

- [练习8.1.1](#)
- [练习8.1.2](#)

## 练习8.1.1

假设**n** 在一个内存位置中，**s**、**i** 分配在寄存器中，为下面的语句序列生成代码，并计算生成的目标代码的代价（其中访存代价为**3**，分支代价为**2**，其他指令代价为**1**）

```
s = 8
i = 8
L1:
  if i < n goto L2
  s = s + i
  i = i - 1
  goto L1
L2:
```

生成的目标代码为：

```
LD R1, 8      # s = 8
LD R2, 8      # i = 8
LD R3, n
L1:
    SUB R4, R2, R3  # R4 = i-n
    BLTZ R4, L2     # if i < n goto L2
    add R1, R1, R2  # s = s + i
    subi R2, R2, 1  # i = i - 1
    BR L1          # goto L1
L2:
```

指令代价依次为：3,3,3,1,2,1,1,2，合计的目标代码代价为16

## 练习8.1.2

假设使用栈式分配，且假设**a** 和**b** 都是元素大小为**4**字节的数组，为下面的三地址语句生成代码

```
x = a[i]
y = b[j]
```

```
a[i] = y
b[j] = x
```

假设 **a**、**b**、**x** 和 **y** 的位置相对于栈指针 **SP** 的偏移量是 **offset(\*)**:

生成的代码为:

```
LD R1, i
MUL R1, R1, 4
ADD R2, SP, offset(a)
LD R3, R1(R2)      # R3 = a[i]
ST offset(x), R3    # x = a[i]

LD R4, j
MUL R4, R4, 4
ADD R5, SP, offset(b)
LD R6, R4(R5)      # R6 = b[j]
ST offset(y), R6    # y = b[j]

ST R1(R2), R6      # a[i]=y
ST R4(R5), R3      # b[j]=x
```