. Затем задаются остальные числа *a* и каждое из них сравнивается со значением *макс*. Если $a > \text{макс}$, то в качестве нового значения *макс* принимается значение числа *a*, а в качестве нового значения *номер_макс* – номер встреченного числа *ном*.

Соответствующая программа:

```
| Ввод первого числа a
ВВОД a
макс := a
номер_макс := 1
НЦ ДЛЯ ном ОТ 2 ДО N
  | Ввод остальных чисел a последовательности
  ВВОД a
  | Сравниваем
  ЕСЛИ a > макс
    ТО
      макс := a
      номер_макс := ном
  ВСЕ
КЦ
ВЫВОД нс, номер_макс
```

Если заведомо известно число *X*, которое заведомо не превышает минимальное число обрабатываемой последовательности известно, то, как и в предыдущей задаче, фрагмент может быть оформлен короче:

```
макс := X
НЦ ДЛЯ ном ОТ 1 ДО k
  | Ввод очередного числа a последовательности
  ВВОД a
  ЕСЛИ a > макс
    ТО
      макс := a
      номер_макс := ном
  ВСЕ
КЦ
ВЫВОД нс, номер_макс
```

При определении номера максимального *натурального* числа последовательности в качестве начального значения переменной *макс* можно принимать значение 0, -1, ..., а начальное значение переменной *номер_макс* – вообще не задавать.

2.7. Типовая задача 7 «Определение максимального значения с заданными свойствами в последовательности чисел»

Пример: определение максимального чётного числа, максимального отрицательного числа и т. п.

Отличие задач данного типа от рассмотренных типовой задачи 5 в том, что учитывать при поиске максимального числа надо не все числа последовательности, а только числа с заданными свойствами. Значит, как и при решении типовой задачи 2 должен быть использован неполный условный оператор (команда **если**).

Краткий вариант программы, который напомним, применим к обработке натуральных чисел, имеет вид:

```
макс := 0
нц для ном от 1 до N
  ввод а
  если <число а обладает заданными свойствами>
  то
    | Сравниваем
    если а > макс
      то
        макс := а
    все
  все
кц
вывод нс, макс
```

2.8. Типовая задача 8 «Определение порядкового номера максимального значения с заданными свойствами в последовательности чисел»

Соответствующий фрагмент программы решения задачи:

```
макс := 0
нц для ном от 1 до N
  ввод а
  если <число а обладает заданными свойствами>
  то
    | Сравниваем
    если а > макс
      то
        макс := а
        номер_макс := ном
    все
  все
кц
вывод нс, номер_макс
```

Вопрос: «Номер какого числа с заданными свойствами будет найден, если в последовательности будет несколько таких чисел?»

Прежде чем предлагать задания для самостоятельной работы, дадим ещё ряд комментариев.

В заданиях экзамена могут быть указаны максимальные значения переменных величин, например, «количество чисел не превышает 1000, введённые числа последовательности не превышают 30 000». Поэтому в программе на языке Паскаль и других языках

программирования, в которых происходит объявление (описание) переменных, для соответствующих переменных должен быть указан соответствующий тип данных. Например, в программе на языке Паскаль для переменных с приведёнными значениями 1000 и 30000 это может тип *integer*.

Для проверки того факта, что некоторое целое число a является чётным следует определить остаток от его деления на 2. Если остаток равен нулю, то число a – чётное, в противном случае – нечётное. Знаки операции определения остатка показаны в таблице:

Язык программирования	Определение остатка	Пример выражения для определения остатка
Паскаль	mod	$a \bmod 2$
Python	%	$a \% 2$

В школьном алгоритмическом языке для определения остатка от деления одной величины целого типа на другую используется не знак операции, а функция mod:

`mod(a, 1000)`

Последнюю цифру любого натурального числа a можно определить, как остаток от деления этого числа на 10.

Ответы на следующие вопросы найдите самостоятельно:

- 1) как определить, что некоторое натуральное число a кратно числу b ?
- 2) как определить число, образованное двумя последними цифрами некоторого натурального числа a ?

