

Задания ЕГЭ. Часть 6

24 Исправление ошибок в
программе

25 Алгоритмы обработки массивов

26 Выигрышная стратегия

27 Программирование

24-2

Требовалось написать программу, при выполнении которой с клавиатуры вводятся последовательность из шести неотрицательных целых чисел, не превышающих 10^6 , подсчитывается и выводится сумма введённых чётных чисел или 0, если чётных чисел в последовательности нет. Ученик написал такую программу:

24-2 программа

```
var n, s: longint; i: integer;
```

```
Begin
```

```
    s:=1;
```

```
    for i:=1 to 6 do
```

```
        begin
```

```
            readln(n);
```

```
            if i mod 2 = 0 then s := s + n;
```

```
        end;
```

```
    write(s);
```

```
end.
```

24-2 задание к программе

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе по следовательности 1, 1, 2, 3, 5, 8.
2. Приведите пример последовательности, при вводе которой программа выдаст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде. Обратите внимание: Вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

24-2 Решение

Первая строка с ошибкой:

```
s:=1;
```

Исправленная строка:

```
s:=0;
```

Вторая строка с ошибкой:

```
if i mod 2 = 0 then
```

Исправленная строка:

```
if n mod 2 = 0 then
```

24-3

Дано целое положительное число N . Необходимо определить наименьшее целое число K , для которого выполняется неравенство:

$$1 + 2 + \dots + K > N.$$

Для решения этой задачи ученик написал программу, но, к сожалению, его программа неправильная.

24-3 Программа

```
var n, k: integer;  
begin  
    read(n);  
    k := 1;  
    while n>0 do  
        begin  
            n := n-k;    k := k+1;  
        end;  
    writeln(k)  
end.
```

24-3 Задание

Последовательно выполните следующее.

1. Приведите пример числа N , при вводе которого программа выведет неверный ответ. Укажите верный ответ и ответ, который выведет программа.

2. Приведите пример числа N , при вводе которого программа выведет верный ответ. Укажите этот ответ.

3. Найдите в программе все ошибки (их может быть одна или несколько). Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

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

Обратите внимание: Вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

24-3 Решение

- 1. Примеры чисел, при вводе которых программа выводит неверный ответ:

Значение N	Верный ответ	Ответ программы
2	2	3
4	3	4
5	3	4
7	4	5

24-3 Решение

2. Примеры чисел, при вводе которых программа выводит верный ответ:

Значение N	Верный ответ	Ответ программы
1	2	2
3	3	3
6	4	4
10	5	5

Заметим, что программа выдаёт верный ответ для тех значений N, которые можно представить в виде суммы $1 + 2 + \dots + K$. При $K = 1, 2, 3, 4$ получим примеры, приведённые в таблице. Во всех остальных случаях программа выдаёт неверный ответ.

24-3 Решение

3. Программа содержит две ошибки:

1) неверное условие цикла;

2) неверный вывод результата (выводится значение, на 1 превышающее верное).

Пример исправления для языка Паскаль:

Первая ошибка:

```
while  $n > 0$  do begin
```

Исправленная строка:

```
while  $n \geq 0$  do begin
```

Вторая ошибка:

```
writeln( $k$ )
```

Исправленная строка:

```
writeln( $k-1$ )
```

Незначительной опiskeй, не влияющей на оценку, следует считать отсутствие служебных слов и знаков после содержательной части исправления.

24-1

- На обработку поступает последовательность из четырёх неотрицательных целых чисел (некоторые числа могут быть одинаковыми). Нужно написать программу, которая выводит на экран количество чётных чисел в исходной последовательности и максимальное чётное число. Если чётных чисел нет, требуется на экран вывести «NO». Известно, что вводимые числа не превышают 1000. Программист написал программу неправильно. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования.

```
const n = 4;
var i, x: integer; maximum, count: integer;
begin
    count := 0; maximum := 1000;
    for i := 1 to n do
        begin
            read(x);
            if x mod 2 = 0 then
                begin
                    count := count + 1;
                    if x > maximum then maximum := i
                end
            end;
        if count > 0 then begin
                writeln(count);    writeln(maximum)
            end
            else    writeln('NO')
        end.
    end.
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности: 2 9 4 3
2. Приведите пример такой последовательности, содержащей хотя бы одно чётное число, что, несмотря на ошибки, приведённая программа печатает правильный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:
 - 1) выпишите строку, в которой сделана ошибка;
 - 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

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

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

Примечание. 0 – чётное число.

Решение

1. Программа выведет 2 1000.
2. Например, набор 2 4 5 1000.
3. Пример исправлений для языка Паскаль

Первая ошибка:

