Exercise 1

1. **Illegal identifiers**

```
int 3b = 0;
```

2. **Illegal numbers**

```
int a = 0x3q
```

Exercise 2

1. `s` itself is both a prefix and a suffix of `s`, so `x = s`.
2. It depends on `s`, for example, is `s = "aabcaa"` , then `y = "aa"` is acceptable, but if `s = "abc"`, then there is no `t` that can be both a prefix and a suffix of `s`. Thus, whether `y` exists isn't guaranteed.

Exercise 3

1. Substrings of length m are equivalent to removing $n - m$ characters of string `s`. As we can remove consecutive characters in both sides, there are $n - m + 1$ choices. So the number of substrings is $n - m + 1$.
2. Consider each character in the string as a binary digit (1 means selected and 0 not), then each binary numbers represents a subsequence. Thus, the total count of subsequences are the number of `n` digits binary number 2^n.

Exercise 4

1.
```
digit -> [0-9]
non_zero -> [1-9]
country -> 86
area -> 755
hyphen -> -
valid_telephone_number -> country hyphen area hyphen non_zero digit{7}
```

2.
```
valid_telephone_number = 86-755-[1-9][0-9]{7}
```

Exercise 5

- $L_1 \subseteq L_2$

Each unit of any strings expressed by 0^*1^* is 0 or 1, which can be represented by $0|1$, so $L(0^*1^*) \subseteq L((0|1)^*)$. And by $a^{**} = a^*$,
$L_1 = L((0^*1^*)^*) \subseteq L((0|1)^{**}) = L((0|1)^*) = L_2$

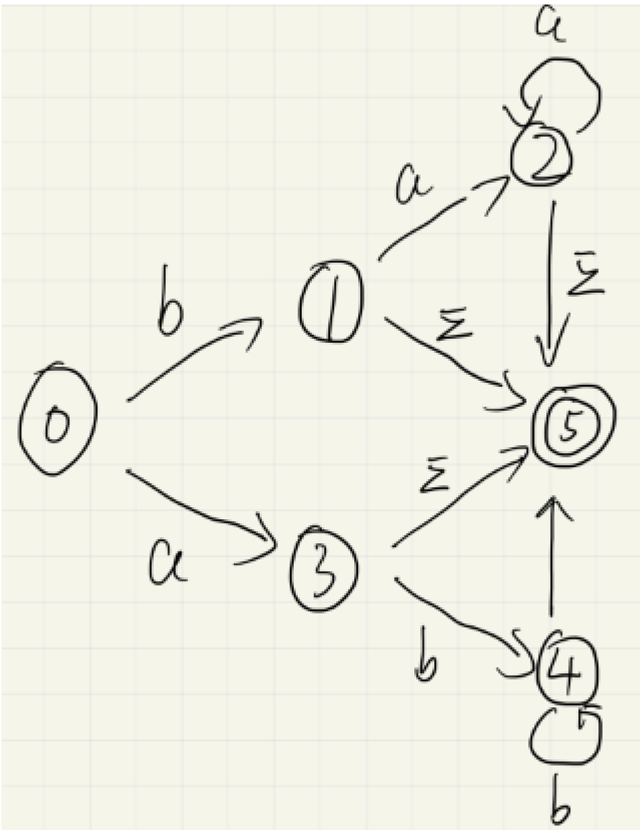
- $L_2 \subseteq L_1$

  Similarly, each unit of any strings expressed by $(0|1)^*$ is 0 or 1 , which can be represented by 0^*1^*. So $L_2 = L((0|1)^*) \subseteq L((0^*1^*)^*) = L_1$

- Thus, $L_1 = L_2$

Exercise 6

1.



2. No. The priority of alphabet is higher than `'|'`, so the regular expression is $(ba^*)|(ab^*)$, it can recognize `"ba"` and `"ab"` respectively but not together.