В качестве примера выполним задание из демонстрационного варианта ОГЭ 2022 года.

Условие

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 4, но не кратных 7. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 4 и не кратное 7. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число: количество чисел, кратных 4, но не кратных 7.

Пример работы программы:

Входные данные	Выходные данные
4	2
16	
28	
26	
24	

Программа на школьном алгоритмическом языке:

```

алг
нач цел N, a, кол, ном
  ввод N
  кол := 0
  нц для ном от 1 до N
    ввод a
    если mod(a, 4) = 0 и mod(a, 7) <> 0
      то
        кол := кол + 1
  все
кц
вывод кол
кон
  
```

3. Задания для самостоятельной работы

Во всех заданиях:

- количество чисел не превышает 1000;
- введённые натуральные числа не превышают 30 000, а при наличии в последовательности и отрицательных чисел все числа по модулю не превышают 30 000;
- в последовательности всегда имеется число с заданными свойствами.

1. Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. Программа должна вывести одно число – количество чисел, оканчивающихся на 6.

Пример работы программы:

Входные данные	Выходные данные
3 16 26 24	2

2. Напишите программу, которая в последовательности натуральных чисел определяет сумму нечётных чисел. Программа получает на вход количество чисел в последовательности, а затем сами числа. Программа должна вывести одно число – сумму нечётных чисел.

Пример работы программы:

Входные данные	Выходные данные
3 13 23 24	36

3. Напишите программу, которая в последовательности целых чисел определяет среднее арифметическое неотрицательных чисел. Программа получает на вход количество чисел в последовательности, а затем сами числа. Программа должна вывести одно число – среднее арифметическое неотрицательных чисел.

Пример работы программы:

Входные данные	Выходные данные
4 -4 2 0 4	2.0

4. Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 5, но не оканчивающих на 0. Программа получает на вход количество чисел в последовательности, а затем сами числа. Программа должна вывести одно число – количество чисел, кратных 5, но не оканчивающих на 0.

Пример работы программы:

Входные данные	Выходные данные
5 20 26	2

25 5 100	
----------------	--

5. Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, оканчивающихся цифрами 123. Программа получает на вход количество чисел в последовательности, а затем сами числа. Программа должна вывести одно число – сумму чисел, оканчивающихся цифрами 123.

Пример работы программы:

Входные данные	Выходные данные
4 123 23 2123 23	2246

6. Напишите программу, которая в последовательности чисел определяет максимальное положительное число. Программа получает на вход количество чисел в последовательности, а затем сами числа. Программа должна вывести одно число – максимальное положительное число.

Пример работы программы:

Входные данные	Выходные данные
4 4 -1 6 4	6

7. Напишите программу, которая в последовательности натуральных чисел определяет номер максимального из чисел, больших 100 и оканчивающихся на 7. Программа получает на вход количество чисел в последовательности, а затем сами числа. Если в последовательности будет несколько таких чисел, должен быть выведен номер *первого* из них».

Пример работы программы:

Входные данные	Выходные данные
4 7 247 340 247	2

4. Частный случай задания

В заключение обсудим особенности выполнения варианта задания, в котором количество введённых чисел неизвестно, а известно, что последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Так как известно условие окончания цикла ввода и обработки чисел последовательности, то в программе можно использовать так называемый «оператор цикла с постусловием». Его общий вид такой:

```
нц
  <тело оператора цикла>
кц_при <условие окончания цикла>
```

причём в теле оператора цикла все действия должны выполняться только для введённых чисел, отличных от нуля:

```
нц
  ввод а
  если а <> 0
    то
      ...
  все
кц_при а = 0
```

Например, в этом случае определение суммы всех чисел последовательности в программе происходит так:

```
сум := 0
нц
  ввод а
  если а <> 0
    то
      сум := сум + а
  все
кц_при а = 0
```

В языке Паскаль общий вид оператора цикла с постусловием следующий:

```
repeat
  <тело оператора цикла>
until <условие окончания цикла>
```

а фрагмент программы определения суммы всех чисел последовательности:

```
sum := 0;
repeat
  readln(a);
  if a <> 0 then
    sum := sum + a
until a = 0;
```