```
maximum := 1000;
```

Исправленная строка: `maximum := -1;`

Вторая ошибка: `maximum := i`

Исправленная строка: `maximum := x`

24-4

Для заданного положительного вещественного числа A необходимо найти максимальное целое число K , при котором выполняется неравенство

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{K} < A.$$

(при $K = 0$ сумма считается равной 0).

Для решения этой задачи ученик написал такую программу.

24-4 Программа

```
var a, s: real; k: integer;
```

```
Begin
```

```
  read(a);
```

```
  k := 1;
```

```
  s := 1;
```

```
  while s < a do
```

```
    begin
```

```
      s := s + 1.0/k;   k := k + 1;
```

```
    end;
```

```
    write(k);
```

```
end.
```

24-4 Задание к программе

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 1.2.
2. Приведите пример числа, при вводе которого программа даст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Обратите внимание: вам нужно исправить приведённую программу, а не написать свою. Вы можете только исправлять ошибочные строки; удалять строки или добавлять новые строки нельзя. Постарайтесь также не внести новые ошибки – за это оценка снижается.

24-4 Решение

Решение использует запись программы на Паскале. Допускается использование программы на других языках.

1. При вводе числа 1.2 программа выведет число 2.
2. Примеры чисел, при вводе которых программа выводит верный ответ: 1.6, 2.05.

Примечание для проверяющего. Программа содержит две ошибки, одна из которых приводит к увеличению ответа, другая – к уменьшению.

В некоторых случаях эти ошибки компенсируют друг друга, и ответ оказывается правильным. Это происходит, если значение A попадает в один из следующих диапазонов: $1.5 < A < 1.83$, $2 < A < 2.08$.

24-4 Решение (продолжение)

3. Программа содержит две ошибки.

1) Неверная инициализация. Начальное значение S должно быть равно нулю. В приведённом варианте вычисленная сумма оказывается на 1 больше правильного значения.

Строка с ошибкой:

```
s := 1;
```

Правильная строка:

```
s := 0;
```

2) Неверное определение ответа. Приведённая программа находит не максимальное K , при котором выполняется неравенство, а минимальное, при котором оно не выполняется, то есть увеличивает верное значение на 1.

Кроме того, использованный порядок действий в цикле (увеличение K после увеличения S) приводит к увеличению ещё на 1. Это можно было бы исправить, изменив порядок действий в цикле и уменьшив K после завершения цикла, но эти действия не разрешены по условию задачи.

Поэтому для исправления ошибки можно просто скорректировать значение при выводе.

Строка с ошибкой:

```
write(k);
```

Правильная строка:

```
write(k-2);
```

25-1 Условие

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\ 000$ до $10\ 000$ включительно. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых сумма элементов делится на 3, но не делится на 9. В данной задаче под парой подразумеваются два соседних элемента массива.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

25-1 Начало программы

```
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.
```

25-1 условие (продолжение)

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 приведённых фрагментах.

25-1 Решение

```
k := 0;
```

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

```
    if ((a[i]+a[i+1]) mod 3=0) and ((a[i]+a[i+1]) mod  
9<>0) then k:=k+1;
```

```
writeln(k);
```


25-2 Условие

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от 0 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 оба числа являются чётными. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 6; 1; 4; 6; 10 – ответ: 2. 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.

25-2 Условие (начало программы)

```
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.
```

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия.

25-2 Решение

