```
679         printf("Database %s, was successfully loaded!",dbload);
680         getch();
681         /* copys no. of records in file into master counter*/
682         add_count = dbfilecount;
683           }
684         }
685         fclose(f1);
686     }
```

The LoadDB function loads the phone book entries from a flat file. The file is opened on line 647 using fopen and the data is loaded into the room and phone arrays (lines 656...675).

```
687     /*************************************************************************

688        MAIN function
689        -------------
690        Menu, ExitMenu, drawscreen and refreshscreen.
691     *************************************************************************/
692     /*-----------------------------------------------------------------------
693        Menu function
694        -------------
695        Display valid options on the screen                                 --*/
696     ------------------------------------------------------------------------
697     char menu(void)
698     {
699     char optrtn;
700          clrscr();
701          window(1,1,80,25); /*Set position and screen mode*/
702          refreshscreen();
703          drawscreen();
704          gotoxy(1,4);
705          printf("[1] - Add entry\n");
706          printf("[2] - Delete entry\n");
707          printf("[3] - Find room number\n");
708          printf("[4] - Find phone number\n");
709          printf("[5] - List all entries\n");
710          printf("[6] - Display total entries in database\n");
711          printf("[7] - Sort entries\n");
712          printf("[8] - Load database from file\n");
713          printf("[9] - Exit");
714          gotoxy(1,25);
715          cprintf("Please select an option between 1 and 9.");
716          gotoxy(1,15);
717          printf("Database loaded: %s",dbload);
718          gotoxy(1,14);
```

```
      printf("Select an option: ");
      optrtn = getch();
      return optrtn;
}
/*-------------------------------------------------------------------
   ExitMenu function
   -----------------
   While exiting to system, askes user if he/she wants to save
   database into a file.
-------------------------------------------------------------------*/
void exitmenu(void)
{     char filename[20],save_opt;
      int k;
      FILE *fl;
      gotoxy(1,6);
      printf("Do You want to Save database before exiting? ");
      gotoxy(1,25);
      cprintf("Press 'Y' to confirm, anykey to cancel.");
      save_opt = getch();
      flushall();
      if (save_opt == 'y' || save_opt == 'Y')
      {  gotoxy(1,8);
    printf("Please Enter the path and filename to save to:");
    gotoxy(1,10);
    printf("Example: c:\\mydbfile.txt");
    gotoxy(48,8); /* move cursor back to line 8 */
    gets(filename);
    flushall();
    fl = fopen (filename,"a"); /*open file for appending mode */
    if (fl== NULL)
    {  gotoxy(1,12);
       fprintf(stderr, "Error opening file %s.",filename);
       gotoxy(1,25);
       cprintf("Database was not saved!
");

       getch();
       exit;
    }
    else
    {   for (k=0; k < add_count; k++)
        {   fprintf(fl, "%d\t%ld\n",room[k],phone[k]);}
         fclose(fl);
         gotoxy(1,25);
         cprintf("Database was successfully saved in %s",filename);
         getch();
       }
```

```
        }
        else
        {   gotoxy(1,25);
              cprintf("Database was not saved!
");
       getch();
        }
        clrscr();
        gotoxy(23,10);
        printf("Thank you for using this program");
        gotoxy(23,11);
        printf("Coded by: Jude D'souza!");
        gotoxy(23,13);
        printf("Email: shrewdjackal@yahoo.com");
        getch();
        exit(0);
}
/*-----------------------------------------------------------------------
      Drawscreen function
      --------------------
      Draws program header.
---------------------------------------------------------------------*/
void drawscreen(void)
{
        gotoxy(1,1);
         cprintf("---------------------------------------------------------
------------");
        gotoxy(1,2);
        cprintf("                              *** PHONE BOOK ***
");
        gotoxy(1,3);
         cprintf("---------------------------------------------------------
------------");
}
/*-----------------------------------------------------------------------
      Refreshscreen function
      ----------------------
      used to refresh colour display.
---------------------------------------------------------------------*/
void refreshscreen(void)
{       clrscr();
        textcolor(WHITE);
        textbackground(BLACK);
        gotoxy(1,25);
        cprintf("
");
        clrscr();
        textcolor(WHITE);
        textbackground(BLUE);
        gotoxy(1,25);
        cprintf("
");
}

/* EOF */
```

The above functions are used to draw the menu and the exit message on the screen. The ExitMenu function performs the task of saving the data to a flat file before closing the application.

