

Вариант 7

C1. 1) $a = -1, b = 1, x = 0$.

Значение x может быть не указано. Значение a может быть любым отрицательным числом, значение b — любым положительным. Также допустим ответ, что программа работает неправильно при любых положительных b и отрицательных a .

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 > ', a, ' или x < 0')
else
write('x < ', a, ' или x > 0')
else
if a > 0 then
write('0 < x < ', a)
else
write(a, ' < x < 0');
```

(могут быть и другие способы доработки).

C2. *Пример правильного описания алгоритма на русском языке.*

Заводим переменную `MaxIncr` для хранения максимального количества подряд идущих в порядке возрастания элементов и счетчик `NumIncr` для хранения числа элементов в последней группе возрастающих элементов. Просматривая элементы массива, сравниваем очередной элемент со следующим за ним. Если очередной элемент массива оказывается не меньше следующего, то сравниваем текущее значение счетчика со значением переменной `MaxIncr`; если он больше, то заменяем значение переменной `MaxIncr` значением счетчика, при этом значение `NumIncr` обнуляется. Так повторяем до конца массива. В конце работы нужно еще раз сравнить значение счетчика со значением переменной `MaxIncr` и переопределить ее, если счетчик больше.

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

| На языке Паскаль | На языке Бейсик |
|--|--|
| <pre>const N = 30; var a:array[1..N] of integer; MaxIncr, NumIncr, i: integer; begin MaxIncr:=0; NumIncr:=0; for i:=1 to N-1 do begin if a[i]<a[i+1] then NumIncr:=NumIncr+1; else begin if NumIncr> MaxIncr then MaxIncr:=NumIncr; NumIncr:=0; end; end; if NumIncr> MaxIncr then MaxIncr:=NumIncr; writeln(MaxIncr); end.</pre> | <pre>N = 30 DIM i, MaxIncr, NumIncr, a(N) AS INTEGER MaxIncr=0 NumIncr=0 FOR i = 1 TO N-1 IF a(i)<a(i+1) THEN NumIncr=NumIncr+1 ELSE IF NumIncr>MaxIncr THEN MaxIncr=NumIncr ENDIF NumIncr=0 ENDIF NEXT i IF NumIncr>MaxIncr THEN MaxIncr=NumIncr ENDIF PRINT MaxIncr END</pre> |

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.

| | 1 ход | 2 ход | 3 ход | 4 ход | |
|-------------------|-------------------------------|-----------------------------|--|--------------------------------|---|
| Стартовая позиция | I-й игрок (все варианты хода) | II-й игрок (выигрышный ход) | I-й игрок (все варианты хода) | II-й игрок (один из вариантов) | Пояснение |
| 1,2 | 3,2 | 3,6 | 9,6 | 27,6 | Второй игрок выигрывает на четвертом ходу, после любого ответа первого игрока, например, утроив число камней в самой большой куче |
| | | | 7,6 | 21,6 | |
| | | | 3,18 | 3,54 | |
| | | | 3,10 | 3,30 | |
| | 5,2 | 5,6 | 15,6 | 45,6 | Второй игрок выигрывает на четвертом ходу после любого ответа первого игрока, например, утроив число камней в самой большой куче |
| | | | 9,6 | 27,6 | |
| | | | 5,18 | 5,54 | |
| | | | 5,10 | 5,30 | |
| | 1,6 | 3,6 или 5,6 | Те же варианты третьего-четвертого ходов | | |

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 должна верно читать входные данные, не запоминая их все, а сразу подсчитывая в массиве, хранящем 12 целых чисел, количество учащихся в каждой из параллелей. Затем с использованием этого массива ищется параллель с максимальным числом учеников. За дополнительный просмотр этого массива распечатывается информация об искомым параллелях. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, параллель с максимальным количеством учеников единственна).

Пример правильной и эффективной программы на языке Паскаль:

```

var pc:array[1..12] of integer;
p:1..12;
class:string[3];
c:char;
max, i, N:integer;
begin
  readln(N);
  for i:=1 to 12 do
    pc[i]:=0;
  for i:=1 to N do
    begin
      repeat
        read(c)
      until c=' '; {считана фамилия}
    end
  end

```



```

repeat
  read(c)
  until c=' '; {считано имя}
  readln(class);
  {определяем номер параллели}
  if length(class)=2 then
    p:=ord(class[1])-ord('0') else
    p:=(ord(class[1])-ord('0'))*10+
      ord(class[2])-ord('0');
  pc[p]:=pc[p]+1; {учитываем ученика этой параллели}
end;
max:=0;
for i:=1 to 12 do
  if pc[i]>max then max:=pc[i];
writeln('Максимум учеников в параллели:', max);
for i:=1 to 12 do
  if pc[i]=max then
    write(i, ' ');
  readln
end.

```

Пример правильной программы на языке Бейсик:

```

DIM i, j, p, n, max, pc(12) AS INTEGER
DIM m(12)
DIM s AS STRING
FOR i = 1 TO 12
  pc(i) = 0
NEXT i
INPUT n
FOR j = 1 TO n
  LINE INPUT s
  c$ = MID$(s, 1, 1)
  i = 1
  WHILE NOT (c$ = " ")
    i = i + 1
    c$ = MID$(s, i, 1)
  WEND
  i = i + 1
  c$ = MID$(s, i, 1)
  WHILE NOT (c$ = " ")
    i = i + 1
    c$ = MID$(s, i, 1)
  WEND
  s = MID$(s, i + 1, 3)
  IF MID$(s, 2, 1) >= "0" AND MID$(s, 2, 1) <= "2" THEN
    p = (ASC(MID$(s, 1, 1)) - ASC("0")) * 10 +
      ASC(MID$(s, 2, 1)) - ASC("0")
  ELSE
    p = ASC(MID$(s, 1, 1)) - ASC("0")
  ENDIF
  pc(p) = pc(p) + 1
NEXT j
max = 0
FOR i = 1 TO 12
  IF pc(i) > max THEN max = pc(i)
NEXT i
PRINT "Max = "; max
FOR i = 1 TO 12
  IF pc(i) = max THEN PRINT i; " ";
NEXT i
END

```