В языке Python оператор цикла с постусловием не предусмотрен, но его можно организовать с помощью оператора цикла с предусловием так:

```
while True:
  a = int(input())
  if a != 0:
    sum = sum + a
  else:
    break
```

Цикл, который начинается с заголовка **while** True, будет выполняться бесконечно, потому что условие True (логическая константа) всегда истинно. Выйти из такого цикла можно только с помощью специальной инструкции **break** (досрочный выход из цикла). В данном случае она сработает тогда, когда станет истинным условие **a == 0**, то есть когда будет введено нулевое значение.

Можно также в программе можно применить так называемый «оператор цикла с предусловием». Его общий вид следующий:

```
нц пока <условие продолжения цикла>
  <тело оператора цикла>
кц
```

Условие продолжения цикла противоположно условию его окончания, то есть в данном случае общий вид оператора такой:

```
нц пока а <> 0
    <тело оператора цикла>
кц
```

Но при этом, чтобы оператор цикла начал «работать», нужно, чтобы условие в нем было истинным. Значит, в программе на предварительно величине *a* присвоить (условно) некоторое ненулевое значение:

```
а := 100
нц пока а <> 0
    ввод а
    если а <> 0
        то
            ...
    все
кц
```

Например, в этом случае определение количества всех чётных чисел последовательности в программе происходит так:

```
кол := 0
а := 100
нц пока а <> 0
    ввод а
    если а <> 0
        то
            кол := кол + 1
    все
кц
```

Задания для самостоятельной работы

Напишите программы решения задач, описанных в заданиях 1–7 выше для случая, когда количество введённых чисел неизвестно, а известно, что последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

ЕГЭ

О заданиях 13 ЕГЭ по информатике

1. Общие вопросы

В Спецификации контрольных измерительных материалов ОГЭ по информатике в качестве проверяемого результата обучения применительно к заданию 7 указывается «Знание принципов адресации в сети Интернет».

1. Общие вопросы

В Спецификации [1] в качестве проверяемого элемента содержания применительно к заданию 13 указывается «Умение использовать маску подсети»⁶.

Уровень сложности задания – II (повышенный).

Максимальный балл за выполнение задания – 1.

Примерное время выполнения задания – 3 мин.

⁶ Такой проверяемый элемент содержания впервые появился в демонстрационном варианте ЕГЭ 2024 года.

Напомним определения ряда терминов.

Узел компьютерной сети – устройство, соединенное с другими устройствами этой сети (узлами могут быть компьютеры, мобильные телефоны, а также специальные сетевые устройства, такие как маршрутизатор, коммутатор или концентратор).

IP-адрес узла (от англ. *Internet Protocol*) – уникальный числовой идентификатор устройства в компьютерной сети, работающей по протоколу IP.

Чаще всего IP-адрес узла – это четыре десятичных числа, которые разделены между собой точкой (такой формат поддерживается в протоколе IPv4). Пример: 192.168.1.49. В каждой части могут быть числа от 0 до 255.

IP-адрес узла состоит из двух частей: адреса сети, в которую входит данный узел, и номера узла в этой сети.

Чтобы определить, какая часть IP-адреса узла сети относится к адресу сети, а какая – к номеру самого узла в этой сети, используется так называемая «маска сети».

Маска подсети состоит из тех же четырех чисел и похожа на IP-адрес, например, 255.255.0.0.

Напомним также о том, что такое «поразрядная конъюнкция».

Поразрядная конъюнкция – это операция над двумя десятичными числами, результат которой определяется следующим образом:

- 1) оба числа-операнда переводятся в двоичную систему счисления;
- 2) в каждом разряде двоичных представлений операндов происходит сложение цифр согласно такой таблице сложения:

Цифры	0	0	1	1
в разряде	0	1	0	1
Результат	0	0	0	1

Внимание! При поразрядной конъюнкции двух цифр 1 и 1 перенос в старший разряд не происходит.

- 3) полученный двоичный результат переводится в десятичную систему.