```
k := 0;
for i := 1 to N - 1 do
    if (a[i] mod 2 = 0) and (a[i + 1] mod 2 = 0) then
        k:=k+1;
writeln(k);
```

25-3 Условие

Дан массив, содержащий 2017 положительных целых чисел, не превышающих 1000. Необходимо найти и вывести максимальный из тех элементов этого массива, восьмеричная запись которых содержит не менее четырёх цифр и оканчивается цифрой 4. Если таких чисел в массиве нет, ответ считается равным нулю. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных.

25-3 Пример начала программы

```
const    N = 2017;  
var     a: array [1..N] of integer;  
         i, m, k: integer;  
begin  
         for i := 1 to N do  
             readln(a[i]);    ...  
end.
```

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия.

25-3 Решение

```
m := 0;
for i := 1 to N do
  if (a[i] >= 512) and (a[i] mod 8 = 4) and
    (a[i] > m) then m := a[i];
writeln(m);
```

26-1

Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 2, а во второй — 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или утраивает число камней в какой-то куче, или добавляет 4 камня в какую-то кучу.

Игра завершается в тот момент, когда общее число камней в двух кучах становится не менее 31. Если в момент завершения игры общее число камней в двух кучах не менее 40, то выиграл Петя, в противном случае — Ваня. Кто выигрывает при безошибочной игре обоих игроков? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

Выигрывает Ваня.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделённые запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучах соответственно.

	1-й ход	2-й ход	3-й ход	4-й ход	Примечания
Стартовая позиция	Петя (все варианты хода)	Ваня (выигрышные ходы)	Петя (все варианты хода, кроме непосредственно проигрышных)	Ваня (выигрышные ходы, экзаменуемому достаточно указать один из вариантов)	
2, 3	2, 9	<u>6, 9</u>	18, 9	<u>22, 9</u>	
				<u>18, 13</u>	
			10, 9	<u>30, 9</u>	
				<u>10, 27</u>	
			6, 13	<u>18, 13</u>	
				<u>18, 21</u>	
	2, 7	<u>6, 7</u>	18, 7	<u>18, 21</u>	
			6, 21	<u>18, 21</u>	
				<u>6, 25</u>	
			10, 7	<u>30, 7</u>	
				<u>10, 21</u>	
	6, 3	<u>6, 9</u> <u>6, 7</u>	Те же варианты 3—4-го ходов		
				<u>6, 33</u>	
				экзаменуемому достаточно рассмотреть один из вариантов: 6, 9 или 6, 7	

Таблица содержит все возможные варианты ходов первого игрока. Из неё видно, что при любом ходе первого игрока у второго имеется ход, приводящий к победе.

26-2

Два игрока, Петя и Вася, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 2, а во второй — 1 камень. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первым ходит Петя.

Ход состоит в том, что игрок или увеличивает в 3 раза число камней в какой-то куче, или добавляет 3 камня в какую-то кучу.

Выигрывает игрок, после хода которого в одной из куч становится не менее 24 камней. Кто выигрывает при безошибочной игре? Каким должен быть первый ход выигрывающего игрока?

Выигрывает Петя, своим первым ходом он должен увеличить в 3 раза количество камней во второй куче. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучах соответственно.

Позиция после первого хода Пети	Вася (все варианты хода)	Петя (выигрышный ход)	Вася (все варианты хода)	Петя (один из вариантов)	Пояснение
<u>2, 3</u>	5, 3	<u>5, 6</u>	5, 18	<u>5, 54</u>	Петя выигрывает после любого ответа Васи, например, утроив число камней в самой большой куче
			5, 9	<u>5, 27</u>	
			8, 6	<u>24, 6</u>	
			15, 6	<u>45, 6</u>	
	2, 6	<u>5, 6</u>	Те же варианты ходов.		
