

ЗАДАНИЕ 27 (ЕГЭ-2019 г.)

На вход программы поступает последовательность из N целых положительных чисел, все числа в последовательности различны. Рассматриваются все пары различных элементов последовательности, находящихся на расстоянии не меньше чем 3 (разница в индексах элементов пары должна быть 3 или более, порядок элементов в паре неважен). Необходимо определить количество таких пар, для которых произведение элементов делится на 23.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($3 \leq N \leq 1000$). В каждой из последующих N строк записано одно целое положительное число, не превышающее 10 000.

В качестве результата программа должна вывести одно число: количество пар элементов, находящихся в последовательности на расстоянии не меньше чем 3, в которых произведение элементов кратно 23.

Пример входных данных:

6
46
2
3
5
4
23

Пример выходных данных для приведённого выше примера входных данных:

5

Пояснение. Из шести заданных элементов с учётом допустимых расстояний между ними можно составить 6 произведений: $46 \cdot 5$, $46 \cdot 4$, $46 \cdot 23$, $2 \cdot 4$, $2 \cdot 23$, $3 \cdot 23$. Из них на 23 делятся 5 произведений.

Требуется написать эффективную по времени и памяти программу для решения описанной задачи.

Решение на 2 балла

Программа **неэффективная** по времени и памяти

```
Const d = 3;  
var i,j, n,k: integer;  
x: array [1..10000] of integer;  
begin  
  writeln ('введите размер массива ');  
  readln (n);  
  for i:=1 to n do  
    readln (x[i]);  
  k:=0;  
  for i:=1 to n-d do  
    for j:=i+d to n do  
      if ((x[i]*x[j]) mod 23=0) then k:=k+1;  
  writeln (k);  
end.
```

Решение на 4 балла

Запоминаем первые три числа в массиве.

Подсчитываем **k23** количество чисел, кратных 23;

k1 количество чисел, некратных 23, но которые могут образовать пару с числами, кратными 23 и отстоящими от него не менее, чем на 3;

k число пар, произведение которых кратно 23, находящихся на расстоянии не меньше чем 3

```
Const d = 3;
```

```
Var
```

```
a:array[1..d] of integer;
```

```
N,k23,k1,x,k,j,i:integer;
```

```
begin
```

```
readln(N);
```

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

```
readln(a[i]);
```

```
for i:=d+1 to N do
```

```
begin
```

```
if(a[1] mod 23 = 0) then inc(k23)
```

```
else inc(k1);
```

```
readln(x);
```

```
if(x mod 23 = 0) then k := k + k23 + k1
```

```
else k:=k+k23;
```

```
for j:=1 to d-1 do {сдвиг массива влево}
```

```
begin
```

```
a[j] := a[j+1];
```

```
end;
```

```
a[d] := x;
```

```
end;
```

```
writeln(k);
```

```
end.
```

Задание 27. Вариант 1.

На вход программе (как вариант, из входного файла text.dat) подается текст на английском языке, заканчивающийся точкой (другие символы «.» в этом файле отсутствуют). Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая будет определять и выводить на экран, какая английская буква встречается в этом тексте чаще всего и сколько именно раз. Строчные и прописные буквы при этом считаются неразличимыми. Если таких букв несколько, то программа должна выводить на экран ту из них, которая стоит по алфавиту раньше. Например, пусть файл содержит следующую информацию:

It is not a simple task. Yes!

Тогда чаще всего встречаются буквы I, S и T (слово Yes в подсчете не участвует, так как расположено после точки). Следовательно, в данном случае программа должна вывести

