

### کامپایلر **TesLang**: گام سوم

در گام سوم از تمرین عملی درس طراحی کامپایلر، برنامه‌ای خواهید نوشت که با خواندن یک فایل **TesLang** از ورودی استاندارد، کد میانی آن را تولید می‌کند. برای مثال به کد زیر توجه کنید:

```
def int add(int a, int b) {
    var int result = a + b;
    result = a + b;
    return result;
}

def int main() {
    var int a;
    var int b;

    a = scan();
    b = scan();

    print(add(a, b));

    return 0;
}
```

برنامه شما پس از خواندن این کد باید **کد میانی (IR)** مربوطه را تولید کند، به مثال زیر توجه کنید:

```
proc add
    # a => r1, b => r2 & return value => r0
    add r3, r1, r2 # calculate (a + b) and move to temp reg r4
    mov r0, r3 # move the value of r3 to r0 to return
    ret

proc main
    # a => r1, b => r2 and return value => r0
    call read, r1 # stdin a into r1
    call read, r2 # stdin a into r2
    call add, r3, r1, r2 # call add function to calculate (a + b)
    call log, r3 # log the output in the console
    mov r0, 0 # move 0 to r0 as return value in 'main'
    ret
```

به نکات زیر توجه کنید:

- برای بررسی صحت کد میانی تولید شده، باید با استفاده از برنامه [tsvm](#) کد میانی را اجرا کنید.
- برای خواندن ورودی و خروجی باید از توابع داخلی TSIR استفاده کنید.