

۱- با استفاده از دو روش imperative و declarative شبه کد تابعی را بنویسید که عدد n را از ورودی بگیرد و جمله‌ی n ام دنباله‌ی فیبوناچی را return کند.

۲- فرض کنید که token های یک زبان برنامه‌نویسی به صورت زیر باشند:

$$R_1 = \{ 'if', 'else', 'while' \}$$

$$R_2 = ['1' - '9'] ['0' - '9']^*$$

$$R_3 = \{ '>', '<', '=' \}$$

$$R_4 = ['a' - 'z', 'A' - 'Z'] (['a' - 'z', 'A' - 'Z'] + ['0' - '9'] + '_')^*$$

$$R_5 = \{ ';' \}$$

$$R_6 = [\backslash n \backslash t]^+$$

الفبای این زبان برنامه‌نویسی فقط شامل کاراکترهایی است که حداقل در یکی از این token ها بتوانند به کار روند. در این زبان برنامه‌نویسی، R_6 مربوط به whitespace است. خروجی تحلیل‌گر لغوی را برای رشته‌ی ورودی زیر بنویسید و در صورت وجود error در زمان lexical analysis از روش panic mode استفاده کنید. رشته‌ی ورودی:

“if else25>40.5 a_1 = 1; else _a = 2;”

۳- فرض کنید که در یک زبان برنامه‌نویسی هر identifier با یک حرف (حرف بزرگ یا کوچک) شروع می‌شود و حداقل ۱ کاراکتر و حداکثر ۴ کاراکتر دارد. کاراکترهای دوم تا چهارم می‌توانند رقم (digit) باشند. در identifier های این زبان برنامه‌نویسی تعداد رقم‌ها نباید بیشتر از تعداد حروف باشد. برای این token یک DFA رسم کنید.

۴- یک عبارت منظم بنویسید که نشان دهنده‌ی همه‌ی رشته‌های باینری باشد که شامل دو عدد ۱ هستند که بین آنها تعداد فرد کاراکتر وجود دارد.