

Мастер - класс по решению задач С1 и С3



Учитель информатики
ГБОУ СОШ №5 «ОЦ»
Константинова С. А.

Что общего между решением задач 25 и 27А?

I
• по сути, в этих задачах обрабатываются данные собираемые в массив.

II
• навыки обработки массивов, применяемые для решения 25 задачи, применимы и для 27А

III
• в этих заданиях не стоит цель оптимальности написанной программы, нужно просто хорошо знать средства и возможности выбранного языка программирования по обработке массивов.

Требования к уровню подготовки выпускников, достижение которых проверяется на едином государственном экзамене по информатике и ИКТ, необходимые для решения задачи 25:

- ✓ **Запись натурального числа в позиционной системе с основанием меньшим или равным 10. Обработка и преобразование такой записи числа.**
- ✓ **Нахождение сумм, произведений элементов данной конечной числовой последовательности (или массива).**
- ✓ **Использование цикла для решения простых переборных задач (поиск наименьшего простого делителя данного натурального числа, проверка числа на простоту, и т.д.).**
- ✓ **Заполнение элементов одномерного и ~~двумерного~~ массива по заданным правилам.**
- ✓ **Операции с элементами массива. Линейный поиск элемента. Вставка и удаление элементов в массиве. Перестановка элементов данного массива в обратном порядке. Суммирование элементов массива. Проверка соответствия элементов массива некоторому условию.**
- ✓ **Нахождение второго по величине (второго максимального или второго минимального) значения в данном массиве за однократный просмотр массива.**
- ✓ **Нахождение минимального (максимального) значения в данном массиве и количества элементов, равных ему, за однократный просмотр массива.**
- ✓ **Операции с элементами массива, отобранными по некоторому условию.** (Например, нахождение минимального четного элемента в массиве, нахождение количества и суммы всех четных элементов в массиве.)
- ✓ **Сортировка массива.**

Технология решения задачи 25 ЕГЭ

Задачи этого типа направлены на обработку массива по некоторым критериям. Полностью программа состояла бы из:

- 1) Описания массива и типов используемых переменных, организация ввода данных (*уже есть*)
- 2) Инициализации начальных значений некоторых переменных (*требуется задать!*)
- 3) Обработки данных (*требуется организовать!*)
- 4) Вывода данных (*требуется организовать!*).

Однако, в 25 задаче учащемуся необходимо не писать программу полностью, а «дописать» её в рамках уже организованного ввода, а также заданного количества переменных и их типов.

Что может обрабатываться в массиве?

элементы массива по одному

пары элементов массива

тройки элементов массива

последовательности элементов

Обработка данных происходит посредством циклического перебора элементов массива и поиска по некоторому комбинированному условию, которое учащемуся необходимо выявить на основе анализа условия задачи.
(например, элемент массива не делится на 2 и кратен 3).

Примеры формализации некоторых типичных для задачи 25 составляющих комбинированного условия:

Признак	Пример его формального описания для некоторого элемента массива A (на языке Паскаль)
принадлежность к положительным /отрицательным числам	$A[i] > 0 / A[i] < 0$
четность/нечетность	$A[i] \bmod 2 = 0$
наличие указанного числа знаков в числе (только двузначное или др.)	(трехзначное натуральное число) $(A[i] \geq 100) \text{ and } (A[i] \leq 999)$
наличие указанной цифры в конце (начале) числа	(средняя цифра трехзначного числа – 5) $(A[i] \bmod 100) \div 10 = 5$
кратность некоторому числу	(делится на 7 нацело) $A[i] \bmod 7 = 0$
окончание числа на указанные цифры	(оканчивается на 5) $A[i] \bmod 10 = 5$
соответствие указанному значению абсолютного значения элементов массива	(элемент массива больше элемента слева и элемента справа – поиск «пика» в тройке элементов) $(A[i-1] < A[i]) \text{ and } (A[i] > A[i+1])$

Критерии проверки задачи 25 ЕГЭ по информатике

Предложен правильный алгоритм, выдающий в качестве результата верное значение	2
Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих:	1

- ✓используется переменная, не объявленная в разделе описания переменных;
- ✓отсутствует вывод ответа;
- ✓в цикле происходит выход за границу массива;
- ✓не инициализируется или неверно инициализируется счётчик
- ✓счётчик в цикле не изменяется или изменяется неверно;
- ✓неверно проверяется условие поиска элементов или условие применяется не к элементам массива, а их индексам;
- ✓в сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции «И» и «ИЛИ»,
- ✓неверно расставлены скобки в логическом выражении)
- ✓не указано или неверно указано условие завершения цикла;
- ✓индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно;
- ✓неверно расставлены операторные скобки.

