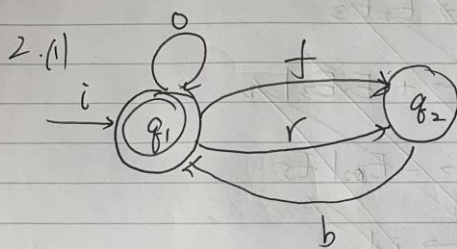


1. ①有量化的内置类型可以提高可读性和可写性，易于控制和使用特征和成分

②自动垃圾回收机制提高了可读性和可靠性。如果程序员没有这样的机制时，如果程序员没有合理处理垃圾变量，则会发生内存泄漏等问题

③函数多返回值也提高了程序的可写性，~~修复~~子程序之间的信息传递更容易

④错误处理机制提高了可靠性。

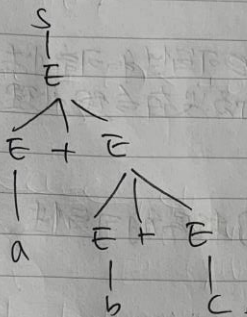
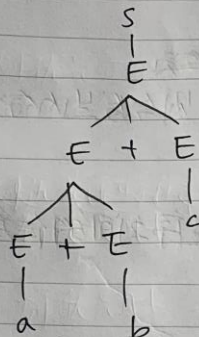


(2)  $\langle S_1 \rangle \rightarrow i \langle S_2 \rangle \cup$

$\langle S_2 \rangle \rightarrow \langle S_2 \rangle \langle S_3 \rangle \mid \langle S_3 \rangle$

$\langle S_3 \rangle \rightarrow nb \mid fb \mid a$

⑤. 3. (1) 有歧义: 对  $a+b+c$  有两个语法分析树



(2) ~~$S \rightarrow E_1$   
 $E_1 \rightarrow E_1 + E_2 \mid E_2$   
 $E_2 \rightarrow -E_3 \mid E_3$   
 $E_3 \rightarrow id$~~

~~$S \rightarrow E_1$   
 $E_1 \rightarrow E_2 E_3$   
 $E_2 \rightarrow + E_3 E_3 \mid \wedge$   
 $E_3 \rightarrow -E_4 \mid E_5$   
 $E_5 \rightarrow id$~~

$S \rightarrow E_1$   
 $E_1 \rightarrow E_3 E_2$   
 $E_2 \rightarrow + E_3 E_2 \mid \wedge$   
 $E_3 \rightarrow -E_4 \mid E_4$   
 $E_4 \rightarrow id$



4. (1) 按值传递 初始时  $b = 8$   
静态作用域 中  $a = 10$

$$b = 8 \times 10 = 80$$

$$a = 10 - 80 = -70$$

$\therefore \text{print}(b)$  会输出 80.

且函数不改变  $a$   $\therefore \text{print}(a)$  会输出 8.

$\therefore \text{print}: 80, 8$

(2) 动态作用域 中  $a = 8$

$$b = 8 \times 8 = 64$$

$$a = 8 - 64 = -56$$

$\therefore \text{print}(b)$  输出 64

而  $\text{print}(a)$  输出 8.

$\therefore \text{print}: 64, 8$

(3) 按引用传递

中  $a = a - b$  会影响实参的值

$\therefore \text{print}(b): 80$

$\text{print}(a): -70$

$\therefore \text{print}: 80, -70$

(4) 同理

$\text{print}(b): 64$

$\text{print}(a): -56$

$\therefore \text{print}: 64, -56$

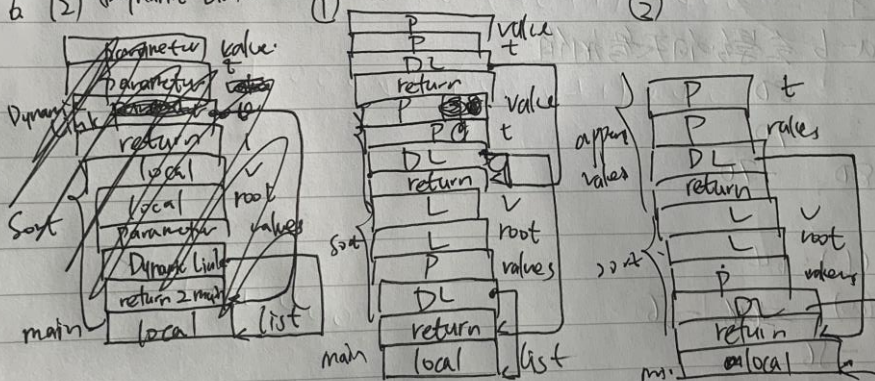
5. (1) 行优先遍历:  $(5 \times 400 + 6) \times 4000 + 7$   
 $= 8024007$

列优先遍历:  $(7 \times 400 + 6) \times 40 + 5$   
 $= 112245$

(2) ~~行优先遍历~~ 根据结果可看出  
 行优先时, 会缓存大约  $(40 \times 400 + 400) \times 4000$  元素  
 即: 65,600,000 元素  
 列优先时, 会缓存大约  $(4000 \times 400 + 400) \times 40$  元素  
 64,016,000 元素

列优先方法比较快

\*  
 b (2) Dynamic Link: DL (Parameter: P, Local: L)





7. 我会把矩阵作为基本变量类型之一，因为矩阵运算在各个方面的应用中可以提高运算效率。矩阵可以看成是多维数组，但与数组不同的是可以对两个以上的矩阵进行运算，比如加、减、乘、除、求逆矩阵、转置等运算。两个矩阵之间的运算要求类型检测，会使用结构体作为判断。还会提供强制类型转换，将 `int`、`float` 类型的矩阵转换成 `double` 类型的。为了提高兼容性，会用到像 `python` 一样的弱类型机制。矩阵的存储会提供两种方法：完全存储方式和稀疏存储方式。完全存储方式会将矩阵中的每一个元素存储（0 元素也存储），对于稀疏矩阵，这样的存储方式是很浪费空间的，因此稀疏存储方式只会将非零元素存储，像存储散列类型一样，由位置（key）和值（value）组成。