Например, при поразрядной конъюнкции чисел 23 и 13:

$$23_{10} = 10111_2$$

$$13_{10} = 1101_2$$

$$\begin{array}{r} 1\ 0\ 1\ 1\ 1 \\ 1\ 1\ 0\ 1 \\ \hline 0\ 0\ 1\ 0\ 1 \end{array}$$

Результат равен $101_2 = 5_{10}$.

Возможны три основных типа заданий на использование масок сетей:

- связанные с определением адреса сети;
- связанные с узлами сети (нахождение их количества, порядкового номера узла и др.);
- связанные с определением маски сети или ее части.

Примеры заданий и методика их выполнения приведены в следующем разделе.

2. Примеры заданий и методика их выполнения

2.1. Пример 1

Условие

Сеть задана IP-адресом 182.110.32.111 и маской сети 255.255.255.240. Определите адрес сети:

- а) как последовательность двоичных цифр;
- б) в десятичном виде;
- в) как последовательность шестнадцатеричных цифр.

Решение

Переведем все десятичные числа в IP-адресе и маске сети в двоичную систему счисления. На экзамене это можно сделать с помощью стандартной программы Калькулятор, в режиме Программист (см. главу 5)⁷:

10110110.01101110.00100000.01101111
 11111111.11111111.11111111.11110000
 (Обратите внимание на добавленные оттененные нули!)

Проведем поразрядную конъюнкцию над каждой частью IP-адреса и маски:

$$\begin{array}{r} 10110110. 01101110. 00100000. 01101111 \\ 11111111. 11111111. 11111111. 11110000 \\ \hline 10110110. 01101110. 00100000. 01100000 \end{array}$$

(Для наглядности части адреса и маски отделены.)

Полученный результат и будет искомым адресом сети.

Другие требуемые виды адреса можно также получить из двоичного вида с помощью программы Калькулятор:

182.110.32.96
 В6.6E.20.60

Обратим внимание на то, что если некоторая часть маски равна 255, то соответствующая часть адреса сети будет такой же, как аналогичная часть IP-адреса.

Добавим также, что в ряде языков программирования предусмотрена операция поразрядной конъюнкции десятичных значений, то есть нужные десятичные значения адреса сети можно получить с помощью несложной программы на таком языке.

Адрес сети в двоичном виде определяется в результате поразрядной конъюнкции IP-адреса узла этой сети и ее маски.

2.2. Пример 2

Условие

Сеть задана IP-адресом 180.125.32.166 и маской сети 255.255.255.240. Сколько узлов возможны в этой сети?

Решение

Как и в предыдущем задании, выделим из заданного IP-адреса адрес сети:

$$\begin{array}{r} 10110100. 01111101. 00100000. 10100110 \\ 11111111. 11111111. 11111111. 11110000 \\ \hline 10110100. 01111101. 00100000. 10100000 \end{array}$$

Но вопрос в условии связан с узлами этой сети. Как выделить их?

Обсудим вопрос – сколько последних разрядов маски могут быть использованы для обозначения узлов в найденной сети? Ответ – только последние четыре цифры (если в пятом справа разряде записать 1, то тогда изменится адрес сети – убедитесь в этом!). Значит, для обозначения узлов в данном случае могут быть использованы только последние разряда. Всего возможны $2^4 = 16$ вариантов номеров (от 0000 до 1111).

Ответ: 16.

⁷ Двоичное представление числа 255 должно быть читателю известно.

Последние рассуждения позволяют сделать важный вывод – для обозначения узлов сети, заданной некоторой маской, в общем случае могут быть использованы только двоичные разряды после последней единицы двоичного представления маски.

Так как все возможные маски сети представляют собой (слева направо) последовательность единиц, после которой идет последовательность нулей, то можно сказать, что для обозначения узлов сети, заданной некоторой маской, в общем случае могут быть использованы ее разряды, в которых записаны нули.

Расскажем, какие IP-адреса будут у узлов рассмотренной в задании сети.

У первого узла IP-адрес совпадает с адресом сети: 0110100.01111101
00100000.10100000.