I hope the above case study has been useful to you and will enable you to write applications in C. You could work upon this *Phone book* application and incorporate more features. Try using link lists and binary trees to store the Phone/Room numbers instead of arrays. Remember the saying 'Practice makes perfect'. Happy programming...

Bibliography

Barkakati, N., *Microsoft C Bible*, SAMS, 1990.
Barker, L., *C Tools for Scientists and Engineers*, McGraw-Hill, 1989.
Berry, R. E. and Meekings, B.A.E., *A Book on C*, Macmillan, 1987.
Hancock, L. and Krieger, M., *The C Primer*, McGraw-Hill, 1987.
Hunt, W. J., *The C Toolbox*, Addison-Wesley, 1985.
Hunter, B. H., *Understanding C*, Sybex, 1985.
Kernighan, B. W. and Ritchie, D. M., *The C Programming Language*, Prentice-Hall, 1977.
Kochan, S. G., *Programming in C*, Hyden, 1983.
Miller, L. H. and Quilici, E. A., *C Programming Language: An Applied Perspective*, John Wiley & Sons, 1987.
Purdum, J. J., *C Programming Guide*, Que Corporation, 1985.
Radcliffe, R. A., *Encyclopaedia C*, Sybex, 1990.
Schildt, H., *C Made Easy*, Osborne McGraw-Hill, 1987.
Schildt, H., *Advanced C*, Osborne McGraw-Hill, 1988.
Tim Grady, M., *Turbo C! Programming Principles and Practices*, McGraw-Hill, 1989.
WSI Staff, *C User's Handbook*, Addison-Wesely, 1984.
Wortman, L.A., and Sidebottom, T.O., *The C Programming Tutor*, Prentice-Hall, 1984.

Index

#define 7-9, 424
#elif 432
#else 430
#endif 430
#error 432
#if 432
#ifdef 430
#ifndef 430
#Include 10, 12, 428
#pragma 432
#undef 428

a mode 372
a+ mode 372
actual arguments 255, 262
actual parameters 255, 259
address of an array 274
address of variable 40, 334
address operator 40, 269, 339
algorithm 439
ALGOL 1
AND operation
  bitwise 60
  logical 55
ANSI 2
append mode 372
append/read mode 372
argc parameter 386
arguments 3,10, 253
arguments, command line 386
arguments in macros 426
argv parameter 386
arrow operator 359
ASI library functions 460
arithmetic expressions 62
arithmetic operators 51
arrays 180, 345
  bounds 183
  character 217
  declaring 183
  elements of 181
  initializing 185
  multidimensional 181,197, 277
  one-dimensional 183
  of pointers 351
  of structures 310
  pointers to 346
  ragged 352
  size of 183
two-dimensional 189, 193
arrow operator 359
ASCII 2-8, 228, 229, 233, 459
assignment operator 38, 56
  shorthand 56
assignment statement 7, 38
associativity of operators 70
atoi function 229
auto 37, 280, 288
automatic type conversion 67
automatic variables 279

B language 1
backslash 23, 28
backspace 28
backward jump 132
BASIC 2
BCPL 1
binary operators 453
binary search 189, 216
binary search 189, 216, 369
binomial coefficients 168
bits 31, 32, 321

bit field 321
bitwise AND 60, 453, 454
bitwise complement operator 457
bitwise EXCLUSIVE OR 60, 453, 456
bit-level programming 453
bitwise operators 60, 453, 456
bitwise OR 60, 453, 455
bitwise shift operators 60, 456
blank space 23
block statement 288
body
  of function 3
  of loop 146
bound of an array 183, 223, 447
brace 3
branching 110
break statement 126, 159
  in loop 159
  in switch statement 126
bubble sorting 188, 200
bytes 319, 334, 345

