第 6 章课后作业

1. 为下面的表达式构造DAG：

（（x+y）-（（x+y）*（x-y））+（（x+y）*（x-y））

2. 将下列赋值语句翻译为四元式序列，三元式序列，间接三元式序列：

(1) a = b[i] + c[j]

(2) a[i] = b*c - b*d

3. 使用下图所示的翻译方案来翻译赋值语句 x = a[i][j] + b[i][j]:

$$
\begin{array}{lll}
S \rightarrow \mathbf{id} = E\ ; & \{\ gen(\ top.get(\mathbf{id}.lexeme)\ '='\ E.addr);\ \} \\
\quad \mid\ L = E\ ; & \{\ gen(L.array\ .base\ '['\ L.addr\ ']'\ '='\ E.addr);\ \} \\
E \rightarrow E_1 + E_2 & \{\ E.addr = \mathbf{new}\ Temp\,();\ gen(E.addr\ '='\ E_1.addr\ '+'\ E_2.addr);\ \} \\
\quad \mid\ \mathbf{id} & \{\ E.addr = top.get(\mathbf{id}.lexeme);\ \} \\
\quad \mid\ L & \{\ E.addr = \mathbf{new}\ Temp\,();\ gen(E.addr\ '='\ L.array.base\ '['\ L.addr\ ']');\ \} \\
L \rightarrow \mathbf{id}\ [\ E\ ] & \{\ L.array = top.get(\mathbf{id}.lexeme);\ L.type = L.array.type.elem;\ L.addr = \mathbf{new}\ Temp\,();\ gen(L.addr\ '='\ E.addr\ '*'\ L.type.width);\ \} \\
\quad \mid\ L_1\ [\ E\ ] & \{\ L.array = L_1.array;\ L.type = L_1.type.elem;\ t = \mathbf{new}\ Temp\,();\ L.addr = \mathbf{new}\ Temp\,();\ gen(t\ '='\ E.addr\ '*'\ L.type.width);\ gen(L.addr\ '='\ L_1.addr\ '+'\ t);\ \}
\end{array}
$$

4.一个按行存放的实数型数组A[i,j,k]的下标i的范围为1~4，下标j的范围为0~4，且下标k的范围为5~10。每个实数占8个字节。假设数组A从0字节开始存放，计算下列元素的位置：

(1) A[3,4,5]

(2) A[1,2,7]

(3) A[4,3,9]

5.使用下图中的翻译方案翻译表达式a == b && (c == d || e == f)，并给出每个子表达式的真值列表与假值列表，你可以假设第一条被生成的指令的地址是100：

1)	$B \rightarrow B_1 \ \texttt{||} \ M \ B_2$	{ $backpatch(B_1.falselist, M.instr)$; $B.truelist = merge(B_1.truelist, B_2.truelist)$; $B.falselist = B_2.falselist$; }
2)	$B \rightarrow B_1 \ \texttt{\&\&} \ M \ B_2$	{ $backpatch(B_1.truelist, M.instr)$; $B.truelist = B_2.truelist$; $B.falselist = merge(B_1.falselist, B_2.falselist)$; }
3)	$B \rightarrow \ \texttt{!} \ B_1$	{ $B.truelist = B_1.falselist$; $B.falselist = B_1.truelist$; }
4)	$B \rightarrow \texttt{(} \ B_1 \ \texttt{)}$	{ $B.truelist = B_1.truelist$; $B.falselist = B_1.falselist$; }
5)	$B \rightarrow E_1 \ \textbf{rel} \ E_2$	{ $B.truelist = makelist(nextinstr)$; $B.falselist = makelist(nextinstr + 1)$; $gen('\texttt{if}' \ E_1.addr \ \textbf{rel}.op \ E_2.addr \ '\texttt{goto}\ \_')$; $gen('\texttt{goto}\ \_')$; }
6)	$B \rightarrow \textbf{true}$	{ $B.truelist = makelist(nextinstr)$; $gen('\texttt{goto}\ \_')$; }
7)	$B \rightarrow \textbf{false}$	{ $B.falselist = makelist(nextinstr)$; $gen('\texttt{goto}\ \_')$; }
8)	$M \rightarrow \epsilon$	{ $M.instr = nextinstr$; }