У второго – четвертая часть будет на 1 больше: 0110100. 01111101. 00100000.
10100001.

...

У последнего узла сети: 0110100.01111101.00100000.10101111 (это так называемый «широковещательный адрес»).

Общее количество узлов сети равно 2^n , где n – число нулей в маске этой сети.

2.3. Пример 3

Условие

Для некоторой сети используется маска 255.255.254.0. Сколько различных IP-адресов узлов, доступных для пользователей, допускает эта маска?

Решение

Здесь надо обратить внимание на фразу «доступных для пользователей». Дело в том, что из всех возможных адресов узлов сети два IP-адреса не используют. Это адрес, совпадающий с адресом самой сети, и широковещательный адрес (см. чуть выше).

Поэтому искомое в задании количество на два меньше общего числа узлов, возможных в данной сети.

Как отмечалось при выполнении предыдущего задания, за номера узлов в маске сети «отвечают» разряды, содержащие нули. Определим их количество.

В маске 255.255.254.0 первые два числа состоят полностью из единиц. Двоичное значения числа в третьей части маски: 11111110. Четвертое число маски в двоичном представлении состоит из восьми нулей. Значит, общее количество нулей в маске равно 9.

В девяти разрядах можно записать $2^9 = 512$ двоичных чисел. Из них для IP-адресов узлов, доступных для пользователей, – 510 чисел.

Ответ: 510.

2.4. Пример 4 (задание из демонстрационного варианта ЕГЭ 2024 года)

Условие

В терминологии сетей TCP/IP маской сети называют двоичное число, которое показывает, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу узла в этой сети. Адрес сети получается в результате применения поразрядной конъюнкции к заданному адресу узла и маске сети. Сеть задана IP-адресом 192.168.32.160 и маской сети 255.255.255.240. Сколько в этой сети IP-адресов, для которых сумма единиц в двоичной записи IP-адреса четна?

Решение

Сначала надо определить общее количество разрядов для обозначения номеров узлов в сети. Хотя это можно сделать только на основе анализа двоичного представления маски, найдем, как ранее, и адрес сети:

11000000. 10101000. 00100000. 10100000
<u>11111111. 11111111. 11111111. 11110000</u>
11000000. 10101000. 00100000. 10100000

Согласно сделанному в п. 2.2 выводу, для номеров узлов указанной сети можно выделить четыре разряда. Так как, в отличие от предыдущего задания, речь не идет о адресах, доступных для пользователей, общее количество узлов в этой сети равно $2^4 = 16$.

Но в данном случае нас интересуют не все 16 адресов, а только те, у которых сумма единиц в двоичной записи IP-адреса четна.

Исследуем вопрос, связанный с четностью суммы единиц в двоичной записи *всего* IP-адреса узла данной сети.

В его «сетевой» части 8 единиц. Значит, в адресах узлов этой сети также должно быть четное число единиц. В четырех разрядах это:

- 1) все нули (такое число одно – 0000);
 - 2) все единицы (такое число тоже одно – 1111);
 - 3) две единицы (количество таких чисел равно числу сочетаний из 4 по 2, то есть 6).
- Общее число искомых адресов – 8.

Ответ: 8.

2.5. Пример 5

Условие

Если маска сети равна 255.255.255.240, а IP-адрес узла – 58.68.0.44, то чему равен порядковый номер этого узла в этой сети?

Решение

Заданные IP-адрес узла и маска сети, а также соответствующий адрес сети:

```
00111010.01000100.00000000.00101100
11111111.11111111.11111111.11110000
00111010.01000100.00000000.00100000
```

Как уже говорилось (см. п.2.2), адрес узла, совпадающий с адресом самой сети (который, напомним, для пользователей недоступен) имеет = порядковый номер – 1. IP-адреса узлов с остальными порядковыми номерами (всего их в данном случае – 16) приведены в таблице:

Номер узла	IP-адрес узла
1	00111010.01000100.00000000.00100000
2	00111010.01000100.00000000.00100001
3	00111010.01000100.00000000.00100010
4	00111010.01000100.00000000.00100011
5	00111010.01000100.00000000.00100100
6	00111010.01000100.00000000.00100101
7	00111010.01000100.00000000.00100110
8	00111010.01000100.00000000.00100111
9	00111010.01000100.00000000.00101000
10	00111010.01000100.00000000.00101001
11	00111010.01000100.00000000.00101010
12	00111010.01000100.00000000.00101011
13	00111010.01000100.00000000.00101100
14	00111010.01000100.00000000.00101101
15	00111010.01000100.00000000.00101110
16	00111010.01000100.00000000.00101111

Какой номер у узла в условии (00111010.01000100.00000000.00101100)? – 12.

Но как получить его, не оформляя таблицу? – Анализ чисел в таблице показывает, что это можно сделать так:

- 1) определить количество цифр в цепочке цифр адреса сети, начинающейся с последней единицы (100000), – оно равно 6 и запомнить соответствующее шестизначное число;
- 2) выделить число, состоящее из этого количества последних цифр IP-адреса (101100);
- 3) вычесть из второго двоичного числа первое: $101100 - 100000 = 1100$ (разность можно получить, отбросив во втором числе начальную единицу);
- 4) полученную разность представить в десятичной системе: $1100_2 = 12_{10}$.

Можно также поступить следующим образом:

- 1) инвертировать двоичное представление маски, то есть заменить все единицы на нули и наоборот: 00000000.00000000.00000000.00001111;
- 2) провести поразрядную конъюнкцию этого представления маски и IP-адреса узла:

$$\begin{array}{r}
 \text{Начальные нули можно не выписывать } 1111 \\
 00111010.01000100.00000000.00101100 \\
 \hline
 \text{Начальные нули можно не выписывать } 1100
 \end{array}$$

- 3) результат перевести в десятичную систему.

Описанные действия можно провести быстрее: выписать только единицы маски и такое же количество последних цифр IP-адреса узла, после чего провести поразрядную конъюнкцию выписанных чисел.

2.6. Пример 6

Условие

Если маска сети 255.255.224.0, а IP-адрес компьютера – 206.158.124.67, то чему равен порядковый номер узла в этой сети?

Решение

В двоичном виде заданный IP-адрес: 11001110.10011110.01111100.01000011, а определенный с помощью маски адрес сети: 11001110.10011110.01100000.00000000.

Согласно методике, описанной при выполнении предыдущего примера:

- 1) цепочка цифр адреса сети, начинающаяся с последней единицы: 10000000000000; в ней – 14 цифр (1 + 5 + 8);
 - 2) число из последних 14 двоичных цифр IP-адреса: 11110001000011;
 - 3) их разность 1110001000011 (в данном случае также – второе число без начальной единицы);
 - 4) соответствующее десятичное число: 7235.
- Ответ:* 7235.

Решение с инвертированием маски:

$$\begin{array}{r}
 11111.1111111 \\
 11100.01000011 \\
 \hline
 11100 01000011
 \end{array}$$

2.7. Пример 7

Условие

Для узла с IP-адресом 153.209.23.240 адрес сети равен 153.209.20.0. Чему равна третья слева часть маски? Ответ запишите в виде десятичного числа.

Решение

Прежде чем приводить решение, зададим вопрос: «Чему равны первые две части маски?» и ответим на него.

Обратим внимание на то, что первые две части IP-адреса узла и адреса сети совпадают. В каком случае после проведения поразрядной конъюнкции IP-адреса узла и маски это может быть? – Если в двоичных представлениях этих частей маски записаны только единицы

(убедитесь в этом, вспомнив таблицу сложения при поразрядной конъюнкции!), то есть десятичные значения этих частей – 255.

Почему третьи части IP-адреса узла и адреса сети разные? – Потому что в этой части маски представлены и нули. Начиная с какого разряда слева?

Методика поиска такого разряда следующая.

Представим третьи части IP-адреса и адреса сети в виде восьмиразрядных двоичных чисел и выпишем их одно под другим:

$$23_{10} = 00010111_2$$

$$20_{10} = 00010100_2$$

Ищем первое слева сочетание цифр «1 в IP-адресе и 0 в адресе сети». Оно – в седьмом разряде (втором справа).