7

Ошибок, перечисленных выше, две или больше, или алгоритм сформулирован неверно (в том числе при отсутствии цикла в явном или неявном виде)	0
--	---

Пример 25-1 ОБРАБОТКА ПАР ЭЛЕМЕНТОВ МАССИВА

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\ 000$ до $10\ 000$ включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов:

6; 2; 9; -3; 6 ответ: 4.

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

Алгоритмический язык

алг

нач

цел N = 20

целтаб a[1:N]

цел i, j, k

нц для i от 1 до N

ввод a[i]

кц

...

кон

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Опишите ... алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 6; 2; 9; -3 ; 6 — ответ: 4.

Паскаль

const

N = 20;

var

a: array [1..N] of integer;

i, j, k: integer;

begin

for i := 1 to N do

 readln(a[i]);

...

end.

Алгоритмический язык

алг
нач
цел $N = 20$
целтаб $a[1:N]$
цел i, j, k
нц для i от 1 до N
ввод $a[i]$
кц
...
кон

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Опишите ... алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3.

Алгоритмический язык

алг

нач

цел $N = 20$

целтаб $a[1:N]$

цел i, j, k

нц для i от 1 до N

ввод $a[i]$

кц

...

кон

Опишите ... алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3.

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\ 000$ до $10\ 000$ включительно.

$k := 0$

нц для i от 1 до $N-1$

кц

вывод k

Алгоритмический язык

алг

нач

цел $N = 20$

целтаб $a[1:N]$

цел i, j, k

нц для i от 1 до N

ввод $a[i]$

кц

...

$k := 0$

нц для i от 1 до $N-1$

кц

вывод k

кон

Опишите ... алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3.

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\ 000$ до $10\ 000$ включительно.

нц для i от 1 до $N-1$

если $\text{mod}(a[i], 3) = 0$ или $\text{mod}(a[i+1], 3) = 0$
то $k = k + 1$
все

кц

Алгоритмический язык

алг

нач

цел $N = 20$

целтаб $a[1:N]$

цел i, j, k

нц для i от 1 до N

ввод $a[i]$

кц

...

кон

Опишите ... алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3.

Вид решения учащегося на Алгоритмическом языке

$k := 0$

нц для i от 1 до $N-1$

если $\text{mod}(a[i], 3) = 0$ или $\text{mod}(a[i+1], 3) = 0$

то $k = k + 1$

все

кц

вывод k

Паскаль

```
const
N = 20;
var
a: array [1..N] of integer;
i, j, k: integer;
begin
for i := 1 to N do
readln(a[i]);
...
end.
```

Опишите ... алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 3.

Вид решения учащегося на языке Паскаль ABC.net

```
k := 0;
for i := 1 to N-1 do
if (a[i] mod 3=0) or (a[i+1] mod
3=0) then k:=k+1;
writeln(k);
```

Пример 25-2 ОБРАБОТКА ТРОЕК ЭЛЕМЕНТОВ МАССИВА (из тренировочных работ ЕГЭ 2014-2016г)

Дан массив, содержащий 2000 неотрицательных целых чисел. Пиком называется не крайний элемент массива, который больше обоих своих соседей. Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите: 'пика нет'.

Например, в массиве из шести элементов, равных соответственно 5, 9, 2, 18, 3, 7, есть два пика – 9 и 18, максимальный пик равен 18.

Напишите на одном из языков программирования программу для решения этой задачи. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных.

Алгоритмический язык

```
алг  
нач  
цел N=2000 | Изменять  
значение  
| этой переменной нельзя  
целтаб a[1:N]  
цел i, j, k  
нц для i от 1 до N  
ввод a[i]  
кц  
...  
кон
```

Дан массив, содержащий 2000 неотрицательных целых чисел. Пиком называется не крайний элемент массива, который больше обоих своих соседей. Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите «пика нет».

Например, в массиве из шести элементов, равных соответственно 5, 9, 2, 18, 3, 7, есть два пика – 9 и 18, максимальный пик равен 18.

