Turing Machine Contest (V Ed.)

- If not otherwise specified, input sequences are always assumed to be *non empty*. That is, all input sequences contain at least one symbol.
- Integer numbers are assumed to be greater than or equal to zero, represented in decimal notation (using digits 0,1,...,9) with no leading zeros which are not significant. For example, 0 and 19 are valid decimal numbers, while 0032 must be written as 32.
- When producing a solution, recall to remove from the final tape all symbols that do not make up the answer!

Exercise 1 [Character substitution] (easy)

Write a Turing machine that, given an initial tape containing an arbitrary sequence of `A`s and `B`s, *substitutes* every occurrence of consecutive symbols `AB` with the symbols `CD`.

Examples:

Input tape	Final tape
`AABABBBAABAAABAABAA`	`ACDCDBBACDAACDACDAA`
`BBBBAAA`	`BBBBAAA`
`AABB`	`ACDB`

Exercise 2 [Balanced brackets] (easy)

A sequence of nested square and curly brackets is said to be *balanced* when the sequence (i) is empty; or (ii) has the form $\{\ S\ \}\ T$ or $[\ S\ ]\ T$, with both S and T balanced sequences.

Write a Turing machine that, given an initial tape with a (non empty) sequence of square and curly brackets, terminates leaving on the tape only the sequence `YES` if the initial sequence was balanced and `NO` otherwise.

Examples:

Input tape	Final tape
[]{[]}	`YES`
[{]}	`NO`
[[{}][]{}]	`YES`

Exercise 3 [*Totocalcio*] (easy)

Totocalcio is a betting pool based on predicting the outcomes of the 13 top-level football matches taking place in the coming week. A bet B is a sequence of `1`s (home team wins), `2`s (away team wins) and `X`s (draw).

Given W (in the same format) that reports the correct outcomes of the matches, we want to check whether B is *winning*, that is, at least 12 of the predicted outcomes are correct with respect to those in W.

Write a Turing machine that, given a bet B and W separated by a $*$, terminates leaving the sequence `YES` on the tape if B is winning and `NO` otherwise.

Examples:

Input tape	Final tape
1X1X2X21X1X12∗1X1X2X21X1X12	YES
1X1X2X21X1X12∗1X1X2X21X1X21	NO
1X1X2X21X1X12∗1X1X2X21X1112	YES

Exercise 4 [Diving in half] (medium)

Write a Turing machine that, given an initial tape with an even integer N, terminates leaving on the tape the result of diving N by 2.

Examples:

Input tape	Final tape
1234	617
130	65
0	0

Exercise 5 [Doubling a sequence] (easy)

Write a Turing machine that, given an initial tape with a sequence S of `A`s, `B`s and `C`s, terminates leaving on the tape the sequence SS, that is, the original sequence repeated twice.

Examples:

Input tape	Final tape
ABACB	ABACBABACB
ABACB	ABACBABACB
C	CC

Exercise 6 [Divisibility by 6] (medium)

Write a Turing machine that, given an initial tape with an integer N, terminates leaving on the tape only the sequence `YES` if N is divisible by 6 and `NO` otherwise.

Examples:

Input tape	Final tape
30	YES
16	NO
0	YES

Exercise 7 [Boolean expressions] (hard)

We want to apply repeatedly the following rewriting rules, where the sequence of two or three symbols (underlined) on the left of the arrow must be substituted with the corresponding sequence of symbols on its right:

- **NOT rewriting rules:** $\underline{!0} \to 1$ and $\underline{!1} \to 0$;
- **AND rewriting rules:** $\underline{\ast 00} \to 0$, $\underline{\ast 01} \to 0$, $\underline{\ast 10} \to 0$, and $\underline{\ast 11} \to 1$;
- **OR rewriting rules:** $\underline{+00} \to 0$, $\underline{+01} \to 1$, $\underline{+10} \to 1$, and $\underline{+11} \to 1$.

A sequence of symbols 0, 1, !, $\ast$ and + is said to be *solvable* if it can be reduced to just one symbol (a 0 or a 1, the *solution*) by repeated application of the above rewriting rules (in any order).

For instance, if S equals to $+\ast 1+01\ast 0!\ast 01$, the following sequence of rewritings reduces S to 1:

$$+\ast 1\underline{+01}\ast 0!\ast 01 \to +\ast 1\underline{1}\ast 0!\ast 01$$
$$+\underline{\ast 11}\ast 0!\ast 01 \to +\underline{1}\ast 0!\ast 01$$
$$+1\ast 0!\underline{\ast 01} \to +1\ast 0!\underline{0}$$
$$+1\ast 0\underline{!0} \to +1\ast 0\underline{1}$$
$$+1\underline{\ast 01} \to +1\underline{0}$$
$$\underline{+10} \to 1$$

It is worth noting that if more rewriting rules are applicable at the same time, the order in which they are applied is not relevant for the final result.

