Makefile

```
HOME.DIR     = $(HOME)
OBJ.DIR      = $(HOME.DIR)/PC/Obj
BIN.DIR      = $(HOME.DIR)/PC/Bin

LEDA.DIR     = $(LEDAROOT)
LEDA.LIB.DIR = $(LEDA.DIR)
LEDA.INC.DIR = $(LEDA.DIR)/incl
LEDA.LIBS    = -lW -lP -lG -lL -lX11

C++          = g++
C++.FLAGS    = -Wall
C++.DIRS     = -I$(LEDA.INC.DIR) -L$(LEDA.LIB.DIR)
C++.LIBS     = $(LEDA.LIBS)

C            = gcc
C.FLAGS      = -g -Wall
C.DIRS       =
C.LIBS       = -lm

SYS = $(shell uname)

ifeq ($(SYS), SunOS)
 C++.LIBS += -lsocket -lthread
endif

ifeq ($(SYS), Linux)
 C++.DIRS += -L/usr/X11R6/lib
endif

C++.LIBS += -lm


%.o: %.C %.H
        $(C++) $(C++.FLAGS) $(C++.DIRS) -c $< -o $@

P: P1.o P2.o P.C
        $(C++) $(C++.FLAGS) $(C++.DIRS) $? -o $(BIN.DIR)/$@ $(C++.LIBS)

clean:
        @printf "Eliminando *.o core\n"
        @rm -f $(OBJ.DIR)/*.o core
```

P1.H

```
extern void g(int);
```

P1.C

```
#include <iostream.h>
#include "P1.H"

static int f(int);

int f(int x)
{
  return x*x;
}

void g(int i)
{
  cout << f(i) << endl;
}
```

P2.H

```
extern void z(int);
```

P2.C

```
#include <iostream.h>
#include "P2.H"

static int f(int);

int f(int x)
{
  return x / 2;
}

void z(int i)
{
  cout << f(i) << endl;
}
```

P.C

```
#include <iostream.h>
#include "P1.H"
#include "P2.H"


int main()
{
  g(3);
  z(8);
  return 0;
}
```