فاز اول:

در فاز اول پروژه قصد داریم تا ماژول اول یعنی lexical analyzer خود را پیاده سازی کنیم.

در این فاز برنامه .lex ما یک فایل به زبان تعریف شده ورودی میگیرد و تمامی توکن های آن را شناسایی و در فایلی به اسم tokens.txt چاپ میکند.

در این فاز ما فقط از ابزار flex استفاده میکنیم و نیازی به ابزار bison نیست.

برای مثال اگر کد ما به شکل زیر باشد:

```
int main()
<
    int a = 9.
    if (a < 10)
    <
        a = a * 10.
    >
    while(a < 10)
    <
        a = a + 1.
    >
>
```

فایل خروجی برنامه (tokens.txt) به شکل زیر می شود:

```
TOKEN_INT
TOKEN_MAIN
TOKEN_LEFTPAREN
TOKEN_RIGHTPAREN
TOKEN_LEFTB
TOKEN_INT
TOKEN_IDENTIFIER
TOKEN_ASSIGN
TOKEN_INT_CONST
TOKEN_DOT
TOKEN_IF
TOKEN_LEFTPAREN
TOKEN_IDENTIFIER
TOKEN_LESS
TOKEN_INT_CONST
TOKEN_RIGHTPAREN
TOKEN_LEFTB
TOKEN_IDENTIFIER
TOKEN_ASSIGN
TOKEN_IDENTIFIER
TOKEN_MULTIPLY
TOKEN_INT_CONST
TOKEN_DOT
TOKEN_RIGHTB
TOKEN_IDENTIFIER
TOKEN_LEFTPAREN
TOKEN_IDENTIFIER
TOKEN_LESS
TOKEN_INT_CONST
TOKEN_RIGHTPAREN
TOKEN_LEFTB
TOKEN_IDENTIFIER
TOKEN_ASSIGN
TOKEN_IDENTIFIER
TOKEN_PLUS
TOKEN_INT_CONST
TOKEN_DOT
TOKEN_RIGHTB
TOKEN_RIGHTB
```

فاز دوم:

در فاز دوم پروژه قصد داریم ماژول دوم یعنی syntax analyzer را پیاده سازی کنیم.

در این فاز برنامه یک فایل به زبان تعریف شده ورودی میگیرد و در نهایت اگر خطای syntax داشت آن را چاپ کرده و فرآیند کامپایل خاتمه می یابد (اولین خطا باعث خاتمه فرآیند کامپایل میشود).

مثال هایی از خطا های syntax در زبان ما:

```
for (int a = 0. a < 5. a=a+1
<...>
```

پرانتز بسته دستور for فراموش شده است.

```
while(12!==10)
<....>
```

توکن ==! در زبان ما تعریف نشده است.

شیوه تحویل:

لطفا فایل های خود را با فرمت studentNumber1_studentNumber2.zip زیپ کرده و آپلود نمایید.