Паскаль

```
const  
N=2000;  
var  
a: array [1..N] of integer;  
i, j, k: integer;  
begin  
for i:=1 to N do  
  readln(a[i]);  
...  
end.
```


Паскаль

```
const
N=2000;
var
a: array [1..N] of integer;
i, j, k: integer;
begin
for i:=1 to N do
readln(a[i]);

...

end.
```

Дан массив, содержащий 2000 неотрицательных целых чисел. Пиком называется не крайний элемент массива, который больше обоих своих соседей. Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите «пика нет».

Например, в массиве из шести элементов, равных соответственно 5, 9, 2, 18, 3, 7, есть два пика – 9 и 18, ¹⁷ максимальный пик равен 18.

Паскаль

```
const  
N=2000;  
var  
a: array [1..N] of integer;  
i, j, k: integer;  
begin  
for i:=1 to N do  
  readln(a[i]);  
  
...  
end.
```

Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите «пика нет».

k:=0; // макс пик

If k=0 then
 writeln('пика нет') else
 writeln(k);

Паскаль

```
const
N=2000;
var
a: array [1..N] of integer;
i, j, k: integer;
begin
for i:=1 to N do
readln(a[i]);

...

end.
```

k:=0;//макс пик

if k=0 then writeln('пика нет')
else writeln(k);

Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите «пика нет».

```
for i:=2 to N-1 do
begin
if (a[i]>a[i-1]) and
(a[i]>a[i+1]) and (a[i]>k)
then k:=a[i];
and;
```

Паскаль

```
const
N=2000;
var
a: array [1..N] of integer;
i, j, k: integer;
begin
for i:=1 to N do
readln(a[i]);

...

end.
```

Необходимо найти в массиве самый высокий пик, то есть пик, значение которого максимально. Если в массиве нет ни одного пика, выведите «пика нет».

решения учащегося на языке

Паскаль ABC.net

```
k := 0;
for i := 2 to N-1 do
begin
if (a[i]>a[i-1]) and (a[i]>a[i+1]) and
(a[i]>k)
then k:=a[i];
and;
if k=0 then writeln('пика нет')
else writeln(k);
```


Алгоритмический язык

алг
нач
цел $N=2000$ |
Изменять значение
| этой переменной
нельзя
целтаб $a[1:N]$
цел i, j, k
нц для i от 1 до N
ввод $a[i]$
кц
...

кон

Необходимо найти в массиве самый
высокий пик, то есть пик, значение
которого максимально. Если в массиве нет
ни одного пика, выведите «пика нет».

Вид решения учащегося на Алгоритмическом языке

$k:=0$
нц для i от 2 до $N-1$
если $(a[i]>a[i-1])$ и $(a[i]>a[i+1])$ и $(a[i]>k)$
то $k=a[i]$
все
кц
если $k=0$
то вывод 'пика нет'
иначе вывод k
все

Пример 25-3

ОБРАБОТКА ПОСЛЕДОВАТЕЛЬНОСТЕЙ ЭЛЕМЕНТОВ МАССИВА

Дан массив, содержащий 30 целых чисел. Необходимо подсчитать максимальное количество подряд идущих отрицательных элементов в этом массиве.

Паскаль

```
const
N=30;
var
a: array [1..N] of integer;
i, m, k: integer;
begin
for i:=1 to N do
readln(a[i]);
...
end.
```

Напишите на одном из языков программирования программу для решения этой задачи. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных.

Паскаль ABC.net

```
const
N=30;
var
a: array [1..N] of integer;
i, m, k: integer;
begin
for i:=1 to N do
readln(a[i]);
```

```
k:=0;// к-во отрицат. эл-тов
m:=0;//макс. к-во отр. эл-тов
```

```
if k>m then m:=k;
writeln(m);
```

```
...
end.
```

Необходимо подсчитать максимальное количество подряд идущих отрицательных элементов в массиве из 30 элементов.

```
for i:=1 to N do
begin
```

```
if a [ i ] < 0 then k := k + 1
else
begin
if k > m then m := k;
k := 0;
end;
```

```
and;
```

Алгоритмический язык

алг

нач

цел $N=30$

целтаб $a[1:N]$

a : array $[1..N]$ of integer;

цел i, m, k

нц для i от 1 до N

ввод $a[i]$

кц

$k:=0$

$m:=0$

если $k>m$ то $m:=k$

все

Вывод m

...

кон

Необходимо подсчитать максимальное количество подряд идущих отрицательных элементов в массиве из 30 элементов.

