کامپایلر TesLang: گام اول

در گام اول از تمرین عملی درس اصول طراحی کامپایلر، باید یک تحلیلگر لغوی بنویسید. در این گام باید برنامه‌ای بنویسید که با خواندن یک فایل از ورودی استاندارد در زبان TesLang، توکن های (Token) آن را چاپ کند. برای مثال، به نمونه کد زیر توجه کنید :

```
function count_even_numbers(A: List): Number => {
    let even_count: Number = 0;

    for (i, v of A) {
        if (v % 2 == 0) {
            even_count = even_count + 1;
        }
    }

    return even_count;
}
```

خروجی گام اول برای مثال بالا باید به شکل زیر باشد :

```
function
count_even_numbers
(
A
:
List
)
:
Number
=>
{
let
even_count
:
Number
=
0
;
for
```

```
(
i
,
v
of
A
)
{
if
(
v
%
2
==
0
)
{
even_count
=
even_count
+
1
;
}
}
return
even_count
;
}
```

نکته: توجه کنید که لزوماً token ها به وسیله فاصله از هم جدا نشده اند!

نکته: رشته های ورودی باید با استفاده از فایل و با استفاده از نشانگر فایل (File Pointer) به صورت کاراکتر به کاراکتر خوانده شوند.

نکته: بهتر است که در این گام شماره سطر و ستون توکن ها را نیز نگهداری کنید!

نکته: می‌توانید برای انجام این پروژه از ابزار های parser generator استفاده کنید مانند PLY, YACC, ...

موفق و پیروز باشید