I 3

Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

<p>Программа читает текст до точки один раз, подсчитывая в массиве, хранящем 26 целых чисел, количество вхождений каждой из букв. Сам текст при этом не запоминается. Затем в этом массиве ищется первое вхождение максимального элемента. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, для строк, состоящих не более чем из 255 символов).</p>	
Указания по оцениванию	
<p>Программа работает верно, т. е. определяет первую по алфавиту из наиболее часто встречающихся букв вместе с количеством таких букв, для любых входных данных произвольного размера, просматривает входные данные один раз, не содержит вложенных циклов, в тексте программы не анализируется каждая английская буква в отдельности.</p> <p>Допускается наличие одной синтаксической ошибки.</p> <p style="text-align: center;">Примеры правильных и эффективных программ</p>	<p>Баллы</p> <p>4</p>
На языке Паскаль	На языке Бейсик
<pre>var a:array['A'..'Z'] of integer; c, cmax: char; begin assign(input,'text.dat'); reset(input); for c:='A' to 'Z' do a[c]:=0; repeat read(c); c:=upcase(c); if c in ['A'..'Z'] then a[c]:=a[c]+1 until c='.'; cmax := 'A'; for c:='B' to 'Z' do if a[c] > a[cmax] then cmax := c; writeln(cmax,' ',a[cmax]) end.</pre>	<pre>DIM i, imax, c, a(26) AS INTEGER OPEN "TEXT.DAT" FOR INPUT AS #1 S\$ = INPUT\$(1, #1) DO WHILE NOT (S\$ = ".") c = ASC(S\$) IF(c>=ASC("A")AND c<=ASC("Z")) THEN c = c - ASC("A") + 1 ENDIF IF(c>=ASC("a")AND c<=ASC("z")) THEN c = c - ASC("a") + 1 ENDIF IF(c>=1 AND c<=26) THEN a(c)=a(c)+1 S\$ = INPUT\$(1, #1) LOOP imax = 1 FOR i = 2 TO 26 IF a(i) > a(imax) THEN imax = i NEXT i PRINT CHR\$(imax + 64), a(imax) END</pre>
<p>Программа составлена верно, но содержит нерациональности: входные данные запоминаются в массиве символов или строке или входной поток просматривается несколько раз, программа может содержать вложенные циклы.</p> <p>Допускается наличие не более трех синтаксических ошибок.</p>	<p>3</p>
<p>Программа составлена в целом верно с вложенными циклами или без, или обрабатывает каждую букву явным образом (26 или 52 оператора IF или оператор CASE, содержащий 26 или 52 вариантов), но, возможно, выводит значение не первой по алфавиту из искомым букв. Возможно в реализации алгоритма содержатся 1–2 ошибки (используется знак «<» вместо «>», «or» вместо «and» и т. п.). Возможно, некорректно организована работа со входным файлом. Допускается наличие не более пяти синтаксических ошибок.</p>	<p>2</p>
<p>Программа, возможно, неверно работает при некоторых входных данных, но по приведенному тексту решения ясно, что экзаменуемый понимает, из каких этапов должно состоять решение задачи.</p> <p>Программа, возможно, неверно обрабатывает некоторые входные данные, например, отсутствует или предложен некорректный алгоритм обработки строчных или прописных букв, или они подсчитываются по отдельности, или программа содержит ошибку в алгоритме поиска максимума. Возможно, выводит только искомую букву и не выводит количество букв. Допускается наличие не более семи синтаксических ошибок.</p>	<p>1</p>

Задание не выполнено или выполнено неверно	0
<i>Максимальный балл</i>	4

Задание 27(С4). Вариант 2.

На вход программе подаются 366 строк, которые содержат информацию о среднесуточной температуре всех дней 2008 года. Формат каждой из строк следующий: сначала записана дата в виде dd.mm (на запись номера дня и номера месяца в числовом формате отводится строго два символа, день от месяца отделен точкой), затем через пробел записано значение температуры – число со знаком плюс или минус, с точностью до 1 цифры после десятичной точки. Данная информация отсортирована по значению температуры, то есть хронологический порядок нарушен. Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая будет выводить на экран информацию о месяце (месяцах) среднемесячная температура для которого (которых) наименее отклоняется от среднегодовой. В первой строке вывести среднегодовую температуру. Найденные значения для каждого из месяцев следует выводить в отдельной строке в следующем виде: номер месяца, значение среднемесячной температуры, отклонение от среднегодовой температуры.

Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)
Программа читает входные данные, сразу подсчитывая в массиве, хранящем 12 вещественных чисел, сумму температур в каждом из месяцев, одновременно суммируя все температуры в году. Затем с использованием этого массива ищется минимальное отклонение среднемесячной температуры от среднегодовой. За дополнительный просмотр этого массива распечатывается информация об искомым месяцах. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, месяц с минимальным отклонением единствен).

