



---

**Index**

---



### Symbols

**& (address of) operator, 45**  
**"" (double quotes) around include files, 206**  
**\* (asterisk)**  
    dereference operator, 37  
    in declarations, 45  
    indirection operator, 55  
**+ (addition) operator, 217**  
**- (subtraction) operator, 220**  
**-> (indirect membership) operator, 55**  
**. (member) operator, 55**  
**<> characters around include files, 206**  
**64K limit, memory models, 760**

### A

**A\_CURSOR.C source file, 117-118**  
**A\_CURSOR.H header file, 118**  
**about boxes, 518-521**  
**ABOUT.C source file, 519-520**  
**accelerated systems analysis and design, 364**  
**accelerator keys, 482**  
    for general help, 514-515  
    help for, 518  
**accessing**  
    memory directly, 231-232  
    orders  
        *FIFO (First In First Out), 87-88*  
        *in-order, 90-99*  
        *LIFO (Last In First Out), 83-87*  
        *post-order, 90-99*  
        *pre-order, 90-99*  
    video memory  
        *ANSI functions, 105-128*  
        *directly, 128*  
        *through BIOS, 128-129*  
**AccSys (Copia International)**  
    for dBASE, 748  
    for Paradox, 747  
**Accusoft Image Format Library (Accusoft Corporation), 746**  
**action bars, 482**  
    adding functions, Medium Code entry and edit screen, 588-601  
    multi-menu, 492-506  
    Musical Items entry and edit screen, 374, 650-661

    single (main) menu, 488-492  
    standards for menu items, 483-487  
**add\_album\_rec() function, 635-640**  
**add\_mdata() function, 564-565**  
**add\_medium\_rec() function, 566-573**  
**add\_new\_songs() function, 634-635**  
**add\_to\_list() function, 81-82**  
**ADDAREC.C source file, 636-640**  
**addition (+) operator, 217**  
**ADDMREC.C source file, 566-570**  
**address of (&) operator, 45**  
**addresses of variables, finding, 45-47**  
**ALBMABAR.C source file, 650-661**  
**ALBUM\_INDEX structure, 604**  
**ALBUM\_REC structure, 400**  
**ALBUMS matrix, 377-378**  
**ALBUMS.C source file, 425-437, 605-623**  
**aligning words in structures, 40-41**  
**allocating memory**  
    at compile time, 6-9  
    dynamically, 9-26  
**alpha testing, 718**  
**alt key, standards for navigating entry and edit windows, 388**  
**ampersand (&) (address of) operator, 45**  
**ANSI (American National Standards Institute), 105**  
    escape sequences, 231  
        *codes, 106-112*  
        *components, 106*  
        *for erasing screens, 115-121*  
        *for moving cursor, 113-121*  
        *for redefining keys, 126-127*  
        *for screen colors, 121-126*  
        *pros and cons, 105-106*  
    standards, 139  
        *case sensitivity, 140-141*  
        *character sets, 142*  
        *classifying variables, 151-156*  
        *converting case, 156-157*  
        *enforcing compatibility, 142-143*  
        *header files (listing), 782-783*  
        *keywords, 139-140*  
        *numeric variables, 143-151*  
        *predefined constants, 159-162*  
        *preprocessor directives, 159*  
        *structures and unions, 157-159*

**ANSI.SYS driver, 105-106**  
**ANSICLRS.H header file, 122**  
**Array size too large error message, 9**  
**ARRAY.C source file, 35-36**  
**arrays, 35-38**  
     of structures, 55-58  
**arrow keys**  
     left, `getline()` function, 320  
     Medium Code entry and edit screen, 411-412  
     menu scan codes, 459  
     right, `getline()` function, 320  
     standards for navigating entry and edit screens, 387  
**ascending sorts, 49-52**  
**ASCII (American Standard Code for Information Interchange)**  
     Character Table, displaying values, 170-176  
     chart, 762-769  
     extended character sets, 127-128  
**assembler language, 740-741**  
**assert() macro, 726-729**  
**ASSERT.C source file, 727-728**  
**ASSERT.H header file, 782**  
**asterisk (\*)**  
     dereference operator, 37  
     in declarations, 45  
     indirection operator, 55  
**auto keyword, 772**  
**Autumn Hill Software's FONT-TOOLS, 746**

## B

**background**  
     colors  
         *ANSI codes, 107*  
         *escape sequences for, 121-126*  
     grids, 338-341  
**backspace key**  
     `getline()` function, 319  
     standards for navigating entry and edit windows, 388  
**bases, number systems, 179**  
**BDATE.C source file, 212-213**  
**beeping, 321-322**  
**beta testing, 718-719**  
**BIGCURS.C source file, 242**  
**binary mode, 534**

**binary number system, 183-189**  
     ASCII chart, 762-769  
     converting to and from hexadecimal, 191-192  
     translating characters into, 176-179  
**binary trees, 89-99**  
**BINDUMP.C source file, 184-186**  
**BIOS (Basic Input/Output System), 128-129, 232-237**  
     compiler-specific functions, 256-257  
         *versus creating functions, 257*  
     keyboard functions, 253-255  
     moving cursor, 237-241  
     sizing cursor, 241-243  
     software interrupts  
         *creating, 255-256*  
         *listing of, 244-252*  
**BIOSREGS.H header file, 235**  
**birthdates, verifying entries, 222-223**  
**Blaise Computing, Inc.'s C Asynch Manager, 747**  
**BOOL data type, 34**  
**boop() function, 321-322**  
**BOOP.C source file, 321-322**  
**borders of screen, colors, 275-276**  
**Borland**  
     `clrscr()` function, 336-337  
     compiler-specific functions  
         *BIOS, 256-257*  
         *far memory, 22*  
     directvideo variable, 128  
     TLIB library program, 209  
**Bounds Checker program, 732**  
**box() function, 120-121, 125, 283-288**  
**BOX.C source file, 284-285**  
**boxes, drawing, 283-288**  
**break keyword, 772**  
**Btrieve (Novell, Inc.), 748**  
**bugs**  
     debugging, 722-732  
     logic errors, 721  
     recursive errors, 722  
     syntax errors, 719-721  
**business rules and constraints, planning computer programs/systems, 357**  
**BYTE data type, 34**  
**BYTEREGS structure, 233-235**  
**bytes, 188**  
     Null terminator, 7



