

## Problem G. DNA Evolution

Program: dna.(cpp|java)  
Input: dna.in  
Balloon Color: White

Fox Ciel just had a new baby born (Fox Jaro). Fox Ciel has always been interested in genetics and DNA sequences and he would like to apply what he knows on his son! A DNA consists of a sequence of nucleotides. There are four types of nucleotides: A, C, G, and T. DNA sequences are passed from a parent to his children, however each newborn fox gets one more nucleotides than his father (the new nucleotide is attached to the beginning of the sequence). For example, Fox Ciel's DNA is CACAA and his son Jaro's DNA is ACACAA (notice the A, that Jaro acquired more than Ciel, attached to the DNA inherited from Ciel).

Ciel was thinking about his ancestors and how his new born looks like them. He noticed that the similarity of his son and one of his grand parents is proportional to the maximum number of common characters they have as a common prefix and he named it as the similarity coefficient. He wrote down all ancestors' DNA and calculated the similarity coefficient between each one of them and Jaro's DNA (ACACAA), then wrote that down on a note similar to the following table. (see note for a formal definition)

|        |   |
|--------|---|
| ACACAA | 6 |
| CACAA  | 0 |
| ACAA   | 3 |
| CAA    | 0 |
| AA     | 1 |
| A      | 1 |

Rabbit Hanako messed up with Ciel's note and deleted Jaro's DNA sequence. If Ciel finds this out, he'll attack Hanako and eat him! Can you help Hanako restore the DNA sequence as soon as possible before Ciel finds this out?

### Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer **T**, the number of test cases ( $1 \leq T \leq 100$ ). Followed by **T** test cases.

Each test case will be presented on a single line by integers separated by single spaces. The first integer on this line will denote the number of characters of the DNA sequence **N** followed by **N** integers  $a_i$  denoting the similarity coefficient at position  $i$  for all  $0 \leq i < n$  (as shown in the table), where ( $1 \leq N \leq 10^5$ ) and ( $0 \leq a_i \leq 10^5$ ).

### Output

For each test case, print a single line containing **S** which is the DNA sequence. If there are multiple valid sequences, print the least lexicographical one. If there are no valid sequences and Hanako can't restore the sequence, print "Impossible"(without the quotes).

String  $x$  is lexicographically less than string  $y$ , if either  $x$  is a prefix of  $y$  or there exists such  $i$  where ( $1 \leq i \leq \min(|x|, |y|)$ ),  $x_i < y_i$  and for any  $j$  ( $1 \leq j < i$ )  $x_j = y_j$ .

### Examples

| dna.in  | standard output |
|---------|-----------------|
| 3       | A               |
| 1 1     | ACA             |
| 3 3 0 1 | Impossible      |
| 3 3 3 3 |                 |

## Note

Formally, given a string  $S$  (DNA) of length  $n$ , the note will contain an array  $A$  where  $A[i]$  is the length of the longest substring starting from  $S[i]$  which is also a prefix of  $S$ . In other words, the maximum  $k$  such that  $S[j] = S[i + j]$  for all  $0 \leq j < k$ .