

декса просматриваемого элемента. Ввод значений в массив описывать не требуется, так как он задан. Присвоим p значение единица. В цикле от первого (нулевого) до последнего (N или $N - 1$) значения индекса совершим следующие действия.

Сравним с нулем значение текущего элемента массива. В случае, если значение текущего элемента меньше нуля, умножим текущее значение переменной p на значение рассматриваемого элемента массива.

По завершении цикла выводим значение переменной p .

Примеры записи алгоритма на языках программирования:

Бейсик	Паскаль	Си
<pre>N=30 DIM A(N), P, I AS INTEGER P = 1 FOR I=1 TO N IF A(I) < 0 THEN P = P * A(I) ENDIF NEXT I PRINT P END</pre>	<pre>const N=30; var a: array [1..N] of integer; p, i: integer; begin p := 1; for i:= 1 to N do if a[i] < 0 then p := p * a[i]; end. writeln (p); end.</pre>	<pre>void main(void) { const N=30; int array [N]; int p; p = 1; for (int i=0; i<N; i++) { if (array [i]<0) { p= p * array[i]; } } printf ("%d", p); }</pre>

С3. Выигрывает второй игрок.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, 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)	II-й игрок (выигрышный ход)	I-й игрок (все варианты)	II-й игрок (выигрышный ход)
2,3	4,3	<u>4,6</u>	7,6	<u>14,6</u>
			8,6	<u>16,6</u>
			4,9	<u>4,18</u>
			4,12	<u>4,24</u>
	5,3	<u>5,6</u>	8,6	<u>16,6</u>
			10,6	<u>20,6</u>
			5,9	<u>5,18</u>
			5,12	<u>5,24</u>
	2,6	<u>4,6</u>	Те же варианты ходов, что и в первом случае	

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