# hw5

## 4.13

```
S' -> {S.loop := false} S
S  -> id := E
S  -> if E then {S1.loop := S.loop}S1
S  -> while E do {S1.loop := true}S1
S  ->begin {S1.loop := S.loop}S1;{S2.loop := S.loop}S2.end
S  ->begin{if not S.loop then print("error")}
```

## 4.15

| 产生式 | 代码段 |
| --- | --- |
| E -> E1 + T | nptr[top-2] = mkNode('+',nptr[top-2],nptr[top]) |
| E -> E1 - T | nptr[top-2] = mkNode('-',nptr[top-2],nptr[top]) |
| E -> T | 值不变，无动作 |
| T -> (E) | nptr[top-2] = nptr[top-1] |
| T -> id | nptr[top] = mkLeaf(id,id,entry) |
| T -> num | nptr[top] = mkLeaf(num,num,val) |

## 5.5

foo的类型表达式为：`array(0...99,record((a*integer)*(b*integer)));`

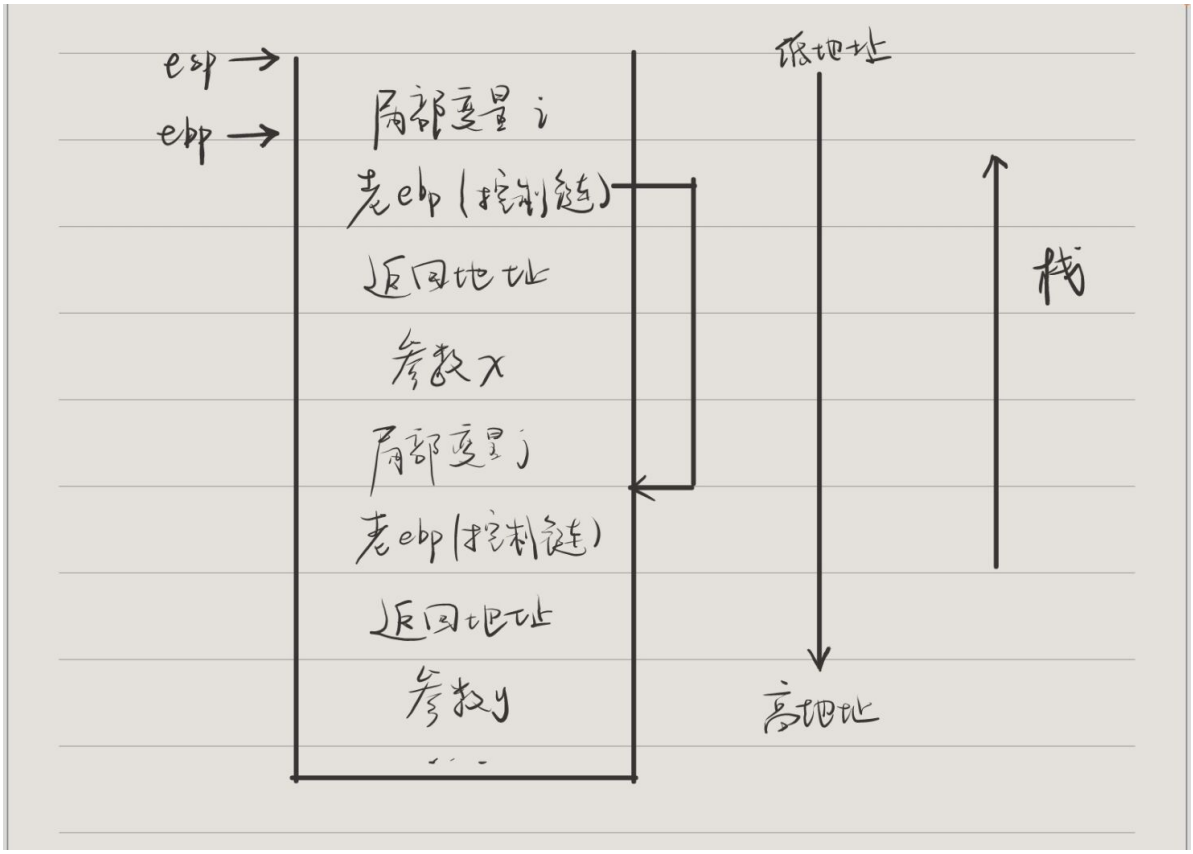
bar的类型表达式为:`(integer*record((a*integer)*(b*integer)))->pointer(record((a*integer)*(b*integer)))`

## 6.5

1.编译器按照结构体成员列表顺序给每个成员分配内存
2.当成员需要满足正确的边界对齐时，成员之间用额外字节填充
3.结构体的首地址必须满足结构体中边界对齐要求最为严格的数据类型所要求的首地址
4.结构体的大小为其最宽数据类型的整数倍

故b的size等于16

## 6.6


esp
ebp
局部变量 i
老ebp（控制链）
返回地址
参数x
局部变量j
老ebp（控制链）
返回地址
参数y
低地址
高地址
栈