

练习15

(1)

栈式

```
PUSH A
PUSH B
PUSH C
ADD
MUL
PUSH D
PUSH B
PUSH C
ADD
DIV
SUB
POP X
```

累加器式

```
LOAD, B
ADD, C
STORE, R1
LOAD, A
MUL, R1
STORE, R2
LOAD, R2
DIV, D
STORE, R2
LOAD, R2
DIV, R1
STORE, X
```

寄存器-内存式

```
LOAD R0,B
ADD R0,R0,C
MUL R1,R0,A
STORE R0, T1
LOAD R2,D
DIV R2,R2,T1
STORE R2,T2
SUB R1,R1,T2
STORE R1,X
```

寄存器-寄存器式

```
LOAD R0,A
LOAD,R1,B
LOAD,R2,C
ADD,R1,R1,R2
MUL R0,R0,R1
LOAD R3,D
DIV R3,R3,R1
SUB R0,R0,R3
STORE R0,X
```

(2)

栈式使用26个时钟周期。

累加器使用28个时钟周期。

寄存器-内存使用19个时钟周期。

寄存器-寄存器使用19个时钟周期。

4

如果 `ss` 被分配在程序运行栈中，每次函数 `foo` 调用结束后，`ss` 都会被销毁，然后在下次 `foo` 调用时重新创建和初始化，所以每次打印的都是初始值0。

如果 `ss` 被分配在静态数据区，那么在整个程序运行期间，`ss` 只会被初始化一次，然后每次函数 `foo` 调用时，都会使用上一次调用结束后 `ss` 的值，所以每次打印的值会递增。打印结果为：0，1。

5

先按序移出所有度小于2的的节点，移出b，

然后没有小于2的节点，所以移出a，移出i，移出x，此时还剩下最后一个节点y，然后倒序排出所有点到图中：所以最后分配情况如下：

a->2, b->1, i->1, x->1, y->2

6

四元式表达的形式如下：

```
(1) T1 = B + C
(2) T2 = A * T1
(3) T3 = D * T1
(4) T4 = T2 - T3
```

对应的X86汇编代码如下：

```
mov eax, [esp+14h]
add eax, [esp+18h]
mov edx, eax
mov eax, [esp+10h]
imul eax, edx
mov ecx, eax
mov eax, [esp+1ch]
imul eax, edx
sub ecx, eax
```

注意，这里假设所有操作数都是整数，并且使用的是32位寻址。