Partition the numbers | Problem Code: **PRTITION**

You are given two integers **x** and **N**. Consider all integers between 1 and **N** inclusive, except **x**. We want to partition these integers into two disjoint sets (each integer has to appear in exactly one set) such that the sums of numbers in these sets are equal.

Find one valid partition or determine that it doesn't exist.

Input

- The first line of the input contains a single integer **T** denoting the number of test cases. The description of **T** test cases follows.
- The first and only line of each test case contains two space-separated integers **x** and **N**.

Output

For each test case:

- If there's no way to partition the numbers into two sets with equal sums, print a single line containing the string `"impossible"` (without quotes).
- Otherwise, print a single line containing a string with length **N**.
- The **x**-th character of this string should be **'2'**.
- For each valid **i** ≠ **x**, the **i**-th character of this string should be **'0'** if number **i** should be in the first set or **'1'** if it should be in the second set.

Constraints

- 1 ≤ **T** ≤ 10,000
- 2 ≤ **N** ≤ 1,000,000
- 1 ≤ **x** ≤ **N**
- 1 ≤ sum of **N** in all test cases ≤ 1,000,000

Subtasks

Subtask #1 (20 points): sum of **N** in all test cases ≤ 50

Subtask #2 (80 points): original constraints

Example

Input:

```
3
2 4
5 5
1 2
```

Output:

```
0201
01102
impossible
```

Author:	6★ kingofnumbers (/users/kingofnumbers)
Date Added:	4-01-2018
Time Limit:	1 secs
Source Limit:	50000 Bytes
Languages:	ADA, ASM, BASH, BF, C, CAML, CLOJ, CLPS, CPP 4.3.2, CPP 6.3, CPP14, CS2, D, ERL, FORT, FS, GO, HASK, ICK, ICON, JAVA, JS, kotlin, LISP clisp, LISP sbcl, LUA, NEM, NICE, NODEJS, PAS fpc, PAS gpc, PERL, PERL6, PHP, PIKE, PRLG, PYPY, PYTH, PYTH 3.5, RUBY, rust, SCALA, SCM chicken, SCM guile, SCM qobi, ST, swift, TCL, TEXT, WSPC