## C Asynch Manager

### C

**C Asynch Manager (Blaise Computing, Inc.), 747**

**“C” EMM Library (SilverWare, Inc.), 748**

**.C file extension, 202**

**C Utility Library (South Mountain Software, Inc.), 748-749**

**C++ language, 742-744**

**calc\_nbr\_songs() function, 635**

**calculating songs, 628-635**

**calloc() function, 17-21**

**capturing data, 411-413, 424-425**

    Musical Items entry and edit screen,  
    625-626

**carriage-return key, Medium Code entry  
and edit screen, 411**

**case**

    converting, portability, 156-157  
    sensitivity, portability, 140-141

**case keyword, 772**

**CDOWN.C source file, 114**

**char data type, 32**

    maximum/minimum values, 33

**char keyword, 772**

**CHAR\_BIT constant, 146**

**CHAR\_MAX constant, 146**

**CHAR\_MIN constant, 146**

**char\_to\_binary() function, 187-189**

**character arrays versus strings, 37-38**

**character graphics, 265-266**

    drawing

*boxes, 283-288*

*lines, 280-281*

    repeating characters, 278-279

    screen colors

*borders, 275-276*

*text, 276-278*

    video mode, getting/setting, 273-275

**character sets**

    extended, 265-266

*ASCII, 127-128, 762-769*

    portability, 142

**character variables, 143**

**characters**

    displaying in number-system formats

*binary, 183-189*

*decimal, 181-183*

*hexadecimal, 190-194*

*octal, 194-196*

    entering for getline() function, 318-319

    null, 38

    translating into multiple number

        systems, 176-179

    values, ANSI codes, 109-112

**clear\_albums\_fields() function, 625**

**clear\_eol() function, 116**

**clear\_medium\_fields() function, 410**

**clear\_screen() function, 116, 125,  
335-338**

**CLEAREOL.C source file, 116**

**clearing**

    screens, 335-338

*escape sequences for, 115-121*

    structure fields, 410

**CLEARSCN.C source file, 336**

**CLEFT.C source file, 114**

**clock (system), returning tick counter**

    value, 323-324

**close\_files() function, 548-549**

**CLOSEFILE.C source file, 548-549**

**closing files, 548-549**

**clrscr() function, 336-337**

**CLRSCRN.C source file, 116**

**CodeBase (Sequiter Software, Inc.), 748**

**codes**

    ANSI escape sequences, 106-112

    assembler, 740-741

    commercial, 744-749

    debugging, 722-732

    errors, 719-721

    logic errors, 721

    scan, 318

*menus, 458-459*

    writing with white space, 135-138

**color\_table structure, 400-401, 452-454**

**colored text, 264**

**colors**

    screen borders, 275-276

    screen text, 276-278

    screens

*ANSI codes, 106-107*

*escape sequences for, 121-126*

    strings, 282-283

    text, 303-304

**COLORTBL.H header file, 452-453**

**.COM file extension, 202**

**comma delimited files, 535-536**

**command line parameters, startup help  
for, 511-514**

**communications libraries, 747**

**compact memory model, 757-758**  
**comparing memory models, 759-760**  
**compatibility with ANSI, enforcing for portability, 142-143**

