```c
void findloops();
void calclivevars();
void calcdeadvars();
void unreachablecodeelim();
void remvbranchchains();
void reversebranches();
void deadasgelim();
void regalloc(int *);
void codemotion(int *);
void filldelayslots();
void readinrules();
void applypeeprules(int *);
void localconstprop(int *);
void localcopyprop(int *);
void localcse(int *);
void cseinblk(struct bblk *, int *);
```

```
#include <stdlib.h>
#include <stdio.h>
#include "opt.h"
#include "misc.h"
#include "opts.h"
#include "flow.h"

/*
 * remvbranchchains − remove branch chains
 */
void remvbranchchains()
{
}
```

```
#include <stdio.h>
#include "opt.h"
#include "io.h"
#include "misc.h"
#include "vars.h"

/*
 * localconstprop - perform constant propagation
 */
void localconstprop(int *changes)
{
}
```

```
#include <stdio.h>
#include "opt.h"
#include "misc.h"
#include "vars.h"

/*
 * localcopyprop - perform copy propagation
 */
void localcopyprop(int *changes)
{
}
```

```
#include <stdio.h>
#include "opt.h"
#include "misc.h"
#include "vars.h"

/*
 * cseinblk - perform local common subexpression elimination in a block
 */
void cseinblk(struct bblk *cblk, int *changes)
{
}

/*
 * localcse - perform local common subexpression elimination
 */
void localcse(int *changes)
{
   struct bblk *cblk;
   extern struct bblk *top;

   for (cblk = top; cblk; cblk = cblk->down)
      cseinblk(cblk, changes);
}
```

```
#include <stdio.h>
#include "opt.h"
#include "vars.h"
#include "analysis.h"
#include "opts.h"

/*
 * deadasgelim − perform dead assignment elimination
 */
void deadasgelim()
{
}
```

```c
#include <stdio.h>
#include "opt.h"
#include "misc.h"
#include "opts.h"

/*
 * filldelayslots - fill the delay slots of the transfers of control in a
 *                  function
 */
void filldelayslots()
{
}
```

```c
#include <stdio.h>
#include "opt.h"
#include "misc.h"
#include "vars.h"
#include "analysis.h"
#include "opts.h"

/*
 * calclivevars − calculate live variable information
 */
void calclivevars()
{
}

/*
 * calcdeadvars − calculate dead variable information
 */
void calcdeadvars()
{
}
```

```
#include <stdio.h>
#include "opt.h"
#include "misc.h"
#include "analysis.h"
#include "vect.h"

struct loopnode *loops;    /* head of linked list of loops */

/*
 * findloops - locate loops in the program and build the loop data structure
 */
void findloops()
{
}
```

```c
#include <stdio.h>
#include "opt.h"
#include "misc.h"
#include "vars.h"
#include "opts.h"

extern int numpeeprules;

/*
 * readinrules − read in the peephole rules from the rules file
 */
void readinrules()
{
}

/*
 * applypeeprules − apply peephole rules to the function
 */
void applypeeprules(int *changes)
{
}
```

```
#include <stdio.h>
#include <strings.h>
#include "opt.h"
#include "misc.h"
#include "vars.h"
#include "opts.h"

/*
 * regalloc − perform register allocation
 */
void regalloc(int *changes)
{
}
```

```c
#include <stdio.h>
#include "opt.h"
#include "misc.h"
#include "opts.h"

/*
 * reversebranches - avoid jumps by reversing branches
 */
void reversebranches()
{
}
```

```
CC = gcc
DIR = /home/faculty/whalley/cop6622proj/lib
DIROBJ = /home/faculty/whalley/cop6622proj/obj
CFLAGS = -g -I$(DIR) -I. -c -Wall
LFLAGS = -g
OBJECTS = chains.o constprop.o copyprop.o cse.o deadasgs.o fillslots.o flow.o i
o.o livedeadvars.o loops.o misc.o motion.o opt.o peephole.o regalloc.o reverse.o
 unreachable.o vars.o vect.o


opt: $(OBJECTS)
        $(CC) $(LFLAGS) -o opt $(OBJECTS)

chains.o: chains.c $(DIR)/misc.h $(DIR)/opt.h opts.h $(DIR)/vect.h
        $(CC) $(CFLAGS) chains.c

constprop.o: constprop.c $(DIR)/io.h $(DIR)/misc.h $(DIR)/opt.h $(DIR)/vars.h
        $(CC) $(CFLAGS) constprop.c

copyprop.o: copyprop.c $(DIR)/misc.h $(DIR)/opt.h $(DIR)/vars.h
        $(CC) $(CFLAGS) copyprop.c

deadasgs.o: deadasgs.c $(DIR)/opt.h $(DIR)/vars.h analysis.h opts.h
        $(CC) $(CFLAGS) deadasgs.c

fillslots.o: fillslots.c $(DIR)/opt.h opts.h $(DIR)/misc.h
        $(CC) $(CFLAGS) fillslots.c

flow.o: $(DIR)/flow.c $(DIR)/flow.h $(DIR)/opt.h
        $(CC) $(CFLAGS) $(DIR)/flow.c

io.o: $(DIR)/io.c $(DIR)/io.h analysis.h $(DIR)/misc.h $(DIR)/opt.h $(DIR)/vars.h
        $(CC) $(CFLAGS) $(DIR)/io.c

livedeadvars.o: livedeadvars.c analysis.h $(DIR)/misc.h $(DIR)/opt.h $(DIR)/vars.h
 opts.h
        $(CC) $(CFLAGS) livedeadvars.c

cse.o: cse.c $(DIR)/vars.h $(DIR)/misc.h $(DIR)/opt.h
        $(CC) $(CFLAGS) cse.c

loops.o: loops.c analysis.h $(DIR)/misc.h $(DIR)/opt.h
        $(CC) $(CFLAGS) loops.c

misc.o: $(DIR)/misc.c $(DIR)/misc.h $(DIR)/opt.h $(DIR)/vars.h
        $(CC) $(CFLAGS) $(DIR)/misc.c

opt.o: $(DIR)/opt.c $(DIR)/opt.h
        $(CC) $(CFLAGS) $(DIR)/opt.c

peephole.o: peephole.c $(DIR)/misc.h $(DIR)/vars.h $(DIR)/opt.h
        $(CC) $(CFLAGS) peephole.c

regalloc.o: regalloc.c $(DIR)/opt.h opts.h $(DIR)/misc.h $(DIR)/vars.h
        $(CC) $(CFLAGS) regalloc.c

reverse.o: reverse.c $(DIR)/opt.h opts.h $(DIR)/misc.h
        $(CC) $(CFLAGS) reverse.c

vars.o: $(DIR)/vars.c $(DIR)/vars.h $(DIR)/opt.h
        $(CC) $(CFLAGS) $(DIR)/vars.c

vect.o: $(DIR)/vect.c $(DIR)/vect.h $(DIR)/opt.h
        $(CC) $(CFLAGS) $(DIR)/vect.c
```