trans(`#include <stdio.h> int main()` *block*) := *blocktrans*(*block*)

blocktrans(`{` *decl statseq* `return 0;}`) := *stseqtrans*(*statseq*, *update*(*decl*, $tab_\emptyset$), 1)

stseqtrans($stat_1\ stat_2\ \ldots\ stat_n, tab, a$) :=
sttrans($stat_1, tab, a.1$)
sttrans($stat_2, tab, a.2$)
...
sttrans($stat_n, tab, a.n$)

sttrans(`{` $stat_1\ stat_2\ \ldots\ stat_n$ `}`, tab, a) := *stseqtrans*($stat_1\ stat_2\ \ldots\ stat_n, tab, a$)

sttrans(*id* `=` *exp*`;`, tab, a) :=
if $tab(id) = (\text{var}, n)$ then *simpleexptrans*(*exp*, *tab*) STORE n;

sttrans(`scanf("%d",&`*id*`);`, tab, a) := if $tab(id) = (\text{var}, n)$ then READ n;

sttrans(`printf("%d",`*id*`);`, tab, a) := if $tab(id) = (\text{var}, n)$ then WRITE n;

sttrans(`if (` *exp* `)` *stat*, tab, a) :=
boolexptrans(*exp*, *tab*)
JMC $a.1$;
sttrans(*stat*, *tab*, $a.2$)
$a.1$:

sttrans(`if (` *exp* `)` $stat_1$ `else` $stat_2, tab, a$) :=
boolexptrans(*exp*, *tab*)
JMC $a.1$;
sttrans($stat_1$, *tab*, $a.2$)
JMP $a.3$;
$a.1$: *sttrans*($stat_2$, *tab*, $a.4$)
$a.3$:

sttrans(`while (` *exp* `)` *stat*, tab, a) :=
$a.1$: *boolexptrans*(*exp*, *tab*)
JMC $a.2$;
sttrans(*stat*, *tab*, $a.3$)
JMP $a.1$;
$a.2$:

boolexptrans($se_1\ rel\ se_2, tab$) :=
simpleexptrans(se_1, tab)
simpleexptrans(se_2, tab)
REL;
wobei $(rel, \texttt{REL}) \in \{(\texttt{==}, \texttt{EQ}), (\texttt{!=}, \texttt{NE}), (\texttt{<}, \texttt{LT}), (\texttt{>}, \texttt{GT}), (\texttt{<=}, \texttt{LE}), (\texttt{>=}, \texttt{GE})\ \}$

simpleexptrans(`x + a * 2, [a/(const, 5), x/(var, 1)])`) =
LOAD 1; LIT 5; LIT 2; MUL; ADD;