Help and Hints for HW4

Help-hint-1:

```
if (sym==callsym) { /* procedure call */
   getsym();
   if (sym!=ident) error(XXXXXX); else
   { i=position(id, ptx);
     if(i==0) error(YYYYY); else
     if (table[i].kind==procedure)
       gen(cal,lev-table[i].level, table[i].adr);
     else error(ZZZZZ);
     getsym();
   }
```

Help-hint-2:

```
block(lev, tx)
int lev;
int tx;
{
  int dx, tx0, cx0;
  dx=3; tx0=tx; table[tx].adr=cx; gen(jmp,0,0); // current cx is saved to table[tx0].adr
                                    // Generate jmp 0,0, the second 0 tentative
  if (lev>levmax) error(?????);
  do {
    if (sym==constsym) {
      getsym();
      do {
        constdeclaration(lev,&tx,&dx);
        while(sym==comma) {
          getsym(); constdeclaration(lev,&tx,&dx);
        }
        if(sym==semicolon)getsym(); else error(?????);
      } while (sym==ident);
    }
    if (sym==varsym) {
      getsym();
      do { vardeclaration(lev,&tx,&dx);
        while (sym==comma) {
          getsym(); vardeclaration(lev,&tx,&dx);
        }
      if(sym==semicolon) getsym(); else error(??????);
```

```
    } while(sym==ident);
   }
   while(sym==procsym) {
    getsym();
    if(sym==ident){
     enter(procedure,&tx,&dx,lev); getsym();
    } else error($$$$$);
    if (sym==semicolon) getsym(); else error(??????);
    block(lev+1, tx);              // Go to a block one level higher
    if(sym==semicolon) {
     getsym();
    } else error(?????);
   }
  }while ((sym==constsym)||(sym==varsym)||(sym==procsym));
  code[table[tx0].adr].a=cx;  // The tentative jump address is fixed up
  table[tx0].adr=cx;          // the space for address for the above jmp is now occupied by
the new cx
  cx0=cx; gen(inc,0,dx);      // inc 0,dx is generated. At run time, the space of dx is
secured
  statement(lev,&tx);
  gen(opr,0,0);
}
```