```
#include "token.h"
#include "nodes.h"
#include "mc_env.h"
#ifndef GENTAC
#define GENTAC

typedef struct env {
  int dstcounter;
  int lblcounter;
  TOKEN* currlbl;
}ENV;

enum tac_op
  {
    tac_plus = 1,
    tac_minus = 2,
    tac_div = 3,
    tac_mod = 4,
    tac_mult = 5,

    tac_proc = 6,
    tac_endproc = 7,
    tac_load = 8,
    tac_store = 9,
    tac_if = 10,
    tac_lbl = 11,
    tac_goto = 12,
    tac_call = 13,
    tac_rtn = 14,
    tac_innerproc = 15
  };

typedef struct simple_tac {
  TOKEN* src1;
  TOKEN* src2;
  TOKEN* dst;
}STAC;

typedef struct proc {
  TOKEN* name;
  int arity;
  TOKENLIST* args;
}PROC;

typedef struct load {
  TOKEN* src1;
  TOKEN* dst;
}LOAD;

typedef struct label {
  TOKEN* name;
}LABEL;

typedef struct iftest {
  TOKEN* op1;
  TOKEN* op2;
  int code;
  TOKEN* lbl;
}IFTEST;
```

```
61 typedef struct gotolbl {
62   TOKEN* lbl;
63 }GOTO;
64
65 typedef struct call {
66  TOKEN* name;
67  int arity;
68  TOKENLIST* args;
69 } CALL;
70
71 typedef struct rtn {
72   int type;
73   union {CALL call; TOKEN* v;};
74 }RTN;
75
76 typedef struct tac {
77 int op ;
78 union {STAC stac; PROC proc; LOAD ld; LABEL lbl; IFTEST ift; GOTO gtl; CALL
   call; RTN rtn;};
79 struct tac* next;
80 }TAC;
81
82 typedef struct bb {
83   TOKEN* id;
84   TAC* leader;
85   TAC* end;
86   struct bb *nexts[2];
87 }BB;
88
89 TAC*gen_tac(NODE*);
90 TAC* gen_tac0(NODE*, ENV*,FRME*,int);
91
92
93 #endif
```