call by address 270, 353
call by pointer 279
call by reference 270, 352
call by value 279, 352
calling a function 251, 255, 257
calling program 251
calloc function 392, 394
casting 69, 358
case labels 125
character arrays 217
character constants 25, 28
character functions 83
character set 22, 23
character strings 28, 89, 217, 349
comma operator 60
command line arguments 386
comments 3
comparing strings 231, 233
compatibility 358
compilation 15
  conditional 429
compiler control directives 429
compound statement 288
concatenating strings 230, 232
conditional compilation 429
conditional expression operator 59
conditional operator 59, 129
constant 24, 25, 44
constant identifier 44
constants 22, 25
  backslash 28
  character 25, 28
  enumeration 36
  hexadecimal 26
  integer 25
  numeric 25, 27
  octal 25
  real 25-27
  strings 25, 28
  symbolic 8, 4-3
  continue statement 164
control statements 110, 146
control strings 40, 85, 94, 377
control structure 146, 439
control-d character 373
control-z character 373
conversion of type 67-69
copying strings 224, 233
counter controlled loops 147
ctype.h file 83, 461

dangling problem 121
data errors 15
data 16, 22
data structures 180, 181, 391
data types 30
  character type 33
  conversion of 67
  floating point type 32
  integer type 31
  range of 31, 32
  void types 33
debugging 448, 450
decimal integer 7
decimal notation 27
decision making 110
decision statement 55
declaration of
  arrays 183, 189
  functions 252
  pointers variables 337
  storage class 36
  string variables 218
  structures 303
  variables 33
decrement operator 58
default values 35
default statement 122, 126

definite repetition loop 147
definition of functions 251, 252
De Morgan's rule 115
dereferencing operator 340
discriminant 76
division, integer 52
documentation 440
do statement 150
dot operator 305
do..while loop 150
double 30-35
double precision 32
dynamic arrays 198
dynamic data structure 391
dynamic memory allocation 198, 391
dynamic memory management 391

edit 15
editor 14
elements of arrays 181
ellipsis 265
else if ladder 122
end of file 373
entry control 146, 148, 152
enum 36
EOF 373
errors
  common programming 441
  computational 66
  in input 92
  latent 449
  logic 15, 449
  overflow 66
  runtime 449
  semantics 16
  syntax 15, 16, 449
  underflow 66
escape sequences 28
exit control 146, 150
exit from a loop 159-163
explicit conversion 68
exponent 27
exponential notation 27
exponentiation 52
expressions
  arithmetic 62
  compound 56
  conditional 126
  integer 52
  logical 56
  pointer 343
  relational 54
  switch 125
extern 37, 284, 288, 290
external declaration 284
external variables 37, 279, 281

factorial 272
fclose function 372
feof function 377
ferror function 377
fibonacci numbers 178
field width specifier 85, 95, 97
FIFO structure 412
file management 370
FILE type 371
FILE pointer 371
files 14-17, 370
  closing 372
  defining 371
  end of 373
  mode 372
  opening 371
  rewinding 382
float 6, 7
floating point numbers
  decimal notation 27
  scientific notation 27
flowchart 112, 116, 119, 123, 439
Floyd's triangle 143
forced exit 163
fopen function 371
formal parameters 253, 259
format string 85
for loop 152
formal arguments 253, 262
formatted input 85
forward jump 132
fprint 377
free function 392, 395
free memory 392
fscan 377
fseek function 381-383
ftell function 381-383
functions 9, 10, 12, 247
  calling 255
  calling through indirection 356
  declaring type of 252, 257
  definition of 252
  nesting of 271

pointers to 355
prototype 252
returning pointer 355
type specifier 252, 255
with arrays 273
function body 253
function call 251, 255, 257
function declaration 252, 257
function definition 251
function header 252
function prototype 252, 257
function type 252

generic pointers 358
getc function 373
getchar function 81, 222
gets function 222
getw function 375
global prototype 258
global variables 13, 279, 281, 392
grammar 22
goto statement 132, 145, 159

heap 392
hierarchical structure 438
hierarchy (of operators) 64
high-level I/O operation 370
histogram 169