	6, 3	<u>6, 6</u>	6, 9	<u>6, 27</u>	Петя выигрывает после любого ответа Васи, например, утроив число камней в самой большой куче
			6, 18	<u>6, 54</u>	
	2, 9	<u>2, 27</u>	<u>Петя выиграл</u>		

Таблица содержит все возможные варианты ходов Васи. Из неё видно, что при любом его ответе у Пети имеется ход, приводящий к победе.

26-3

Два игрока играют в следующую игру. На координатной плоскости стоит фишка. Игроки ходят по очереди. В начале игры фишка находится в точке с координатами $(3, 2)$. Ход состоит в том, что игрок перемещает фишку из точки с координатами (x, y) в одну из трёх точек: или в точку с координатами $(x + 3, y)$, или в точку с координатами $(x, y + 2)$, или в точку с координатами $(x, y + 4)$. Выигрывает игрок, после хода которого расстояние по прямой от фишки до точки с координатами $(0, 0)$ больше 12 единиц. Кто выиграет при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Как должен ходить выигрывающий игрок? *Постройте дерево партии для выигрышной стратегии (в виде рисунка или таблицы).*

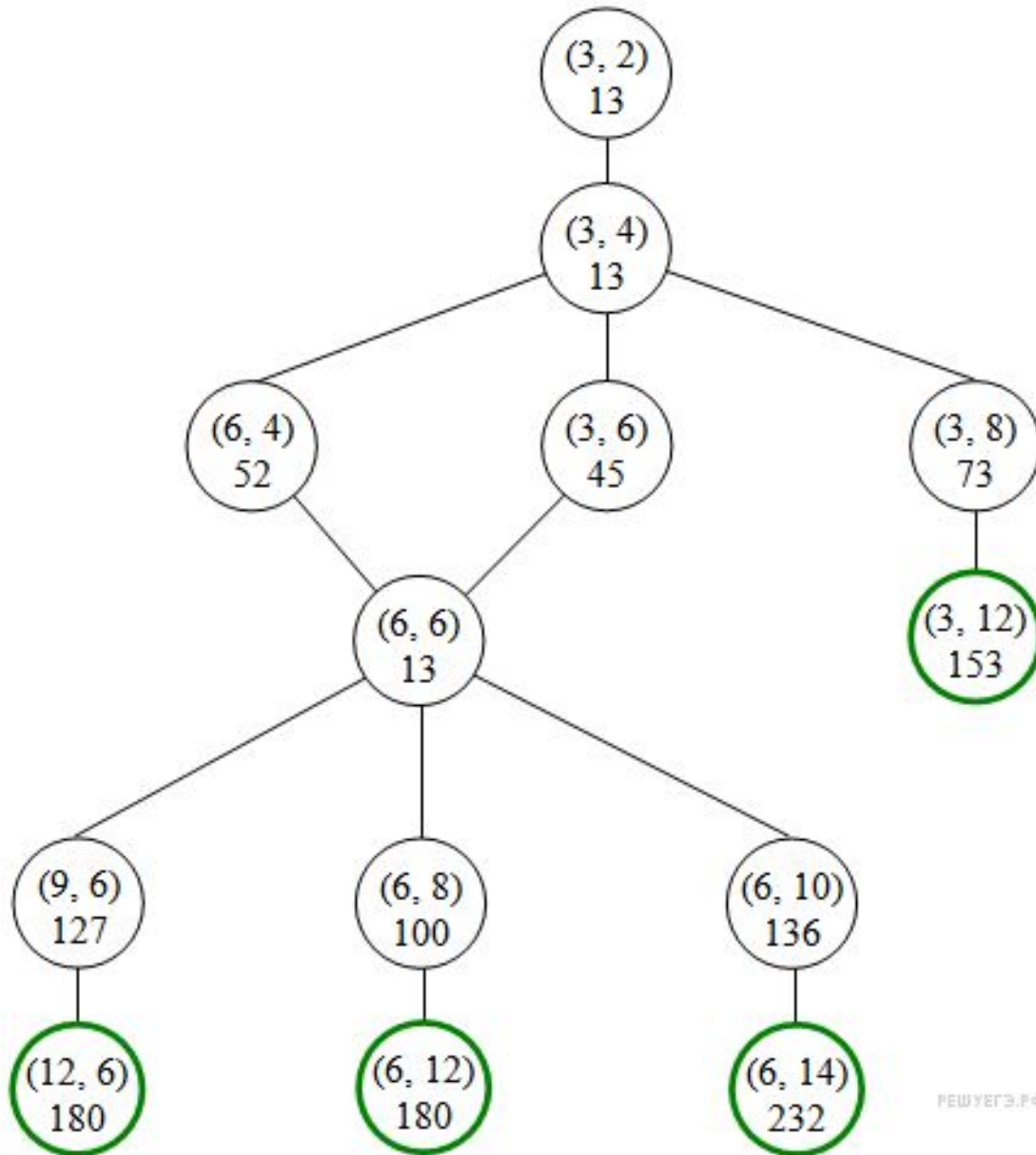
26-3 Решение

Квадрат расстояния от фишки до точки с координатами $(0, 0)$: $r^2 = x^2 + y^2$. Побеждает игрок, после хода которого $r^2 > 144$. Алгоритм выигрышной стратегии определим при помощи дерева всех возможных партий. Не будем приводить здесь полное дерево, отметим лишь, что при ходе первого игрока в точку $(3, 4)$ первый игрок при любом ответе противника имеет выигрышный набор ходов.

Построим дерево партии для выигрышной стратегии первого игрока: в узлах будем указывать координаты фишки и квадрат расстояния до начала координат. Зелёным отмечены позиции, в которых выигрывает первый игрок.

Дерево содержит все возможные варианты ходов второго игрока. Из него видно, что при любом ответе второго игрока у первого имеется ход, приводящий к победе.

26-3Решение



27-1 Условие

На спутнике «Фотон» установлен прибор, предназначенный для измерения энергии космических лучей. Каждую минуту прибор передаёт по каналу связи неотрицательное вещественное число — количество энергии, полученной за последнюю минуту, измеренное в условных единицах. Временем, в течение которого происходит передача, можно пренебречь. Необходимо найти в заданной серии показаний прибора минимальное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Количество энергии, получаемое прибором за минуту, не превышает 1000 условных единиц. Общее количество показаний прибора в серии не превышает 10 000. Напишите на любом языке программирования программу для решения поставленной задачи.

27-1 Условие (продолжение)

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

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

Задание Б является усложненным вариантом задания А, оно содержит дополнительные требования к программе. Перед программой укажите версию языка программирования.

27-1 Условие (продолжение)

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

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

Максимальная оценка за выполнение задания **А** – 2 балла.

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

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

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

27-1 Условие (продолжение)

Перед программой укажите версию языка и кратко опишите использованный алгоритм. В первой строке задаётся число N — общее количество показаний прибора. Гарантируется, что $N > 6$. В каждой из следующих N строк задаётся одно неотрицательное вещественное число — очередное показание прибора.

Пример входных данных:

