

کامپایلر TesLang: گام دوم

در گام دوم از تمرین عملی درس طراحی کامپایلر، به **تحلیل نحوی** می‌پردازیم. در این گام برنامه‌ای باید بنویسیم که با خواندن یک فایل **TesLang** از ورودی استاندارد و تحلیل نحوی آن، پیغام‌های مناسبی چاپ کند. دقت کنید که **تجزیه (parse)** برنامه ورودی الزامی است و منطق برنامه باید به کمک **عملیات مفهومی** نوشته شود. در این پیغام‌ها نوع خطا را چاپ کنید :

1. **فراخوانی** یک تابع با **تعداد** نادرست پارامترها.
2. **فراخوانی** یک تابع با **نوع** نادرست پارامترها.
3. **برگرداندن** مقداری با **نوع** نادرست از یک تابع (**Return Type**).
4. **دسترسی** به متغیرهایی که **تعریف نشده اند** (**Undefined Variables**).
5. **دسترسی** به متغیرهایی که **مقداردهی نشده اند** (**Unassigned Variables**).
6. **مشخص کردن نادرست نوع داده‌ای** متغیرها (**Unknown Type**).

```
function find(A: List, n: Number): Number => {  
    let k: Number;  
    let j: 0;  
    for (i, a of A) {  
        if (n == k) {  
            return j;  
        }  
        j = j + 1;  
    }  
  
    return -1;  
}
```

```
function main(): List => {  
    let A: Number;  
    let a: Number;  
  
    A = makeList(3);  
    A[0] = 1;  
    A[1] = 2;  
    A[2] = 3;  
  
    log(find(A, a));  
    log(find(A));  
    log(find(a, A));  
  
    return A;  
}
```

تمرین های برنامه نویسی درس اصول طراحی کامپایلر

مطلوب است که خطاهای زیر را پس از خواندن فایل، چاپ کنید :

```
Error:
  function 'main': variable 'A' expected to be of type 'List' but it is
  'Number' instead.

Error:
  function 'find': Variable 'k' is used before being assigned.

Error:
  function 'find': expected 'A' to be of type 'List', but got 'Number'
  instead.

Error:
  function 'find' expects 2 arguments but got 1.

Error:
  function 'find': expected 'n' to be of type 'Number', but got 'Null'
  instead.

✓ Error:
    function 'find': wrong type '0' found. types must be one of the
    following 'Number', 'List', 'Null'

Error:
  function 'find': Variable 'j' is not defined.

Error:
  function 'main': wrong return type. expected 'List' but got 'Number'
  instead.

Warning:
  expected return type for function 'main' is 'Number'!
```

نکته: توجه کنید که خروجی شما ممکن است دقیقاً منطبق با خطاهای بالا نباشد، سعی کنید حداکثر خطاهای ممکن را بیابید و به بهترین نحو آنها را چاپ کنید.

نکته: در این گام بهتر است در پیغام ها به شماره خطوطی که در آن خطاها را یافتید نیز اشاره کنید.

نکات تکمیلی :

- برای پیاده سازی این بخش از کامپایلر، نیاز به یک ساختمان داده دارید تا بتوانید متغیرها و توابع تعریفی را نگهداری کنید. برای پیاده سازی **جدول نمادها** (Symbol Table) باید از ساختمان داده پشته استفاده کنید.
- برای پیاده سازی این جدول می‌توانید به صورت درختی و **تودرتو** (nested) عمل کرده و یا در یک جدول این کار را انجام دهید.
- در صورتی که از یک جدول استفاده می‌کنید، باید یک field در ساختار داده‌ای خود برای **محدوده** (Scope) نگهداری کنید.

نمونه ساختار داده‌ای برای جدول نمادها:

```
struct SymbolTableNode {  
    string name;  
    string type;  
    ...others;  
}
```