گام دوم تمرین عملی درس طراحی کامپایلر

در گام دوم از تمرین عملی درس طراحی کامپایلر، تحلیل نحوی را انجام می‌دهید. می‌توانید یکی از دو حالت زیر را انجام دهید. در هر یک از این دو حالت، برنامه‌ای می‌نویسید که با خواندن یک فایل تسلنگ از ورودی استاندارد و تحلیل نحوی آن، پیغام‌هایی را چاپ می‌کند.

حالت اول: اجرای دستورات

کد تسلنگ را می‌توانید مشابه یک مفسر اجرا کنید. برای این کار، در هنگام تجزیه، باید عناصر روی پشته را نگه داشته باشید و دستورات را اجرا کنید. در این حالت، عملگرهای اعداد صحیح (مثل جمع) و توابع if و print را پیاده‌سازی کنید.

حالت دوم: شمارش ورودی‌ها و خروجی‌ها

برنامه‌ی شما باید به ازای همه‌ی نام‌هایی که با دستور def تعریف می‌شوند، تعداد ورودی‌ها و خروجی‌های آنها را چاپ کند. ورودی‌های یعنی چند مقدار از روی پشته برداشته می‌شوند و خروجی یعنی پس از اجرای تابع چند مقدار جدید به پشته اضافه می‌گردند. در این حالت، فقط توابع اعداد صحیح (مثل جمع) و توابع def و print را در تحلیل کنید. برای نمونه، فایل زیر را در نظر بگیرید:

```
1     /x {1 2 +} def
2     /sum3 {+ +} def
3     /say {(saying) print print} def
```

برنامه‌ی شما پس از خواندن این فایل باید خط‌های زیر را چاپ نماید.

```
x: 0 -> 1
sum3: 3 -> 1
say: 1 -> 0
```

در پیاده‌سازی این گام می‌توانید به هر متغیر تجزیه، دو مقدار مفهومی نسبت دهید: مقدار اول مشخص می‌کند هر متغیر چند عنصر به پشته اضافه می‌کند و مقدار دوم مشخص می‌کند هر متغیر چند عنصر از پشته بر می‌دارد. برای مثال، یک عدد (مثل ۳) چیزی از پشته بر نمی‌دارد و یک مقدار به پشته اضافه می‌کند و یک عملگری دودویی (مثل + یا >)، دو مقدار از پشته بر می‌دارد و یک مقدار به پشته اضافه می‌کند. همچنین، def دو مقدار از پشته بر می‌دارد.

برای دستورات بین } و { نیز مقادیر مفهومی را باید محاسبه کنید. برای مثال «{ + 5 3 2 }» فقط یک مقدار روی پشته است اما اگر اجرا شود (توسط if یا اجرا پس از تعریف توسط def)، دو مقدار روی پشته اضافه می‌کند.