Write a Turing machine that, given an initial tape with a solvable sequence S, terminates leaving on the tape the solution of S. It is not important how rewritings are implemented on the Turing machine, it is indeed sufficient that the final solution is correct.

Examples:

Input tape	Final tape
1	1
∗1!∗1+01	0
!!1	1

Exercise 8 [Sum] (medium)

Write a Turing machine that, given an initial tape with two integers N and M separated by +, terminates leaving their sum on the tape.

Examples:

Input tape	Final tape
30+85	115
23+0	23
0+0	0

Exercise 9 [Prefix sequence] (hard)

A sequence of `A`s, `B`s and `C`s is a *prefix sequence* if it adheres to the following definition: (i) the sequence is made of just one symbol; or (ii) if S is a prefix sequence, then the sequences $SS\texttt{A}$, $SS\texttt{B}$ and $SS\texttt{C}$ (that is, built by repeating S twice and appending a symbol at the end) are prefix sequences. For example, according to our definition `A`, $\texttt{A}\underline{\texttt{A}}\texttt{A}$, $\texttt{A}\underline{\texttt{A}}\texttt{C}$, $\texttt{AAC}\underline{\texttt{AAC}}\texttt{C}$, and $\texttt{AACAACC}\underline{\texttt{AACAACC}}\texttt{A}$ are prefix sequences, whereas `AA`, `ABA`, `AABA`, `ABAABAC` are not (`ABAABAC` is not a prefix sequence because `ABA` isn't).

Write a Turing machine that, given an initial tape with a sequence of `A`s, `B`s and `C`s, terminates leaving on the tape only the sequence `YES` if the sequence is a prefix sequence and `NO` otherwise.

Examples:

Input tape	Final tape
`B`	`YES`
`AB`	`NO`
`AABAABC`	`YES`

Exercise 10 [Sieve of Eratosthenes] (hard)

A natural number $q > 1$ is said to be *prime* if it is divisible just by 1 and itself. For example, 2, 3, 5, 7, 11, 13, 17 and 19 are all primes.

Given a base 10 integer M we want to find all the prime numbers $q < M$ using the following (simplified) version of the sieve of Eratosthenes. First, we mark all the naturals between 2 and M as primes. Then, let q be the last prime we found (start with $q = 2$), and mark as "non-prime" all the multiples of q (for example, for $q = 2$ the naturals 4, 6, 8, 10, 12, 14, ... will be marked as "non-prime"). Finally, update q with the next number marked as prime, and repeat the previous step until there are no primes left to inspect.

For example, when $M = 20$, the sieve performs the following steps

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
N	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P

($q \leftarrow 2$)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
N	P	P	$\underline{\texttt{N}}$	P	$\underline{\texttt{N}}$	P	$\underline{\texttt{N}}$	P	$\underline{\texttt{N}}$	P	$\underline{\texttt{N}}$	P	$\underline{\texttt{N}}$	P	$\underline{\texttt{N}}$	P	$\underline{\texttt{N}}$	P	$\underline{\texttt{N}}$

($q \leftarrow 3$)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
N	P	P	N	P	N	P	N	$\underline{\texttt{N}}$	N	P	N	P	N	$\underline{\texttt{N}}$	N	P	N	P	$\underline{\texttt{N}}$

($q \leftarrow 5, 7, 11, 13, 17, 19$)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
N	P	P	N	P	N	P	N	N	N	P	N	P	N	N	N	P	N	P	N

For $M \geq 2$, consider initial tapes of the form

$$\texttt{N}\underbrace{\texttt{P}\ldots\texttt{P}}_{M-1 \text{ times}}$$

where the i-th symbol corresponds to the natural number i ($1 \leq i \leq M$).

Write a Turing machine that, given the said initial tape, performs the sieve of Eratosthenes as described above and terminates leaving on the tape the sequence `NPPNPNPNNNPNPNNNPNPN`... where each `P` corresponds to a prime number $q \leq M$.

Examples:

Input tape	Final tape
NPPPPPPPPPPP	NPPNPNPNNNPN
NPPPPPPPPPPPPPPP	NPPNPNPNNNPNPNNN
NP	NP