identifiers 24, 25
if statements 110-125
if..else statement 115-124
nesting of 118
implicit type conversion 67
increment operator 58
indentation 125
index 180
indirection operator 340, 341
information 22
infinite loop 133, 147, 156
initialization of
arrays 185, 193
arrays of structure 310
control variable 152
pointer variables 338, 402
strings arrays 218
structures 306
variables 39
input-output header file' 81
inputting
characters 81
data from key board 40
real numbers 88
integer numbers 85
strings 89
insortion sort 188
int 5-7
int number, inputting 85
integer arithmetic 52
integer constant 25
integer division 52
integer expression 52
integer numbers
input of 85
output of 95
size of 31
internal variables 279
isalpha function 82
isdigit function 82
islower function 84
ISO 2

jump
backward 132
foreward 132
jumps in loops 159
justification
left 96, 99
right 96, 99

keywords 7, 24
K& R C 1

label
case 125
goto 132
latent errors 449
left justification 96, 99
lifetime 289
life time of variables 279, 287
LIFO structure 413
linked list 181, 391, 397-401
advantages of 400
application of 412
circular 401
circular doubly 401
creating 404
deleting an item from 400, 410
inserting an item into 400, 407, 413
two-way 401

linker 17
linking 15
local prototype 258
local variables 37, 264, 392
logical AND operation 55
logic (al) errors 15, 449
logical expression 56
logical NOT operation 55
logical operators 55, 453
logical OR operation 55
long 20
long double 30-32
long float 30-32
long int 30-32
longevity 279
longevity of variables 279
loop
  control 146
  entry controlled 146
  exit controlled 146
  jumping out of 159
  skipping a part of 163
low-level I/O operations 370

macros 424-428
  arguments with 426
  nesting 427
  parameters with 426
  parentheses in definitions 426-428
  undefining 428
macro call 426
main function 3, 5, 247, 386
  arguments to 386
malloc function 392
mantissa 27
masking 457
math.h file 10, 73, 80, 461
math library 10, 72, 80
median 200
mean 203, 274
member operator 305, 359
member selection operator 359
members of structures 302
members, union 319
memory call 334
memory layout 196
memory management functions 198
minus, unary 52
mixed mode arithmetic 54
  mode 311

modules 251
modular chart 293
modulus operator 52-54
modular programming 251
MS-DOS 17
multidimensional arrays 181
multifile program 289
multifunction program 248
multiple indirections 342

nesting of
  blocks 288
  for loops 156
  functions 271
  if...else statements 118
  macros 427
  structures 314
newline character 4, 28
node 398
NOT operation, logical 55
null character 29, 219, 447
Null pointer 380, 403
null statement 156, 163
null terminator 184, 218
number
  floating point 27
  hexadecimal 25
  integer 25, 95
  octal 25
  real 26, 96
numeric constants 25

object code 15
opening a file 317
operating systems 1, 14
operands 52
operators 51
  address 40, 269
  arithmetic 51
  arrow 359
  assignment 56
  associativity of 70
  comma 60
  bitwise 60, 453, 456, 457
  conditional 59, 129
  decrement 58
  dereferencing 340
  increment 58
  indirection 269
  logical 55

order of evaluation 64
precedence of 64
relational 54
shift left 60
shift right 60
shorthand 56
sizeoff 61
ternary 59, 129
unary minus 52
unary plus 52
OR operation
bitwise 60, 455
logical 55
output parameters 269
outputting
characters 99
integers 95
mixed data 100
real numbers 96
strings 99, 224
overflow of data 45, 66

parameters 253, 255
parameter list 253
parantheses 64, 425, 448
Pascal triangle 215
pass by address 270, 274
pass by pointer 270, 274, 279, 353
pass by value 279
passing arrays 273, 276
passing strings 278
period operator 305
permanent storage area 392
pointers 270, 333, 349, 352, 355, 358
arithmetic with 343
array of 351
declaring 337
expressions 343
incrementing 344
initializing 338
passing as parameters 352
to arrays 346
to functions 355
to structures 358
variables 335, 337, 338
variables, declaring 337
variables, initializing 338
pointer constants 335
pointer values 335
pointer variable 198, 335, 337
preprocessor 9, 12, 17
precedence of operators 64, 70
pretest 146
prototype 252, 258
printf function 4, 94
posttest 146
pseudocode 439
push-down list 412
putc function 373
putchar function 84, 228
puts function 228
putw function 375

