Task: BIT
Three Bit Computer

day 3, source file `bit.*`, available memory 32 MB *CPSPC 2004, June 10*

The scientists in the Kingdom of Byteland want to develop a new type of computer, namely the Three Bit Computer (TBC). They predict that the new machine will have the power to solve many hard and still unsolved problems. However, there are some technical difficulties that have to be resolved first. You have been asked to assist the scientists in solving one of them.

The scientists are now working on the initialization procedure for the computer memory. The current version of TBC has n bits of memory numbered from 1 to n. Each bit can have one of three values a, b, c or might be uninitialized. The following memory initialization operations are supported by the computer:

- two consecutive uninitialized bits, i.e., the bits numbered i and $i+1$ for $1 \leq i < n$, can be set to two different values,
- two consecutive bits, one unitialized and one initialized to value x, can be set to two different values not equal to x.

For example, the following initialization procedure is possible for $n = 4$, $uuuu \to uuab \to ucbb \to babb$, where u denotes an uninitialized bit of memory.

Task

Write a program that:

- reads the values to which the the memory has to be initialized,
- checks if the initialization is possible,
- writes the answer to the output file.

Input

The input file `bit.in` can contain several initialization patterns. The first line of the input file contains a single positive integer N, $(1 \leq N \leq 10)$, the number of patterns. The description of the patterns follows. Description of a single pattern consists of two consecutive lines. The first one contains a positive integer l_i, $(1 \leq l_i \leq 100\,000)$, the length of the i-th pattern (i.e., the size of computer's memory). The second line contains a sequence of length l_i consisting of letters a, b, c — the pattern itself.

Output

The output file `bit.out` should have N lines, one for each pattern. The i-th line should consist of a single word `YES`, if the initialization is possible, or `NO` otherwise.

Example

For the input file `bit.in`:

```
2
4
aabb
4
aaab
```

the correct result is the output file `bit.out`:

```
NO
YES
```