нц для i от 1 до N

если $a[i] < 0$

то $k := k + 1$

иначе

если $k > m$ то

$m := k$ все

$k := 0$

все

кц

Питон 3.3

```
N=30  
a = []  
for i in range(0, N):  
    a.append(int(input()))
```

Необходимо подсчитать максимальное количество подряд идущих отрицательных элементов в массиве из 30 элементов.

```
k=0 #к-во отрицат. эл-тов  
m=0 #макс. к-во отр. эл-тов
```

```
if k>m: m=k  
print(m)
```

...

```
for i in range (0, N):
```

```
    if a [ i ] <0: k=k+1  
    else:
```

```
        if k> m: m=k  
        k=0
```

Питон 3.3

25-4 Задание для самостоятельной работы:

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Опишите на одном из языков программирования алгоритм, который находит и выводит значение 3-го положительного элемента массива (если из массива вычеркнуть все неположительные элементы, этот элемент стоял бы в получившемся массиве на третьем месте). Если в массиве меньше, чем три положительных элемента, вывести об этом сообщение. Исходные данные объявлены так, как показано ниже.

Алгоритмический язык

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, j, k
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

Паскаль

```
const
  N = 40;
var
  a: array [1..N] of integer;
  i, j, k: integer;
begin
  for i:=1 to N do
    readln(a[i]);
  ...
end.
```

Особенности решения задачи 27 ЕГЭ по информатике или что такое 27А?

**Решением задания 27А является написанная учеником программа не эффективная по времени и памяти:
вводимые данные сначала с помощью цикла запоминаются в массив, а после (другим циклом) массив данных обрабатывается.
Именно так, как это делается в задаче 25.**

**Задачи 27А и 27Б проверяют умение
создавать собственные программы на
языке программирования (30–50 строк)
для решения задач средней сложности.**

Следует иметь в виду:

Из демоверсии ЕГЭ 2016:

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

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

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

Критерии оценивания 27А

Критерии оценивания задания А	
<p>Программа решает поставленную задачу для любых соответствующих условию входных данных. Например, допускается переборное решение, аналогичное приведённой выше программе 4.</p> <p>Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии оценивания задания Б на 4 балла).</p> <p>Допускается до двух содержательных ошибок, описанных в критериях оценивания задания Б на 3 балла</p>	2
<p>Не выполнены условия, позволяющие поставить 2 балла.</p> <p>Из описания алгоритма или общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. При этом программа может быть представлена отдельными фрагментами, без ограничений на количество синтаксических и содержательных ошибок. 1 балл ставится также за решения, верные лишь в частных случаях</p>	1
<p>Не выполнены критерии, позволяющие поставить 1 или 2 балла</p>	0

Для получения максимального балла за задачу 27 от учащегося требуется:

1. уметь анализировать «накрученное» условие и представлять метод решения в виде формальной записи на языке программирования;
2. хорошо знать способы описания и обработки переменных целого, вещественно, строкового, символьного, логического типов данных
3. обладать знаниями и умениями из уже рассмотренного в 25 перечня плюс следующими:
 - А. Слияние двух упорядоченных массивов в один без использования сортировки.
 - В. Обработка отдельных символов данной строки. Подсчет частоты появления символа в строке.
 - С. Работа с подстроками данной строки с разбиением на слова по пробельным символам. Поиск подстроки внутри данной строки, замена найденной подстроки на другую строку.
4. Иметь достаточный опыт написания программ, чтобы «чувствовать» и исправлять возможные ошибки без использования компьютера.

Алгоритм решения задачи

27А:

- ✓ Анализ условия, определение цели.
- ✓ Выявление связей между искомыми и вводимыми данными.
- ✓ Запись описания алгоритма решения задачи на естественном языке.
- ✓ Оформление программы на языке программирования, включающее в себя: описание переменных и их типов, инициализацию нужных переменных, организацию ввода и обработки, организацию вывода.
- ✓ Анализ программы, выявление ошибок на 2-3 подобранных самостоятельно тестах (кроме данного в задаче), коррекция.

Пример 27А-1 (анализ решения из демоверсии КИМ 2016 года)

В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передаётся положительное целое число – текущее показание прибора «Сигма 2015».

Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000.

Временем, в течение которого происходит передача, можно пренебречь.

Необходимо вычислить «бета-значение» серии показаний прибора – минимальное чётное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удастся, ответ считается равным -1 .

