С4. Программа верно читает входные данные, не запоминая их все, а сразу подсчитывая в массиве, хранящем 99 целых чисел согласно номерам школ, количество участников олимпиады из каждой школы. Затем подсчитывается количество школ, приславших хотя бы одного участника, и вычисляется среднее количество участников от одной школы.

```
var nc:array[1..99] of integer;
 p:1..99;
 c:char;
 i, k, N: integer;
begin
 readln(N);
 for i:=1 to 99 do nc[i]:=0;
 for i:=1 to N do
 begin
  repeat
   read(c)
  until c=' '; {считана фамилия}
  repeat
   read(c)
  until c=' '; {считаны инициалы}
  readln(p);
  nc[p]:=nc[p]+1;
 end;
 k:=0;
 for i:=1 to 99 do
  if nc[i]>0 then k:=k+1;
 writeln('Среднее количество участников из одной школы', N/k)
end.
```

Вариант 5

С1. 1) $a = -1$, $b = 0$, $x = 0$.
Значение x может быть не указано. Значение a может быть любым отрицательным числом. Также допустим ответ, что программа работает неправильно при любых отрицательных a и $b = 0$).
2) Лишняя часть:
не нужно вводить x с клавиатуры;
верно: readln(a, b).
3) Возможная доработка:

```
readln(a, b);
if b = 0 then
if a > 0 then
write('x > 0 или x < 0')
else
write('нет решений')
else
if a > 0 then
write('x > 0 или x <', -b)
else
write(-b, '< x < 0');
```

(могут быть и другие способы доработки).

С2. Введем целочисленную переменную SumNeg и целочисленную переменную NumNeg, в которые будем заносить соответственно сумму и число отрицательных элементов в просмотренной части массива, и присвоим им значение 0. В цикле до конца массива: проверяем, является ли очередной элемент отрицательным. Если да, то прибавляем его к

SumNeg и увеличиваем счетчик NumNeg на единицу. По окончании цикла выводим SumNeg/NumNeg. С4.

Пример правильной и эффективной программы (на основе алгоритма, использующего однократный проход по массиву):

На языке Паскаль	На языке Бейсик

Const N = 30;
Var a:array [1..N] of integer;
SumNeg, NumNeg, I: integer;
begin
SumNeg :=0;
NumNeg :=0;
for I := 1 to N do
if a[I]<0 then
begin
SumNeg := SumNeg + a[I];
NumNeg := NumNeg + 1;
end;
writeln (SumNeg/NumNeg);
end.
	N = 30
	DIM I, SumNeg, NumNeg, A(N) AS INTEGER
	SumNeg=0
	NumNeg=0
	FOR I = 1 TO N
	IF A(I)<0 THEN
	SumNeg = SumNeg + A(I)
	NumNeg = NumNeg + 1
	ENDIF
	NEXT I
	PRINT SumNeg/NumNeg
	END

С3. Выигрывает второй игрок.

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.

<table>
<tr><th></th><th>1 ход</th><th>2 ход</th><th>3 ход</th><th>4 ход</th><th></th></tr>
<tr><th>Стартовая позиция</th><th>I-й игрок (все варианты хода)</th><th>II-й игрок (выигрышный ход)</th><th>I-й игрок (все варианты хода)</th><th>II-й игрок (один из вариантов)</th><th>Пояснение</th></tr>
<tr><td rowspan="7">6,5</td><td rowspan="4">12,5</td><td rowspan="4"><u>12,10</u></td><td>24,10</td><td><u>72,10</u></td><td rowspan="4">Второй игрок выигрывает на четвертом ходу, после любого ответа первого игрока, например, утроив число камней в самой большой куче</td></tr>
<tr><td>36,10</td><td><u>108,10</u></td></tr>
<tr><td>12,20</td><td><u>12,60</u></td></tr>
<tr><td>12,30</td><td><u>12,90</u></td></tr>
<tr><td>6,10</td><td><u>12,10</u></td><td colspan="3">Те же варианты третьего-четвертого ходов</td></tr>
<tr><td>18,5</td><td><u>54,5</u></td><td colspan="3">Второй игрок выигрывает ответным ходом</td></tr>
<tr><td>6,15</td><td><u>6,45</u></td><td colspan="3">Второй игрок выигрывает ответным ходом</td></tr>
</table>

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

С4. Программа считывает входные данные, сразу подсчитывая в массиве, хранящем 12 вещественных чисел, сумму температур в каждом из месяцев. Затем с использованием этого массива ищется максимальная среднемесячная температура. За дополнительный просмотр среднемесячных температур (их можно как запомнить в массиве, так и вычислить заново) распечатывается информация об искомых месяцах. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, месяц с максимальной температурой единственен).

Пример правильной и эффективной программы на языке Паскаль:

```
const d:array[1..12] of integer =
 (31,28,31,30,31,30,31,31,30,31,30,31);
var m:array[1..12] of real;
 max,t:real;
 i,j:integer;
 c1,c2:char;
begin
 for j:=1 to 12 do
  m[j]:=0;
 for i:=1 to 365 do
 begin
  readln(c1,c1,c1,c1,c2,t);
  j:=(ord(c1)-ord('0'))*10+
   ord(c2)-ord('0');
 m[j]:=m[j]+t
 end;
 max:=m[1]/d[1];
 for j:=2 to 12 do
  if m[j]/d[j] > max then
   max:=m[j]/d[j];
 for j:=1 to 12 do
  if abs(m[j]/d[j]-max) < 0.0001
  then writeln(j,' ',m[j]/d[j]:0:1)
end.
```

Пример правильной программы на языке Бейсик:

```
DATA 31,28,31,30,31,30,31,31,30,31,30,31
DIM i, j, d(12) AS INTEGER
DIM m(12)
DIM dat AS STRING * 5
FOR i = 1 TO 12
 m(i) = 0
 READ d(i)
NEXT i
FOR i = 1 TO 365
 INPUT dat, t
 j = (ASC(MID$(dat, 4, 1)) - ASC("0")) * 10 +
  ASC(MID$(dat, 5, 1)) - ASC("0")
 m(j) = m(j) + t
NEXT i
max = m(1) / d(1)
FOR j = 2 TO 12
 IF m(j) / d(j) > max THEN max = m(j) / d(j)
NEXT j
FOR j = 1 TO 12
 IF ABS(m(j) / d(j) - max) < .0001 THEN
  PRINT j; " ";
  PRINT USING "##.#"; m(j) / d(j)
 ENDIF
NEXT j
END
```