**compilers**

Borland

*BIOS functions, 256-257*

*far memory functions, 22*

Microsoft

*BIOS functions, 256*

*far memory functions, 22*

non-standard header files, 783-784

writing, 739

**computer systems**

design methodologies

*program specifications, 365-366*

*prototyping, 364-365*

*rapid or accelerated systems analysis and design, 364*

*Structured Systems Analysis and Design, 355-363*

versus computer programs, 354

**const keyword, 772**

**constants**

CHAR\_BIT, 146

CHAR\_MAX, 146

CHAR\_MIN, 146

DBL\_DIG, 147

DBL\_MAX, 147

DBL\_MAX\_10\_EXP, 147

DBL\_MAX\_EXP, 147

DBL\_MIN, 148

DBL\_MIN\_10\_EXP, 148

DBL\_MIN\_EXP, 148

ELEMENT, 74

FLT\_DIG, 147

FLT\_MAX, 147

FLT\_MAX\_10\_EXP, 147

FLT\_MAX\_EXP, 147

FLT\_MIN, 147

FLT\_MIN\_10\_EXP, 147

FLT\_MIN\_EXP, 147

INT\_MAX, 147

INT\_MIN, 147

LDBL\_DIG, 147

LDBL\_MAX, 148

LDBL\_MAX\_10\_DBL, 148

LDBL\_MAX\_EXP, 148

LDBL\_MIN, 148

LDBL\_MIN\_10\_EXP, 148

LDBL\_MIN\_EXP, 148

\_\_LINE\_\_, 12

LIST, 72

LISTLINK, 72

LONG\_MAX, 147

LONG\_MIN, 147

\_MSC\_VER, 164

portable non-ANSI, 162-164

predefined, portability, 159-162

SCHAR\_MAX, 146

SCHAR\_MIN, 146

SHRT\_MAX, 146

SHRT\_MIN, 146

\_\_STDC\_\_, 142

UCHAR\_MAX, 146

UINT\_MAX, 147

ULONG\_MAX, 147

USHRT\_MAX, 146

*see also* keywords

**constraints in business, planning**

**computer programs/systems, 357**

**construction of computer programs/systems, 361-362**

**context-sensitive help, 521-529**

**continue keyword, 772**

**controls**

text, ANSI codes, 107-108

video mode, ANSI codes, 108

**conv\_to\_upper() function, 156-157**

**convert\_str() function, 684-685**

**converting binary numbers to and from hexadecimal, 191-192**

**CONVSTR.C source file, 684-685**

**Copia International's AccSys**

for dBASE, 748

for Paradox, 747

**CPUT.C source file, 113**

**CRIGHT.C source file, 114**

**ctype.h header file, 782**

**CUP.C source file, 113**

**current date, 236-237**

**CURSOFF.C source file, 332-333**

**CURSON.C source file, 333-334**

**cursor**

displaying, 333-335

hiding, 332-333

moving, 269-272

*BIOS functions, 237-241*

*escape sequences for, 113-121*

sizing, BIOS functions, 241-243



## cursor() function

**cursor() function, 238-241, 269-271**

**cursor\_off() function, 332-335**

**cursor\_on() function, 333-335**

**custom file formats, 536**

## D

### data

capturing, 411-413, 424-425  
*Musical Items entry and edit screen,*  
625-626

displaying, Xs versus 9s, 668

**data files, 533**

*see also* files

**data trees, binary, 89-99**

**data types**

arrays of structures, 55-58

basic, 32-33

creating, 34

ELEMENT, 74

grouping, 35

*arrays, 35-38*

*structures, 38-41*

*unions, 41-44*

LIST, 72

LISTLINK, 72

pointers to structures, 52-55

variable-length structures, 58-63

**databases, 533**

components, 532-533

**DATE structure, 400**

**DATE.C source file, 211-212**

**\_\_DATE\_\_ predefined constant, 160**

**dates**

birthdates, verifying entries, 222-223

current, 236-237

**DBL\_DIG constant, 147**

**DBL\_MAX constant, 147**

**DBL\_MAX\_10\_EXP constant, 147**

**DBL\_MAX\_EXP constant, 147**

**DBL\_MIN constant, 148**

**DBL\_MIN\_10\_EXP constant, 148**

**DBL\_MIN\_EXP constant, 148**

**deallocating (freeing) memory bytes,**  
12-15

**debugging, 722**

debugger programs, 731-732

print statements, 723-726

walk-throughs, 722-723

**DECUMP.C source file, 182-183**

**decimal number system, 179-181**

ASCII chart, 762-769

displaying characters in number-system  
formats, 181-183

translating characters into, 176-179

**declaring**

pointers, 44-45

structures, 38

*variable-length, 59*

type definitions, 34

unions, 41

**default keyword, 772-773**

**defaults in getline() function, 317**

**del\_alb\_rec() function, 641-646**

**del\_med\_rec() function, 573-579**

**DELAREC.C source file, 641-645**

**delete key**

getline() function, 319

standards for navigating entry and edit  
windows, 388

**deleting**

functions from libraries, 220

links from linked lists, 82

records

*from Medium Code entry and edit*  
*screen, 578-579*

*from Musical Items entry and edit*  
*screen, 641-646*

**DELMREC.C source file, 573-577**

**dereference (\*) operator, 37**

**descending sorts, 49-52**

**design methodologies, *see***  
**methodologies**

**detailed checklist testing, 715**

**Detailed Information reports, creating,**  
691-709

**device drivers, ANSI.SYS, 105-106**

**digits**

binary, 188

decimal, 180-181

hexadecimal, 190

**direct memory access, 128, 231-232**

**directives (preprocessor),**

**portability, 159**

**directvideo variable, 128**

**display, *see* video display**

**display\_about\_box() function, 519-521**

**display\_cntxt\_help\_msg() function, 527**

**display\_context\_help() function,**  
521-527

**display\_groups\_fields() function,**  
423-424

**display\_header() function, 98**  
**display\_medium\_fields() function, 410-411**  
**display\_medium\_help() function, 516-517**  
**display\_menu() function, 98, 456-458**  
 headers, 452-454  
 listing of, 445-452  
 parameters, 454-456  
 scan codes, 458-459  
 using, 460-464  
**display\_msg\_box() function, 401-402**  
**do keyword, 773**  
**do\_main\_menu() function, 401**  
**do\_medium\_actionbar() function, 494, 504-506**  
**do\_medium\_menu2() function, 599**  
**do\_medium\_menu3() function, 600-601**  
**do\_medium\_screen() function, 410**  
**do\_type\_of\_music\_menu() function, 472-475**  
**documentation, 528-529**  
 of designing computer programs/systems, 363  
 tutorials, 527-528  
**double data type, 32**  
 maximum/minimum values, 33  
**double keyword, 773**  
**double quotes (“”) around include files, 206**  
**double variables, 144**  
**double-linked lists, 88-89**  
 binary trees, 89-99  
**down arrow key**  
 Medium Code entry and edit screen, 411  
 menu scan codes, 459  
 standards for navigating entry and edit windows, 387  
**draw\_borders() function, 401**  
**draw\_groups\_prompts() function, 423**  
**draw\_medium\_prompts() function, 410**  
**draw\_medium\_screen() function, 410-411, 495**  
**DRAWBOX.C source file, 286**  
**drawing**  
 boxes, 283-288  
 lines, 280-281

prompts  
*field, 410*  
*group, 424*  
**dump\_inorder() function, 98-99**  
**dump\_postorder() function, 98-99**  
**dump\_preorder() function, 98-99**  
**DWORD data type, 34**  
**dynamic memory allocation, 9**  
 allocating memory bytes, 10-12, 17-21  
 deallocating (freeing) memory bytes, 12-15  
 far (over 64K), 21-26  
 initializing and allocating memory bytes, 17-21  
 reallocating (resizing) memory blocks, 15-17