quadratic equation 74
qualifiers 26
queue 181, 412

r mode 372
r+ mode 372
random access 381
read/write mode 372
reading
characters 81
characters from file 373
from key board 40
integer 85
integer from file 375
mixed data 91
real numbers 88
strings 89, 219
text 221
real arithmetic 53
real constant 25-27
real numbers , 26
input of 88
output of 96
realloc function 392, 396
recursion 272
reference, passing by 270
register 37
register variables 279, 287
relational expression 54, 56
relational operators 54
remainder 52
return 253, 254
returning a value 253-261, 265-270
rewinding a file 381
right justification 96, 99
roots of a quadratic equation 75
runtime errors 441, 449

scanf function 40, 85, 92
  control string for 85
  skipping value in 86
scale factor 344
scope 279, 289
scope of function 258
semantic errors 16
searching 188
selection sort 188, 216
self-referential structure 398
semantic errors 16
sentinel controlled loops 147
sentinel loops 147
sequential search 189
scientific notation 27
scope of variables 279, 287
sizeof operator 61, 320, 345
skipping values in scanf 86
slack bytes 309
source program 15
sorting 188, 200
stack 181, 392, 412
standard deviation 202, 274
standard I/O header file 10, 80
standard input 81
standard output 81
static 37, 288
static arrays 198
static memory allocation 198
static variable 279, 288
stdio.h file 10, 80, 223, 462
stdlib.h file 462
storage class 36, 279, 287
strcat function 232
strcmp function 233
strcpy function 233
stringizing operator 432, 434
strings 217, 349
  comparing 231, 233
  concatenating 230, 232
  constant 25, 28
  copying 224, 233
  inputting 89
  outputting 99
  reading 89
  sorting 238
  writing 99
string.h file 235, 463
string variable 218
strlen function 234
struct 302, 303, 321
structure elements 302
structure members operator 305
structure tag 302
structure variables 302, 303
structures
  arrays of 310
  arrays within 313
  declaring 303
  initializing 306
  members 302, 304
  nesting 314
  passing as parameters 316
  self-referential 398
  size of 320
  tags 302
  templates 302
  variables 303, 306
structured programming 163
single-entry 251
single-exit 251
size of structures 320
subroutine 9
subscripted variables 182
subscripts 180
switch statement 125
system library 15
symbolic constant 8, 43
symbolic name 43
syntax error 15, 16, 449
syntax rules 22

tag of structure 302
target variable 38
template of structure 302
ternary operator 59
test condition in
  do loop 150
  for loop 152
  if...else statement 115
  while loop 148
test expression in
  if statement 111
  if...else statement 115
testing 449
test editor 14
time delay loop 156
time.h file 463
token pasting operator 432, 434
tokens 24

tolower 84
toupper 84
top-down programming 248
trees 181
trigraph 23
truth table 56
two dimensional arrays 181, 277
type specifier 252, 255
typedef 35

unary minus 52
unary plus 25, 52
undefining a macro 428
underflow of data 45, 66
union 301, 319
UNIPLUS SYSTEM 16
UNIX 1, 14, 225
unsigned int 31, 32
user-defined function 247

variable field width 225
variables 22, 29
  assigning values to 7, 38
  automatic 37, 279, 280
  declaration of 6, 33
  external 37, 279, 281
  global 13, 37, 279, 281
  initialization of 39
  internal 279
  local 37
  longevity of 279
  names 6
  pointer 270
  register 37, 279, 287
  scope of 279
  sentinal 147
  static 37, 279
  string 218
  structure 302, 303
  subscripted 182
variance 203
visibility 279
void 5, 253, 258, 260
volatile 45

w mode 372
w+ mode 372
while loop 8, 147
white space 23
width of field, specifying 85, 95, 97, 225
word boundaries 309
write mode 372
write/read mode 372
writing
  character 84
  character to file 373
  integer to file 375
  real numbers 96
  strings 99, 224