IOI’98 – Setubal/Portugal

Problem Portfolio

Party Lamps

1. Introduction to the Problem

The major challenge of this problem is to discover that it is not necessary to "generate" all the possible combinations of buttons presses because there are no more then 8 possible final configuration. The time necessary to solve one problem is constant since it does not depend on the number of buttons presses.

The problem classification, formulated by the Portuguese Scientific Committee, is presented next:

Problem type:	Algorithmic
Problem understanding	Easy
Algorithm effort	Medium
Implementation effort	Easy
Number of possible solutions per test	1
Several levels of testing	Yes

This problem does not allow the contestants to develop only one part of the algorithm, with the only exception that the additional information about the lamps that are known to be on or off may not be programmed in order to pass some of the tests. The evaluation of the solutions of the contestants will focus on the capability to generate all solutions and in the correct processing of the special cases (buttons pressed only one or two times).

2. Algorithms

2.1. Overall description

To solve this problem it is necessary to "discover" that:

1. the order buttons are manipulated is not relevant;
2. when a button is pressed twice the consequences of the first movement is "erased".

Taking in account the previous rules, one can find out that there are 16 (2^4 — four buttons, and two possibilities: button pressed or not pressed) possible combinations of the buttons. For example, when no button is pressed, or all the buttons were pressed an even number of times, all lamps are on, the initial state. But if only the button that changes the state of all the even lamps is pressed an odd number of times then all the even lamps are on and all the odd lamps are off. Following this line of reasoning we obtain the table 1

Table 1. Combinations of the buttons

Button 1	**Button 2**	**Button 3**	**Button 4**	**# movements**	**Status of the lamps**
0	0	0	0	0	all on
0	0	1	0	1	even on; odd off
0	1	0	0	1	even off; odd on
0	1	1	0	2	all off
1	0	0	0	1	all off
1	0	1	0	2	even off; odd on
1	1	0	0	2	even on; odd off
1	1	1	0	3	all on
0	0	0	1	1	all on but (3k+1) are off
0	0	1	1	2	even on but (3k+1) are off; odd off but (3k+1) are on
0	1	0	1	2	even off; odd on but (3k+1) are off but (3k+1) are on
0	1	1	1	3	all off but (3k+1) are on
1	0	0	1	2	all off but (3k+1) are on
1	0	1	1	3	even off but (3k+1) are on; odd on but (3k+1) are off
1	1	0	1	3	even on but (3k+1) are off; odd off but (3k+1) are on
1	1	1	1	4	all on but (3k+1) are off but (3k+1) are on

From the previous table we can conclude that there are only 8 possibilities, since the last column is common to two lines. So we will build the table 2, where all the lamps will be associated with a type:

Table 2. Lamp types

Type	**Meaning**
1	lamp is even and is not successor of 3k
2	lamp is even and is successor of 3k
3	lamp is odd and is not successor of 3k
4	lamp is odd and is successor of 3k

The eight possible combinations can be expressed by table 3, where a "1" means that a lamp is on, and a "0" means that a lamp is off:

Table 3. Lamp combinations

Combination	**Type 1**	**Type 2**	**Type 3**	**Type 4**
1	1	1	1	1
2	1	1	0	0
3	0	0	1	1
4	0	0	0	0
5	1	0	1	0
6	1	0	0	1
7	0	1	1	0
8	0	1	0	1

Using this last table and the column of the first table that contains the minimum number of times that a button has to be pressed to obtain that combination we can find out that if only one button is pressed the only possible final combinations are 2, 3 4 and 5.

If the counter of buttons presses contains the number two then two possibilities may occur:

1. The same button were pressed twice (combination 1)
2. Two different buttons were pressed (combinations 2, 3, 4,6,7 and 8)

If the counter of buttons presses contains the number three, then all combinations may occur:

1. The same button were pressed three times (combination 2,3,4, or 5)
2. One button is pressed twice (combinations 2, 3, 4, 6, 7, or 8)
3. Three different buttons were pressed (combination 1, 6, 7, or 8)

So, we can conclude that when the counter of buttons presses if greater or equal 3 all combinations may occur.

The information about lamps that are known to be on or off may be used to reject some of the possible combinations.

2.2 Implementation

The C program named "party.C" and the PASCAL program "party.p" implements this algorithm.

3. Testing

1 - To verify that it was taken into account that with one movement in the buttons there are only four possibilities. The last two lines of the input do not give any additional information.

INPUT 1
10 1 -1 -1

OUTPUT 1
0101010101 1010101010 0000000000 0110110110

2 - To verify that it was taken into account that with two movements in the buttons there are only seven out of eight possibilities. Again, no additional information is given.

INPUT 2
20 2 -1 -1

OUTPUT 2
11111111111111111111 01010101010101010101 10101010101010101010 00000000000000000000 11000111000111000111 00111000111000111000 10010010010010010010

3 - To verify that it was taken into account that with three movements in the buttons all possibilities can occur. The last two lines have additional information on lamps of type 1, type 2 and type 4.

INPUT 3
40
3
16 -1
2 25 -1

OUTPUT 3
0011100011100011100011100011100011100011

4 - To verify that it was taken into account that with two movements in the buttons there are only seven out of eight possibilities. No additional information is given to testes the ability to generate and output all the combinations.

INPUT 4
50
4
-1
-1

OUTPUT 4
1111111111111111111111111111111111111111111111111
0101010101010101010101010101010101010101010101010
1010101010101010101010101010101010101010101010101
0000000000000000000000000000000000000000000000000
0110110110110110110110110110110110110110110110110
1100011100011100011100011100011100011100011100011
0011100011100011100011100011100011100011100011100
1001001001001001001001001001001001001001001001001

5 - To verify that no solution of the type "generate the tree of all possibilities" (the ramification factor is four) was tried. The last lines are used to verify if the information about lamps of type 3 is correctly taken into account. It is used information about a lamp with type 2 to get only two possible solutions.

INPUT 5
70
99999
51 -1
4 -1

OUTPUT 5
10101010101010101010101010101010101010101010101010101010101010 10101010
01101101101101101101101101101101101101101101101101101101101101 10110110