## E

**E-MEM (TeraTech), 748**

### edit functions

birthdate, 212-213  
 date, 211-212  
 sex code, 210-211  
 state, 209-210

**Edit menu, standards, 484-485**

**editing, *see* entry and edit screens**

**EDITS.H header file, 215**

**efficiency, programming for, 134-138**

**ELEMENT data type, 74**

**else keyword, 773**

**encapsulation, 742-751**

### end key

menu scan codes, 459  
 standards for navigating entry and edit windows, 388

### enter key

Medium Code entry and edit screen, 411  
 menus, 459  
 Musical Items entry and edit screen, 372  
 standards for navigating entry and edit windows, 387

### entry and edit screens

components, 386  
 flow of, 402  
 Group Information, 414-425  
*access requirements, 377*  
*keys, 375*





## entry and edit screens

---

- prototype screen, 375*
- Medium Code, 403-413
  - access requirements, 377*
  - adding functions to action bars, 588-601*
  - adding records, 566-573*
  - deleting records, 578-579*
  - extended help, 516-518*
  - keys, 376*
  - multi-menu action bars, 492-506*
  - processing next/previous records, 579-583*
  - prototype screen, 376*
  - updating files for, 550-566*

### menus

- temporary system, 390-402*
- Type of Music, 472-475*
- Musical Items, 425-438, 604-625
  - access requirements, 376-377*
  - action bar, 374, 650-661*
  - adding records, 635-640*
  - adding/changing and calculating songs, 628-635*
  - capturing data, 625-626*
  - deleting records, 641-646*
  - keys/function keys, 372-373*
  - page up and page down functions, 626-628*
  - processing next/previous records, 646-650*
  - prototype screen, 371*
  - updating files, 602-604*

**entry and edit standards, 386-388**

**enum keyword, 773**

**erasing screens, escape sequences for, 115-121**

**errno.h header file, 782**

### error messages

- Array size too large, 9
- Multiple declaration for 'var1' in function main, 141
- Undefined symbol '\_ \_xxxx\_' in function main, 160

### errors

- debugging, 722-732
- logic, 721
- recursive, 722
- syntax, 719-721
- system, displaying messages about, 729-731

### escape key

- Medium Code entry and edit screen, 412
- menus, 460
- standards for navigating entry and edit windows, 389

### escape sequences, 231

- codes, 106-112
- components, 106
- for erasing screens, 115-121
- for moving cursor, 113-121
- for redefining, 126-127
- for screen colors, 121-126
- pros and cons, 105-106

**Essential B-Tree (South Mountain Software, Inc.), 748**

**Essential Communications (South Mountain Software, Inc.), 747**

**estimates, preliminary, 358**

**.EXE file extension, 202**

**exit keys, getline() function, 304**

**extended character sets, 265-266**

- ASCII, 127-128

**extended help, 515-518**

**extern keyword, 773**

**extracting functions from libraries, 221**

## F

---

**far keyword, 25**

**far memory, 21-26**

**faralloc() function, 22-26**

**farfree() function, 22-26**

**farmalloc() function, 22-26**

**farrealloc() function, 22-26**

**\_falloc() function, 22-26**

**\_ffree() function, 22-26**

**field prompts, drawing, 410**

**fields, 533**

- getline() data-retrieval, underlining, 410-411
- padding with zeros, 401-402
- structure, clearing, 410

**FIFO (First In First Out) order of access, 87-88**

**file management libraries, 747-748**

**File menu, standards, 483-484**

**\_FILE\_ predefined constant, 160**

**files**

- closing, 548-549
- comma delimited, 535-536



- custom format, 536
- data, 533
- extensions
  - .C*, 202
  - .COM*, 202
  - .EXE*, 202
  - .H*, 202
  - .LIB*, 202, 217
  - .LST*, 219
  - .OBJ*, 202, 217
- flat, 535
- header, *see* header files
- I/O (Input/Output) functions,
  - preparing for, 538-540
- index, 536-537
- modes, 534
- opening, 540-543
- records
  - reading*, 543-545
  - writing*, 545-546
- source, *see* source files
- TEST, 181
- finding**
  - elements, with binary trees, 90-99
  - variable addresses, 45-47
- First In First Out (FIFO) order of access, 87-88**
- flat files, 535**
- float data type, 32**
  - maximum/minimum values, 33
- float keyword, 773**
- float variables, 144**
- float.h header file, 782**
- FLT\_DIG constant, 147**
- FLT\_MAX constant, 147**
- FLT\_MAX\_10\_EXP constant, 147**
- FLT\_MAX\_EXP constant, 147**
- FLT\_MIN constant, 147**
- FLT\_MIN\_10\_EXP constant, 147**
- FLT\_MIN\_EXP constant, 147**
- \_fmalloc() function, 22-26**
- FONT-TOOLS (Autumn Hill Software), 746**
- for keyword, 773**
- foreground colors**
  - ANSI codes, 106-107
  - escape sequences for, 121-126
- format\_name() function, 55**
- formats**
  - comma delimited files, 535-536
  - custom files, 536
  - flat files, 535
- fprintf() function, 683**
- \_frealloc() function, 22-26**
- free() function, 12-15**
- free-form testing, 715**
- function keys**
  - F10
    - main-menu case*, 491-492
    - multi-menu case*, 494-495
  - help for, 518
  - Medium Code entry and edit screen, 412-413
  - Musical Items entry and edit screen, 372-373
  - standards, 388-390
- function numbers, 233**
- functions**
  - add\_album\_rec()*, 635-640
  - add\_mdata()*, 564-565
  - add\_medium\_rec()*, 566-573
  - add\_new\_songs()*, 634-635
  - add\_to\_list()*, 81-82
  - adding to action bars, Medium Code entry and edit screen, 588-601
  - ANSI, *see* escape sequences
  - boop()*, 321-322
  - box()*, 120-121, 125, 283-288
  - calc\_nbr\_songs()*, 635
  - calloc()*, 17-21
  - char\_to\_binary()*, 187-189
  - clear\_albums\_fields()*, 625
  - clear\_eol()*, 116
  - clear\_medium\_fields()*, 410
  - clear\_screen()*, 116, 125, 335-338
  - close\_files()*, 548-549
  - clrscr()*, 336-337
  - compiler-specific, 256-257
    - versus creating functions*, 257
  - conv\_to\_upper()*, 156-157
  - convert\_str()*, 684-685
  - cursor()*, 238-240-241, 269-271
  - cursor\_off()*, 332-335
  - cursor\_on()*, 333-335
  - del\_alb\_rec()*, 641-646
  - del\_med\_rec()*, 578-579
  - display\_about\_box()*, 519-521
  - display\_cntxt\_help\_msg()*, 527
  - display\_context\_help()*, 521-527
  - display\_groups\_fields()*, 423-424
  - display\_header()*, 98
  - display\_medium\_fields()*, 410-411