В каком случае может быть такое сочетание цифр? – Когда в этом же разряде маски записан 0.

Является ли этот ноль первым в цепочке цифр маски? – В данном случае (!) является, так как слева от него в маске записана единица (убедитесь в этом).

Так как это первый ноль в маске, то двоичный вид третьей части маски: 11111100. Переведа это число в десятичную систему счисления, получим 252⁸.

Ответ: 252.

Дополнительный вопрос: «Какое десятичное значение у четвертой части маски?». Ответьте на него.

2.8. Пример 8

Условие

Для узла с IP-адресом 111.81.208.27 адрес сети равен 111.81.192.0. Чему равно наименьшее возможное значение третьей слева части маски? Ответ запишите в виде десятичного числа.

Решение

Здесь также первые две части маски в десятичном виде имеют значения 255. Исследуем третью часть.

Запишем третьи части IP-адреса и адреса сети в виде 8-разрядных двоичных чисел и выпишем их одно под другим:

$$208_{10} = 11010000_2$$

$$192_{10} = 11000000_2$$

Здесь первое слева сочетание цифр «1 в IP-адресе и 0 в адресе сети» находится в четвертом разряде. Является ли этот ноль этого разряда маски первым в цепочке ее цифр? – Необязательно, так ноль *может быть* и слева от него (в третьем разряде). Значит, возможны два варианта (два числа с различными третьими слева цифрами) – 11000000 или 11100000. По условию задачи требуется найти наименьшее значение – это 11000000. Переведа его в десятичную систему, получим $11000000_2 = 192_{10}$.

Ответ: 192.

Обобщая два последних примера, общую методику определения положения первого нуля в маске можно описать следующим образом:

- определяются отличающиеся части IP-адреса и адреса сети;
- эти части записываются в виде восьмиразрядных двоичных чисел;
- ищется первое слева сочетание цифр «1 в IP-адресе и 0 в адресе сети»;
- возможны два варианта, которые схематически можно оформить так:

⁸ Такой результат можно получить, вычтя из 255 число 3.

1)

IP-адрес							1			
Адрес сети						1	0			
Маска	1	1	1	...	1	1	0	0	...	0

2)

IP-адрес							1			
Адрес сети			1				0	0		
Маска		1	1	?	?	?	?	0	0	...

В первом случае маска определяется однозначно, во втором – возможны варианты, из которых надо выбрать минимальное или максимальное значение данной части маски;

- найденное двоичное значение части маски переводится в десятичную систему.

Можно было также при выполнении примера 8 рассуждать так.

С каким числом нужно произвести поразрядную конъюнкцию числа 11010000_2 , чтобы получить 11000000_2 ? – Очевидно, что первые две цифры этого числа должны быть единицами, а четвертая – нулем. Такими числами могут два числа с различными третьими слева цифрами – 11000000 или 11100000 . Из них минимальное – первое.

3. Задания для самостоятельного выполнения

1. Сеть задана IP-адресом 10.50.128.6 и маской сети 255.255.255.0. Определите адрес сети.
2. Сеть задана IP-адресом 206.158.124.67 и маской сети 255.255.224.0. Сколько узлов возможны в этой сети?
3. Для некоторой сети используется маска 255.255.248.0. Сколько различных IP-адресов узлов, доступных для пользователей, допускает эта маска?
4. Сеть задана IP-адресом 11.53.128.33 и маской сети 255.255.255.248. Сколько в этой сети IP-адресов, для которых сумма единиц в двоичной записи IP-адреса кратна пяти.
5. Сеть задана IP-адресом 181.53.128.45 и маской сети 255.255.255.248. Сколько в этой сети IP-адресов, для которых сумма единиц в двоичной записи IP-адреса нечетная?
6. Если маска сети равна 255.255.240.0, а IP-адрес узла в сети – 162.198.75.44, то чему равен порядковый номер этого узла в этой сети?
7. Для узла с IP-адресом 224.128.112.142 адрес сети равен 224.128.64.0. Чему равна третья слева часть маски? Ответ запишите в виде десятичного числа.
8. Для узла с IP-адресом 158.198.228.220 адрес сети равен 158.198.128.0. Чему равна третья слева часть маски? Ответ запишите в виде десятичного числа.
9. Для узла с IP-адресом 225.209.23.200 адрес сети равен 225.209.23.192. Чему равно наибольшее возможное значение последней части маски? Ответ запишите в виде десятичного числа.

