Вариант 6

С1. 1) $a = 1$, $b = -1$, $x = 0$.

Значение x может быть не указано. Значения a и b могут быть любыми ненулевыми числами с разными знаками. Ошибка программиста состоит в том, что программа работает неправильно при любых ненулевых a и b, имеющих разные знаки.

2) Лишняя часть:

не нужно вводить x с клавиатуры;

верно: readln(a, b).

3) Возможная доработка:

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

(могут быть и другие способы доработки).

С2. *Пример правильного описания алгоритма на русском языке.*

Заводим переменную MaxNeg для хранения максимального количества подряд идущих отрицательных элементов и счетчик NumNeg для хранения числа отрицательных элементов в последней группе отрицательных элементов. Просматривая элементы массива, сравниваем очередной элемент с 0. Если очередной элемент массива оказывается неотрицательным, то сравниваем текущее значение счетчика NumNeg со значением переменной MaxNeg; если он больше, то заменяем значение переменной MaxNeg значением счетчика, при этом значение NumNeg обнуляется. Так повторяем до конца массива. В конце работы нужно еще раз сравнить значение счетчика со значением переменной MaxNeg 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

const N=30;
var a:array[1..N] of integer;
MaxNeg, NumNeg, i: integer;
begin
  MaxNeg:=0;
  NumNeg:=0;
  for i:=1 to N do
  begin
  if a[i]<0 then NumNeg:=NumNeg+1
  else begin
  if NumNeg> MaxNeg then
  MaxNeg:=NumNeg;
  NumNeg:=0;
  end;
  end;
  if NumNeg> MaxNeg then
  MaxNeg:=NumNeg;
  writeln(MaxNeg);
end.
	N=30
	DIM i, MaxNeg, NumNeg, a(N) AS INTEGER
	MaxNeg=0
	NumNeg=0
	FOR i = 1 TO N
	IF a(i)<0 THEN
	  NumNeg=NumNeg+1
	  ELSE
	  IF NumNeg>MaxNeg THEN
	  MaxNeg=NumNeg
	  ENDIF
	  NumNeg=0
	ENDIF
	NEXT i
	IF NumNeg>MaxNeg THEN
	  MaxNeg=NumNeg
	ENDIF
	PRINT MaxNeg
	END

C3. 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**
2,3	4,3	4,6	8,6	24,6	Второй игрок выигрывает на четвертом ходу, после любого ответа первого игрока, например, утроив число камней в самой большой куче
			12,6	36,6	
			4,12	4,36	
			4,18	4,54	
	6,3	6,6	12,6	36,6	Второй игрок выигрывает на четвертом ходу, после любого ответа первого игрока, например, утроив число камней в самой большой куче
			18,6	54,6	
			6,12	6,36	
			6,18	6,54	
	2,6	6,6	Те же варианты третьего-четвертого ходов		
	2,9	2,27	Второй игрок выигрывает ответным ходом		

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

C4. Программа читает входные данные, сразу подсчитывая минимальную длину встречающихся слов. За второй проход исходных данных производится замена букв латинского алфавита и печать расшифрованного сообщения.

Пример правильной и эффективной программы на языке Паскаль:

```
var f:boolean;
 i, k, min: integer;
 c,cnew:char;
 s:string;
begin
 s:='';
 min:=250; k:=0;
 f:=false;
 repeat
  read(c);
  s:=s+c;
  if f then {слово началось}
   if c in ['a'..'z','A'..'Z']
    then k:=k+1
    else begin
     if k<min then min:=k;
      f:=false
    end
```

```
    else {f=false}
     if c in ['a'..'z','A'..'Z']
      then begin f:=true; k:=1 end
    until c='.';
    for i:=1 to length(s) do
    begin
     cnew:=chr(ord(s[i])+min);
     case s[i] of
     'a'..'z':if cnew>'z' then write(chr(ord(cnew)-26))
       else write(cnew);
    'A'..'Z':if cnew>'Z' then write(chr(ord(cnew)-26))
       else write(cnew);
      else write(s[i])
     end;
    end;
    readln
   end.
```

C1.

Пример правильной программы на языке Бейсик:

```
DIM i, j, min, k, f, a(26) AS INTEGER
DIM s AS STRING
INPUT s
i = 1
k = 0
min = 250
f = 0
WHILE NOT (MID$(s, i, 1) = ".")
 c$ = MID$(s, i, 1)
 IF f = 1 THEN
  IF (c$ >= "A") AND (c$ <= "Z") OR
   (c$ >= "a") AND (c$ <= "z") THEN
  k = k + 1
 ELSE IF k < min THEN min = k
   f = 0
  ENDIF
 ELSE
  IF (c$ >= "A") AND (c$ <= "Z") OR
   (c$ >= "a") AND (c$ <= "z") THEN
  f = 1: k = 1
  ENDIF
 ENDIF
 i = i + 1
WEND
IF k < min THEN min = k
FOR j = 1 TO i
 cnew$ = CHR$(ASC(MID$(s, j, 1)) + min)
 IF (MID$(s, j, 1) >= "a") AND (MID$(s, j, 1) <= "z") THEN
  IF cnew$ > "z" THEN
  PRINT (CHR$(ASC(cnew$) - 26));
  ELSE PRINT cnew$;
  ENDIF
 ELSE
 IF (MID$(s, j, 1) >= "A") AND (MID$(s, j, 1) <= "Z") THEN
  IF cnew$ > "Z" THEN
  PRINT (CHR$(ASC(cnew$) - 26));
  ELSE PRINT cnew$;
  ENDIF
 ELSE PRINT MID$(s, j, 1);
 ENDIF
 ENDIF
NEXT j
END
```

C2.