```
11
12
45
5
4
25
23
21
20
10
12
26
```

Программа должна вывести одно число — описанное в условии произведение.
Пример выходных данных для приведённого выше примера входных данных:

```
48
```

27-1 Решение

Для построения программы, эффективной по времени, можно определить для каждого элемента входных данных минимальное значение от начала данных до этого элемента включительно. Затем нужно умножать каждый элемент, начиная с седьмого, на значение этого минимума, взятого на шесть элементов раньше, и выбрать наименьшее из этих произведений. Предложенный алгоритм реализован в следующей программе на алгоритмическом языке

```

program c4;
  const s = 6; {требуемое расстояние между показаниями}
Var  N: integer;
     a: array[0..s - 1] of real; {хранение показаний прибора}
     {k-е введенное число записываем в ячейку a[k mod 6]}
     a_: real; {ввод очередного показания}
     mn: real; {минимальное введенное число}
           {не считая 6 последних}
     m: real; {минимальное значение произведения}
     i: integer;
begin
  readln(N);
  { Пролог. Ввод первых шести чисел}
  for i:=1 to s do
    begin
      readln(a_);
      a[i mod s] := a_
    end;
  {Ввод остальных значений, поиск минимального произведения}
  mn := 1001; m := 1000 * 1000+1;
  for i := s + 1 to N do
    begin
      readln(a_);
      if a[i mod s] < mn then mn := a[i mod s];
      if a_ * mn < m then m := a_ * mn;
      a[i mod s] := a_
    end;
  writeln(m)
end.

```

27-1

Решение

Успеха, Вам на экзамене!