Итоги конкурса № 66

Напомним, что в качестве задания конкурса было предложено решить головоломку, в которой в таблице нужно синие клетки заполнить натуральными числами от 1 до 16 так, чтобы каждое число встречалось *ровно* один раз и все равенства были выполнены. Некоторые числа уже записаны.

	−		+	11	−	14	−6
−		*		*		−	
1	*	10	−		+		18
*		+		−		−	
	*		−		−	3	58
−		−		−		+	
	+	12	+		+		36
−6		91		82		10	

Решение

6	−	9	+	11	−	14	−6
−		*		*		−	
1	*	10	−	8	+	16	18
*		+		−		−	
5	*	13	−	4	−	3	58
−		−		−		+	
7	+	12	+	2	+	15	36
−6		91		82		10	

Ответы на задание (не простое!) представили и стали победителями конкурса:

— Анохина Екатерина, Ломакина Ульяна, Попова Полина и Скорикова Алина, Волгоградская обл., г. Фролово, школа № 1 имени А.М. Горького, учитель Фирсова М.Н.

— Бородин Антон и Шахназаров Армен, Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецкая Т.С.**;

— Вагизов Ислам, Республика Татарстан, Мамадышский р-н, совхоз “Мамадышский”, политехнический колледж, преподаватель **Порываева Н.С.**;

— Лубков Виктор и Чемизов Артур, г. Пенза, школа № 73, учитель **Диков А.В.**;

— Кобзев Максим, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Чумаков Леонид, Хабаровский край, г. Комсомольска-на-Амуре, школа № 38, учитель **Плюснина Н.П.**

Все перечисленные читатели будут награждены дипломами. Поздравляем!

Отметим ответы Ислама Вагизова, представившего развёрнутое обоснование решения, и Леонида Чумакова, использовавшего, наряду с рассуждениями, электронную таблицу.

Ответы и решения задач и головоломок, опубликованных в интернет-журнале «Мир информатики» № 87 (март 2024 г.), представили:

— Амосов Максим и Зубрицкая Мария, средняя школа села Сердар, Республика Марий Эл, учитель **Чернова Л.И.**;

— Грезин Михаил, ученик 3-го (!) класса школы № 11 г. Струнино Владимирской обл., учитель **Волкова Т.П.**;

— Джафаров Вагиф, Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецакая Т.С.**;

— Голубева Марина и Макаренко Анастасия, средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Ковальчук Иван, Смирнова Полина, Сысуев Илья и Терещенко Артём, Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Козырева Ольга, Рогов Максим и Ушакова Елена, г. Пенза, школа № 73, учитель **Диков А.В.**;

— Колесниченко Степан, Кудряшев Никита, Пилипенко Виктория, Ткачева Варвара и Третьякова Анна, Хабаровский край, г. Комсомольска-на-Амуре, школа № 38, учитель **Плюснина Н.П.**;

— Судаков Владислав, средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**

СПАСИБО ВСЕМ! МОЛОДЦЫ!

Уважаемые читатели!

В июньском номере нашего журнала возобновится публикация рубрики «Доска почёта “Мира информатики”», в которой будут опубликованы фотографии ВСЕХ учащихся, в 2023-2024 учебном году принявших участие в конкурсах и в решении задач и головоломок.

Просим присылать фотографии (можно и не индивидуальные, лучше – неформальные) по адресу mirinfo@infojournal.ru (тема письма: «Доска почёта»). Учащиеся могут присылать фотографии самостоятельно. Срок представления – 5 июня 2024 г.