## functions

display\_medium\_help(), 516-517  
display\_menu(), 98, 445-471  
display\_msg\_box(), 401-402  
do\_main\_menu(), 401  
do\_medium\_actionbar(), 494, 504-506  
do\_medium\_menu2(), 599  
do\_medium\_menu3(), 600-601  
do\_medium\_screen(), 410  
do\_type\_of\_music\_menu(), 472-475  
draw\_borders(), 401  
draw\_groups\_prompts(), 424  
draw\_medium\_prompts(), 410  
draw\_medium\_screen(), 410-411, 495  
dump\_inorder(), 98-99  
dump\_postorder(), 98-99  
dump\_preorder(), 98-99  
edit  
    *birthdate, 212-213*  
    *date, 211-212*  
    *sex code, 210-211*  
    *state, 209-210*  
farcalloc(), 22-26  
farfree(), 22-26  
farmalloc(), 22-26  
farrealloc(), 22-26  
\_fcalloc(), 22-26  
\_ffree(), 22-26  
\_fmalloc(), 22-26  
format\_name(), 55  
fprintf(), 683  
\_frealloc(), 22-26  
free(), 12-15  
get\_alb\_info(), 650  
get\_alb\_selection(), 702-709  
get\_albums\_input\_data(), 625-626  
get\_cursor(), 238-241, 270-271  
get\_groups\_data(), 424  
get\_groups\_input\_data(), 424-425  
get\_it(), 318  
get\_med\_info(), 582-583  
get\_medium\_data(), 411-413  
get\_medium\_input\_data(), 411-413  
get\_names(), 98  
get\_rec(), 543-545  
get\_space(), 99  
get\_stack\_data(), 87  
get\_timer\_ticks(), 323-324  
get\_video(), 273-274  
getline(), *see* getline() function  
grid(), 338-341  
initialize\_color\_table(), 401, 463  
insert\_list(), 82  
int86(), 237  
is\_stack\_empty(), 87  
is\_valid\_birthdate(), 212-213  
is\_valid\_date(), 211-213  
is\_valid\_sex(), 210-211  
is\_valid\_state(), 209-210  
isalnum(), 153  
isalpha(), 153  
iscntrl(), 153  
isdigit(), 154  
isgraph(), 154  
islower(), 154  
isprint(), 154  
ispunct(), 154  
isspace(), 154  
isupper(), 154  
isxdigit(), 154  
kbwait(), 254-255  
keyboard, 253-255  
keyhit(), 253-255  
libraries, *see* libraries  
list\_groups(), 685-690  
list\_medium\_codes(), 678-685  
lowercase(), 157  
main()  
    *in recofrec.c source file, 401*  
    *TESTMENU.C source file, 463*  
malloc(), 10-12, 74  
memcpy(), 627  
memset(), 628  
move\_cursor\_down(), 114  
move\_cursor\_left(), 114  
move\_cursor\_right(), 114  
move\_cursor\_up(), 113  
music\_rpt(), 702  
open\_files(), 540-543  
open\_songs(), 625  
pause(), 288-289  
perror(), 729-731  
pointers to, 48-52  
pop\_stack(), 87  
print\_group(), 690  
print\_grp\_header(), 690  
print\_med\_hdr(), 683  
printf(), 178-179  
proc\_med\_rec(), 579-583  
process\_med\_list(), 683-684  
prototypes, consolidating in header  
    files, 400  
push\_stack(), 87

put\_cursor(), 113  
 put\_rec(), 545-546  
 qsort(), 49-52  
 realloc(), 15-17, 21  
 repeat\_char(), 279-281  
 reset\_memory(), 635  
 restore\_cursor\_position(), 115  
 restore\_screen\_area(), 345-347  
 rewrite\_menu\_items(), 458-459  
 save\_cursor\_position(), 115  
 save\_screen\_area(), 341-347  
 search\_list(), 99  
 search\_med\_rec(), 600-601  
 set\_border\_color(), 275-276  
 set\_cursor\_size(), 241-243  
 set\_video(), 274-275  
 setup\_exit\_keys(), 321  
 setup\_today(), 683  
 show\_list(), 82  
 sprintf(), 338  
 switch\_globals(), 702  
 toupper(), 157  
 update\_header(), 546-548  
 update\_songs(), 635  
 updating, 271-272  
 verify\_mdata(), 565  
 waitsec(), 322-323  
 write\_char(), 276-278  
 write\_string(), 282-283, 458-459  
 yes\_no\_box(), 401-402  
 zero\_fill\_field(), 401-402, 424-425

## G

**GCURSOR.C source file, 238-239, 270-271**

### Genus Microprogramming

GX Effects, 746  
 GX Printer, 749  
 PCX Toolkit, 746

**get\_alb\_info() function, 650**  
**get\_alb\_selection() function, 702-709**  
**get\_albums\_input\_data() function, 625-626**  
**get\_cursor() function, 238-241, 270-271**  
**get\_groups\_data() function, 424**  
**get\_groups\_input\_data() function, 424-425**  
**get\_it() function, 318**  
**get\_med\_info() function, 582-583**