Решение задачи 27А имеет следующие обязательные составляющие:

Описание используемых переменных, их типов, размера и типов элементов массива



Инициализация переменных



Организация ввода



Организация обработки



Организация вывода

Описание используемых переменных, их типов, размера и типов элементов массива

```
const s = 6; //требуемое  
расстояние между показаниями  
var  
N: integer;  
a: array[1..10000] of integer;  
//все показания прибора  
mp: integer; //мин значение  
произв-я  
i, j: integer;
```

По каналу связи каждую минуту в лабораторию передаётся положительное целое число...

...Количество передаваемых чисел в серии известно и не превышает 10 000.

Все числа не превышают 1000. ...

Инициализация переменных

Ввод данных

По каналу ... передается положительное целое число –... Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000.

Найти: минимальное чётное произведение двух показаний

```
begin
readln(N); //кол-во измерений
mp := 1000 * 1000 + 1; // значение для минимума
for i:=1 to N do readln(a[i]); // организация
считывания значений измерений и занесение их в массив

...
end.
```

Организация обработки массива

Необходимо вычислить «бета-значение»
серии показаний прибора –
минимальное чётное произведение двух
показаний, между моментами передачи
которых прошло не менее 6 минут. Если
получить такое произведение не
удастся, ответ считается равным -1 .

```
...  
for i := 1 to N-s do begin  
  for j := i+s to N do begin  
    if (a[i]*a[j] mod 2 = 0) and (a[i]*a[j] < mp)  
      then mp := a[i]*a[j]  
  end;  
end;  
...  
...
```


Организация вывода

Необходимо вычислить «бета-значение» серии показаний прибора – минимальное чётное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удаётся, ответ считается равным -1.

```
...  
if mp = 1000 * 1000 + 1 then writeln(' -1' ) else  
writeln(mp); // если значение tr в массиве не изменилось, то  
нужное произведение элементов не найдено и надо выводить -1  
...
```

Общий вид решения задачи 27A на языке Паскаль ABC.net

```
const s = 6; //  
var  
N: integer;  
a: array[1..10000] of integer;  
mp: integer;  
i, j: integer;
```

Необходимо вычислить «бета-значение» серии показаний прибора – минимальное чётное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удаётся, ответ считается равным -1.

```
begin  
readln(N);  
mp := 1000 * 1000 + 1; // значение для минимума  
for i:=1 to N do readln(a[i]);
```

```
for i := 1 to N-s do begin  
  for j := i+s to N do begin  
    if (a[i]*a[j] mod 2 = 0) and (a[i]*a[j] < mp)  
      then mp := a[i]*a[j]  
    end;  
  End;
```

```
end.
```

```
if mp = 1000 * 1000 + 1 then writeln(' -1') else  
writeln(mp);
```

Пример 27А-2

Пример входных данных:

5
40
1000
7
28
55

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

Пример выходных данных для приведённого выше примера входных данных:

28000

Последовательность натуральных чисел характеризуется числом X – наибольшим числом, кратным 14 и являющимся произведением двух элементов последовательности с различными номерами.

Напишите программу ... находящую число X для последовательности натуральных чисел, значение каждого элемента которой не превосходит 1000.

Программа должна напечатать найденное число, если оно существует для заданной последовательности, или ноль в противном случае.

Перед текстом программы кратко опишите используемый Вами алгоритм решения.

Словесное описание алгоритма решения задачи, предваряющее ее запись на языке программирования является обязательной частью

Программа, вычисляющая число X , будет работать так:

1. Так как количество вводимых целых чисел не превышают 10000 и их максимальное значение не превышает 1000, то нам понадобятся массив размерности 10000 целого типа и переменные n , i, j и x .
2. Считаем значение n . Инициализируем x – самое большое число среди всех элементов последовательности. Чтобы найти максимум среди натуральных чисел начальное значение надо сделать равным 0.
3. Сначала программа с помощью цикла со счетчиком читает и запоминает все входные данные в массиве (ниже n – количество прочитанных чисел).
4. Затем с помощью цикла в цикле вычисляются произведения всех пар элементов последовательности и определяется среди них наибольшие кратное 14 произведение.
5. В конце программы выводится значение x .

