Automata Theory
Homework 1 Specifications

Problem 1

Input format. The first line consists of an integer $l < 10^7$. The second line consists of a string of l characters composed of `0`'s, `1`'s, `.`'s, `+`'s, `*`'s, `(`'s, and `)`'s, which is a completely parenthesized regular expression.
e.g.

```
16
((0.((0+1)*)).1)
```

Output format. The first line consists of four integers: n, m, q_0, and F.

- n: number of states in the NFA
- m: number of transitions in the NFA
- q_0: initial state of the NFA
- F: final state of the NFA

The next m lines are the transitions in the NFA. Each line consists of two integers and a character: q, q', and c ($0 \leq q, q' < n$ and $c \in \{0, 1, \mathrm{e}\}$) (we use e for ϵ).

- q: source state of a transition
- q': destination state of a transition
- c: label of a transition

e.g.

```
12 14 0 11
0 1 0
2 3 0
4 5 1
6 4 e
6 2 e
5 7 e
3 7 e
8 6 e
8 9 e
7 9 e
7 6 e
1 8 e
10 11 1
9 10 e
```

Problem 2

Input format. The first line contains an integer l. The second line contains a string of l characters composed of `0`'s and `1`'s. The remaining part describes an NFA using the same format as above. (It is guaranteed that $l \times m < 10^7$ for this problem, where m is the number of transitions in the NFA.)

e.g.

```
18
001001010110101011
12 14 0 11
0 1 0
2 3 0
4 5 1
6 4 e
6 2 e
5 7 e
3 7 e
8 6 e
8 9 e
7 9 e
7 6 e
1 8 e
10 11 1
9 10 e
```

Output format. There is one line of output. "yes" if the NFA accepts the string; "no" otherwise.

e.g.

```
yes
```

Grading

Your program must use standard input/output. For each instance, your program may use up to 1 second and 1 GB of memory. **You are not allowed to use any non-standard libraries or regular expression libraries (e.g., regex of C++ or re of Python).**

Environment. We suggest using one of C, C++ or Python. If it is difficult for you to use one of the languages mentioned above, please contact the TA (`ta@theory.snu.ac.kr`). The compilers/interpreters and compilation commands that will be used for grading are as follows:

- C: gcc (Debian 12.2.0-14) 12.2.0
  `gcc -std=gnu99 -O2 main.c -o main`
- C++: g++ (Debian 12.2.0-14) 12.2.0
  `g++ -std=gnu++17 -O2 main.cc -o main`
- Python: Python 3.11.5

Submission. Please compress your submission as `AT_HW1_[student ID].zip` (e.g., `AT_HW1_2024-12345.zip`). Your submission should include exactly 11 files:

- Source code for problem 1 (Choose one from `main1.c`, `main1.cc`, `main1.py`)
- Source code for problem 2 (Choose one from `main2.c`, `main2.cc`, `main2.py`)
- Report (`report.pdf`)
- Two running examples for problem 1
  (`prob1_input1.txt`, `prob1_output1.txt`, `prob1_input2.txt`, `prob1_output2.txt`)
- Two running examples for problem 2
  (`prob2_input1.txt`, `prob2_output1.txt`, `prob2_input2.txt`, `prob2_output2.txt`)

Hand in your submission to eTL. Please make sure that the submission files are placed at the top level of the compression and that submission file names are the same as above. **Please follow the rules for submission file names.**