**get\_medium\_data() function, 411-413**  
**get\_medium\_input\_data() function, 411-413**  
**get\_names() function, 98**  
**get\_rec() function, 543-545**  
**get\_space() function, 99**  
**get\_stack\_data() function, 87**  
**get\_timer\_ticks() function, 323-324**  
**get\_video() function, 273-274**  
**getline() function, 296-297**  
     boop() function, 321-322  
     code listing, 305-321  
     entering strings, 325-327  
     get\_timer\_ticks() function, 323-324  
     parameters, 304-305  
         *option, 303-304*  
     prototype, 303  
     return value, 305  
     TYAC.H header file for, 297-302  
     waitsec() function, 322-323  
**GETLINE.C source file, 306-316**  
**GETREC.C source file, 543-544**  
**GETTICKS.C source file, 324**  
**GL\_TEST.C source file, 325-326**  
**goto keyword, 774**  
**GraphiC (Scientific Endeavors Corporation), 746**  
**Graphical User Interfaces (GUIs), 383-384**  
**graphics**

    character, 265-266  
         *drawing boxes, 283-288*  
         *drawing lines, 280-281*  
         *repeating characters, 278-279*  
         *screen border colors, 275-276*  
         *screen text colors, 276-278*  
         *video mode, getting/setting, 273-275*  
     colored text, 264  
     monochrome text, 264  
     pixel, 265  
**graphics libraries, 746**  
**graphics programming, 739**  
**Graphics-MENU (Island Systems), 746**  
**gray-scale, 264**  
**Greenleaf CommLib (Greenleaf Software, Inc.), 747**  
**Greenleaf Functions™ (Greenleaf Software, Inc.), 749**  
**grid() function, 338-341**  
**GRID.C source file, 339**



## Group Information

---

### Group Information

- entry and edit screen, 414-425
  - access requirements*, 377
  - keys*, 375
  - prototype screen*, 375
- list
  - creating*, 685-690
  - prototype*, 668-669
- matrix, 378
- GROUP\_REC structure**, 400
- grouping data types**, 35
  - arrays, 35-38
  - structures, 38-41
  - unions, 41-44
- GROUPS.C source file**, 414-422
- GUIs (Graphical User Interfaces)**, 383-384
- GVIDEO.C source file**, 273
- GX Effects (Genus Microprogramming)**, 746
- GX Printer (Genus Microprogramming)**, 749

## H

---

- .H file extension**, 202
- head pointers**, 69-70
- header files**
  - A\_CURSOR.H, 118
  - ANSI standard (listing), 782-783
  - ANSICLRS.H, 122
  - BIOSREGS.H, 235
  - COLORTBL.H, 452-453
  - EDITS.H, 215
  - including in source files, 204-207
  - LIST0701.H, 204-205
  - LIST0702.H, 205
  - LIST0914.H, 289-291
  - non-standard compiler, 783-784
  - RECOFREC.H, 396-397, 493-494, 550, 602-603, 677-678
  - RECORDS.H, 397-399, 551, 603-604
  - TYAC.H, 266-268, 539-540
    - for getline() function*, 297-302
    - updating*, 289-291
- headers, updating information in records**, 546-548
- help**, 510
  - about boxes, 518-521
  - context-sensitive, 521-529
  - documentation/manuals, 528-529

- general, 514-518
- startup, 511-514
- tutorials, 527-528

- help files, MEDIUM.HLP**, 525
- Help menu, standards**, 486-487
- HEX.C source file**, 196-198
- hexadecimal number system**, 190-194
  - ASCII chart, 762-769
  - translating characters into, 176-179
- HEXDUMP.C source file**, 192-193
- HFILE data type**, 34
- hiding cursor**, 332-333
- Hold Everything (South Mountain Software, Inc.)**, 749
- home, moving cursor to**, 113
- home key**
  - getline() function, 320
  - menu scan codes, 458-459
  - standards for navigating entry and edit windows, 388
- HREG structure**, 235-236
- huge memory model**, 758-759

## I-J

---

- I/O (Input/Output)**, 533
  - file functions, preparing for, 538-540
- if keyword**, 774
- in-order access**, 90-99
- #include preprocessor directive**, 204-207
- index files**, 536-537
  - see also* files
- index help**, 518
- indirect membership (->) operator**, 55
- indirection (\*) operator**, 55
- inheritance**, 743
- initialize\_color\_table() function**, 401, 463
- initializing allocated memory bytes**, 17-21
- insert key, getline() function**, 320
- insert\_list() function**, 82
- installation manuals**, 528
- int data type**, 32
  - maximum/minimum values, 33
- int keyword**, 774
- INT\_MAX constant**, 147
- INT\_MIN constant**, 147
- int86() function**, 237

**integer variables, 144**

**integers**

allocating memory for, 18-20  
converting to strings, 684-685

**integration testing, 362, 717**

**interactive tutorials, 527**

**interrupts, software, 129, 232**

creating, 255-256  
listing of, 244-252

**is\_stack\_empty() function, 87**

**is\_valid\_birthdate() function, 212-213**

**is\_valid\_date() function, 211-213**

**is\_valid\_sex() function, 210-211**

**is\_valid\_state() function, 209-210**

**isalnum() function, 153**

**isalpha() function, 153**

**isctrl() function, 153**

**isdigit() function, 154**

**isgraph() function, 154**

**Island Systems' Graphics-MENU, 746**

**islower() function, 154**

**isprint() function, 154**

**ispunct() function, 154**

**isspace() function, 154**

**isupper() function, 154**

**isxdigit() function, 154**

## K

**kbwait() function, 254-255**

**KBWAIT.C source file, 254**

**keyboard functions, 253-255**

**keyboard shortcuts**

character values, ANSI codes, 109-112  
escape sequences for redefining, 126-127

**keyhit() function, 253-255**

**KEYHIT.C source file, 253-254**

**keys**

accelerator, 482  
*for general help, 514-515*  
backspace, getline() function, 319  
character values, ANSI codes, 109-112  
delete, getline() function, 319  
exit, getline() function, 304  
function  
*F10 main-menu case, 491-492*  
*F10 multi-menu case, 494-495*  
*standards, 388-390*  
Group Information entry and edit screen, 375