алг

нач

цел i, j, n, x

целтаб $a[1..10000]$ of integer;

$x := 0$

ввод n

нц для i от 1 до n ввод $a[i]$

нц для i от 1 до $n - 1$

нц для j от $i + 1$ до n

если $(\text{mod}(a[i]*a[j]), 14) = 0$ и $((a[i]*a[j]) > \text{max})$

то $\text{max} := a[i]*a[j]$ все

кц

кц

вывод x

кон

```
var
i,j, n: integer;
x: integer;
a: array[1..10000] of integer;
begin
x := 0;
readln(n);
for i := 1 to n do readln(a[i]);

for i := 1 to n - 1 do
    for j := i + 1 to n do
        if ((a[i]*a[j]) mod 14 = 0) and ((a[i]*a[j]) > max)
        then max := a[i]*a[j];
writeln(x);
end.
```

Общий вид решения задачи

```
a=[]
n=int(input())
x = 0
for i in range (0, n): a.append(int(input()))

for i in range(0, n - 1):
    for j in range( i + 1, n):
        if ((a[i]*a[j]) % 14 == 0) and ((a[i]*a[j]) > max):
            max = a[i]*a[j]

print(x)
```

Пример 27А-3 Обработка целочисленных и строковых данных

Соревнования по игре «Тетрис-онлайн» проводятся по следующим правилам.

1. Каждый участник регистрируется на сайте игры под определённым игровым именем. Имена участников не повторяются.
2. Чемпионат проводится в течение определённого времени. В любой момент этого времени любой зарегистрированный участник может зайти на сайт чемпионата и начать зачётную игру. По окончании игры её результат (количество набранных очков) фиксируется и заносится в протокол.
3. Участники имеют право играть несколько раз. Количество попыток одного участника не ограничивается.
4. Окончательный результат участника определяется по одной игре, лучшей для данного участника.
5. Более высокое место в соревнованиях занимает участник, показавший лучший результат.
6. При равенстве результатов более высокое место занимает участник, раньше показавший лучший результат.

В ходе соревнований заполняется протокол, каждая строка которого описывает одну игру и содержит результат участника и его игровое имя. Протокол формируется в реальном времени по ходу проведения чемпионата, поэтому строки в нём расположены в порядке проведения игр: чем раньше встречается строка в протоколе, тем раньше закончилась соответствующая этой строке игра.

Описание входных данных

Первая строка содержит число N – общее количество строк протокола.

Каждая из следующих N строк содержит записанные через пробел результат участника (целое неотрицательное число, не превышающее 100 миллионов) и игровое имя (имя не может содержать пробелов). Строки исходных данных соответствуют строкам протокола и расположены в том же порядке, что и в протоколе.

Гарантируется, что количество участников соревнований не меньше 3.

Описание выходных данных

Программа должна вывести имена и результаты трёх лучших игроков по форме, приведённой ниже в примере.

Пример входных данных:

```
9
69485 Jack
95715 qwerty
95715 Alex
83647 M
197128 qwerty
95715 Jack
93289 Alex
95715 Alex
95715 M
```

Пример выходных данных для данных:

```
1 место. qwerty (197128)
2 место. Alex (95715)
3 место. Jack (95715)
```



```
var
a:array [1..1000] of longint;
b: array [1..1000] of string;
i,N: integer;
m1, m2, m3:longint;
begin
  readln(N);
  for i:= 1 to N do readln (a[i], b[i]);
  m1:=1; // индекс 1-го места
  m2:=1; // индекс 2-го места
  m3:=1; // индекс 3-го места
  for i:= 2 to N do
  begin
    if (a[i]>a[m1]) then m1:=i;
    if (i<>m1) and (b[i]<>b[m1]) and (a[i]>a[m2]) then m2:=i;
    if (i<>m1) and (b[i]<>b[m1]) and (i<>m2) and (b[i]<>b[m2])and(a[i]>a[m3]) then m3:=i;
  end;
  writeln('1 место.',b[m1], ' ',a[m1]);
  writeln('2 место.',b[m2],' ',a[m2]);
  writeln('3 место.',b[m3],' ',a[m3]);
end.
```

Словесное описание программы:

Нам потребуется 2 массива – числовой и строковый.

Считываем количество участников и одним циклом построчно заполняем оба массива.

Зададим начальное значение для индекса участников с 1,2 и 3 местом. Это потребуется для поиска трех максимальных результатов.

Начиная со второго элемента массивов ищем сначала индекс первого максимума, потом индекс второго максимума(не равного первому), потом третьего (отличного от первого и второго). При этом имеем в виду, что один и тот же участник турнира может занять только одно место и при равных баллах побеждает первый в списке.

В конце программы выводим победителей.