Указания по оцениванию	Баллы
<p>Программа работает верно, т. е. определяет все месяцы, в которых среднемесячная температура минимально отклоняется от среднегодовой, не содержит вложенных циклов, в тексте программы не анализируется каждый месяц в отдельности. Допускается наличие в тексте программы одной синтаксической ошибки.</p> <p>Пример правильной и эффективной программы на языке Паскаль:</p> <pre> Const d:array[1..12] of integer = (31,29,31,30,31,30,31,31,30,31,30,31); var tm:array[1..12] of real; m:1..12; data:string[5]; min,ty,t:real; i:integer; begin for i:=1 to 12 do tm[i]:=0; ty:=0;{среднегодовая температура} for i:=1 to 366 do begin readln(data,t); {вычисляем номер месяца} m:=(ord(data[4])-ord('0'))*10 +ord(data[5])-ord('0'); tm[m]:=tm[m]+t; ty:=ty+t; end; for i:=1 to 12 do tm[i]:=tm[i]/d[i]; ty:=ty/366; min:=100; for i:=1 to 12 do if abs(tm[i]-ty)<min then min:=abs(tm[i]-ty); writeln('Среднегодовая температура = ',ty:0:2); for i:=1 to 12 do if abs(abs(tm[i]-ty)-min)<0.0001 then writeln(i,' ',tm[i]:0:2,' ',abs(tm[i]-ty):0:2); readln end. </pre> <p>Пример правильной и эффективной программы на языке Бейсик:</p> <pre> DATA 31,29,31,30,31,30,31,31,30,31,30,31 DIM i, m, d(12) AS INTEGER DIM tm(12) DIM dat AS STRING *5 FOR i = 1 TO 12 tm(i) = 0 READ d(i) NEXT i ty = 0 'ty – среднегодовая температура FOR i = 1 TO 366 INPUT dat, t </pre>	4

<pre> 'Вычисляем номер месяца m=(ASC(MID\$(dat,4,1))-ASC("0"))*10+ ASC(MID\$(dat,5,1))-ASC("0") tm(m) = tm(m) + t ty = ty + t NEXT i FOR i = 1 TO 12 tm(i) = tm(i) / d(i) NEXT i ty = ty / 366 min = 100 FOR i = 1 TO 12 IF ABS(tm(i) - ty) < min THEN min = ABS(tm(i) - ty) NEXT i PRINT "Среднегодовая температура = "; PRINT USING "##.##"; ty FOR i = 1 TO 12 IF ABS(ABS(tm(i) - ty) - min) < .0001 THEN PRINT i; PRINT USING "##.## "; tm(i); ABS(tm(i) - ty) END IF NEXT i END </pre>	
<p>Программа составлена верно, но содержит вложенные циклы (от 1 до 12 и от 1 до 366) и, возможно, запоминает значения всех температур в массиве. Допускается наличие от одной до трех синтаксических ошибок.</p>	3
<p>Программа составлена в целом верно с вложенными циклами или без них, или обрабатывает каждый месяц явным образом (12 операторов IF или оператор CASE, содержащий 12 вариантов), или предварительно сортирует входные данные в хронологическом порядке. Возможно, выводит значение только одного месяца с минимальным отклонением температуры. Возможно в реализации алгоритма содержатся 1–2 ошибки (используется знак «<» вместо «>», «or» вместо «and» и т.п.). Допускается наличие не более пяти синтаксических ошибок.</p>	2
<p>Программа, возможно, неверно работает при некоторых входных данных, но по приведенному тексту решения ясно, что экзаменуемый понимает, из каких этапов должно состоять решение задачи. Программа, возможно, некорректно определяет номер месяца или неверно вводит или выводит данные, или содержит ошибку в алгоритме поиска минимума или средней температуры, или отклонение берется не по абсолютной величине. Допускается наличие не более семи синтаксических ошибок.</p>	1
<p>Задание не выполнено или выполнено неверно.</p>	0
<p><i>Максимальный балл</i></p>	4