home, getline() function, 320

insert, getline() function, 320

left arrow, getline() function, 320

Medium Code entry and edit screen, 376

mnemonic, 444-445

Musical Items entry and edit screen, 372-373

navigating menus, 444-445

non-scan, menus, 459-460

page down, Musical Items entry and edit screen, 626-628

page up, Musical Items entry and edit screen, 626-628

redefining, escape sequences for, 126-127

right arrow, getline() function, 320

scan

*getline() function, 320*

*menus, 458-459*

setting for Medium Code entry and edit screen, 411-413

standards for navigating entry and edit windows, 387-388

**keys help, 518**

**keywords**

ANSI C, 139-140

far, 25

reserved in C (listing), 772-776

see also *constants, modifiers, operators*

typedef, 34

## L

**languages**

assembler, 740-741

C++, 742-744

object-oriented programming, 742-744

procedural, 742

**large memory model, 758**

**Last In First Out (LIFO) order of access, 83-87**

**LDBL\_DIG constant, 147**

**LDBL\_MAX constant, 148**

**LDBL\_MAX\_10\_DBL constant, 148**

**LDBL\_MAX\_EXP constant, 148**

**LDBL\_MIN constant, 148**

**LDBL\_MIN\_10\_EXP constant, 148**

**LDBL\_MIN\_EXP constant, 148**

**leaf nodes, trees, 89**

**left arrow key**



## left arrow key

- getline() function, 320
- menu scan codes, 459
- standards for navigating entry and edit windows, 387
- .LIB file extension, 202, 217**
- LIB library program, 208**
- libraries, 207-209**
  - commercial, 744-745
  - communications, 747
  - creating, 213-218
  - file management, 747-748
  - functions
    - adding, 217-218*
    - deleting, 220*
    - extracting/moving, 221*
    - listing, 218-220*
    - organizing, 266-269*
    - updating, 221*
  - general, 748-749
  - graphics, 746
  - sources, 208, 221
  - TYAC.LIB file, updating, 289-291
- LIFO (Last In First Out) order of access, 83-87**
- limits.h header file, 782**
- \_\_LINE\_\_ predefined constant, 12, 160**
- linear (single) linked lists, 69-73**
- lines, drawing, 280-281**
- linked index files, 537**
- linked lists, 68**
  - adding elements
    - first link, 73-75*
    - to beginning, 75-76*
    - to end, 77*
    - to middle, 76*
  - binary trees, 89-99
  - deleting elements, 82
  - double, 88-89
  - queues, 87-89
  - single (linear), 69-73
  - stacks, 83-87
  - using, 77-82
- LIST data type, 72**
- list\_groups() function, 685-690**
- list\_medium\_codes() function, 678-685**
- LIST0101.C source file, 5-6**
- LIST0102.C source file, 7-8**
- LIST0103.C source file, 10-11**
- LIST0104.C source file, 12-14**
- LIST0105.C source file, 16-17**
- LIST0106.C source file, 18-20**
- LIST0107.C source file, 23-25**
- LIST0205.C source file, 48**
- LIST0207.C source file, 53-54**
- LIST0208.C source file, 55-57**
- LIST0209.C source file, 61-63**
- LIST0301.C source file, 70-72**
- LIST0302.C source file, 77-81**
- LIST0304.C source file, 91-98**
- LIST0403.C source file, 119-120**
- LIST0405.C source file, 123-124**
- LIST0406.C source file, 126-127**
- LIST0501.C source file, 135-136**
- LIST0502.C source file, 136-137**
- LIST0503.C source file, 140-141**
- LIST0504.C source file, 144**
- LIST0505.C source file, 148-149**
- LIST0506.C source file, 150-151**
- LIST0507.C source file, 152-153**
- LIST0508.C source file, 154-156**
- LIST0509.C source file, 158**
- LIST0510.C source file, 161**
- LIST0511.C source file, 162-163**
- LIST0601.C source file, 171**
- LIST0602.C source file, 177-178**
- LIST0701.H header file, 204-205**
- LIST0702.H header file, 205**
- LIST0703.C source file, 205-207**
- LIST0709.C source file, 214**
- LIST0711.C source file, 222-223**
- LIST0802.C source file, 236-237**
- LIST0805.C source file, 239-240**
- LIST0811.C source file, 254-255**
- LIST0907.C source file, 278**
- LIST0909.C source file, 280-281**
- LIST0914.H header file, 289-291**
- LIST1103.C source file, 334-335**
- LIST1110.C source file, 347-348**
- LIST1404.C source file, 464-470**
- LIST1405.C source file, 472-474**
- LIST1602.C source file, 516-517**
- LIST1604.C source file, 522-524**
- LIST1605.C source file, 525**
- LIST2002.C source file, 723-724**
- LIST2003.C source file, 725-726**
- listings**
  - 1.1. C variable sizes, 5
  - 1.2. A program showing wasted memory, 7-8
  - 1.3. Using the malloc() function, 10-11
  - 1.4. Allocating memory without freeing it, 12-14



- 1.5. The use of `realloc()`, 16-17
- 1.6. Using `calloc()`, 18-19
- 1.7. Use of far memory allocation functions, 23-24
- 2.1. Accessing arrays and their data, 35-36
- 2.2. Use of the date structure, 38-39
- 2.3. Unions in memory, 42-43
- 2.4. A pointer review, 46
- 2.5. Using a function pointer, 48
- 2.6. Passing pointers to functions, 49-51
- 2.7. Using a pointer to a structure, 53-54
- 2.8. Using an array of structures, 55-57
- 2.9. Program using variable-length structure, 61-62
- 3.1. A first look at a linked list, 70-71
- 3.2. Adding to a linked list of characters, 77-80
- 3.3. `STACK.C`. Using a stack, 83-86
- 3.4. Using a binary tree, 91-97
- 4.1. `CPUT.C`. Place the cursor on the screen, 113
- 4.2. `CUP.C`. Move the cursor up, 113
- 4.3. `CDOWN.C`. Move the cursor down, 114
- 4.4. `CRIGHT.C`. Move the cursor right, 114
- 4.5. `CLEFT.C`. Move the cursor left, 114
- 4.6. `SAVECURS.C`. Save the current cursor position, 115
- 4.7. `RSTRCURS.C`. Restore the saved cursor position, 115
- 4.8. `CLRSCRN.C`. Clear the screen, 116
- 4.9. `CLEAR_EOL.C`. Clear to the end of the line, 116
- 4.10. `A_CURSOR.C`. The ANSI cursor functions, 117-118
- 4.11. `A_CURSOR.H`. The ANSI cursor functions header file, 118
- 4.12. `LIST0403.C`. Using the ANSI cursor functions, 119-120
- 4.13. `ANSICLRS.H`. The ANSI colors, 122
- 4.14. `LIST0405.C`. Using the ANSI colors, 123-124
- 4.15. Illustrates re-mapping of keys using the ANSI escape sequences, 126-127
- 5.1. Code with spacing, 135-136
- 5.2. Compact code, 136-137
- 5.3. Case sensitivity, 140
- 5.4. The size of the data types, 144
- 5.5. The values stored in the ANSI-defined constants, 148-149
- 5.6. Using the ANSI-defined constants, 150
- 5.7. Is the character an alphabetic letter?, 152
- 5.8. Using character classification functions, 154-155
- 5.9. A program that may not be portable, 158
- 5.10. The predefined ANSI constants in action, 161
- 5.11. A portable program with compiler specifics, 162-163
- 6.1. ASCII values, 171
- 6.2. A character translation program, 177-178
- 6.3. Program to type a file in decimal, 182
- 6.4. Program to type a file in binary, 184-186
- 6.5. Program to type a file in hexadecimal, 192-193
- 6.6. Program to type a file in octal, 194-195
- 6.7. Program to type a file in hexadecimal with the character representations, 196-197
- 7.1. Using include files, 204-205
- 7.2. A second header file, 205
- 7.3. The source file, 205
- 7.4. Partially precompiled version of Listing 7.3, 206-207
- 7.5. A state edit, 209-210
- 7.6. A sex code edit, 210-211
- 7.7. A date edit, 211-212
- 7.8. A birthdate edit, 212-213
- 7.9. Using the edit functions, 214
- 7.10. A header file for the edit functions, 215
- 7.11. Using the `EDITS.LIB`—again!, 222-223
- 8.1. `BIOSREGS.H`, 235
- 8.2. `LIST0802`. Using BIOS interrupt call, 236





- 8.3. PCURSOR.C. Placing the cursor at screen coordinates, 238
- 8.4. GCURSOR.C. Getting the coordinates of the cursor, 238-239
- 8.5. LIST0805.C. Using the BIOS cursor functions, 239-240
- 8.6. set\_cursor\_size(), manipulating the cursor's shape, 241-242
- 8.7. BIGCURS.C. Using the set\_cursor\_size() function, 242
- 8.8. SMLCURS.C. Using the set\_cursor\_size() function, 242
- 8.9. keyhit(). A function to determine if a keyboard character has been pressed, 253-254
- 8.10. kbwait(). A function to clear the keyboard buffer, 254
- 8.11. LIST0811.C. A program demonstrating the previous two functions, 254-255
- 9.1. The enhanced TYAC.H header file, 266-268
- 9.2. PCURSOR.C. A function to put the cursor on screen, 269-270
- 9.3. GCURSOR.C. A function to get cursor from screen, 270-271
- 9.4. GVIDEO.C gets the video mode, 273
- 9.5. SVIDEO.C sets the video mode, 274-275
- 9.6. SBRDCLR.C sets the border color, 275-276
- 9.7. WRITECH.C. A function to write a character in color, 276-277
- 9.8. Using a for loop to repeat a character, 278
- 9.9. REPEATCH.C. A better repeating character function, 279
- 9.10. Using the repeat\_char() function and drawing lines, 280-281
- 9.11. WRITESTR.C writing a string in color, 282
- 9.12. BOX.C. The box() function, 284-285
- 9.13. DRAWBOX.C. Using the box() function to draw boxes, 286
- 9.14. PAUSE.C. A pausing function, 288-289
- 9.15. LIST0914.H. A new version of TYAC.H, 289-291
- 10.1. A new TYAC.H header file, 298-302
- 10.2. GETLINE.C. The getline() function, 306-316
- 10.3. BOOP.C. The boop() function, 321-322
- 10.4. WAITSEC.C. The waitsec() function, 323
- 10.5. GETTICKS.C. The get\_timer\_ticks() function, 324
- 10.6. GL\_TEST.C using the getline() function, 325-326
- 11.1. CURSOFF.C. The cursor\_off() function, 332-333
- 11.2. CURSON.C. The cursor\_on() function, 333-334
- 11.3. LIST1103.C. Using the cursor functions, 334-335
- 11.4. CLEARSCN.C. The clear\_screen() function, 336
- 11.5. TESTCLR.C. Test the clear\_screen() function, 337
- 11.6. GRID.C. The grid() function, 339
- 11.7. TESTGRID.C. Testing the grid() function, 340
- 11.8. SAVESCRN.C. Saving a portion of the screen, 341-343
- 11.9. RESSCRN.C. Restoring the saved portion of the screen, 345-346
- 11.10. LIST1110.C. Demonstration of saving and restoring the screen, 347-348
- 13.1. RECOFREC.C. A temporary *Record of Records!* menu, 391-396
- 13.2. RECOFREC.H. The *Record of Records!* program header, 396-397
- 13.3. RECORDS.H. *Record of Records!* program header containing the structures for the record layouts, 397-399
- 13.4. MEDIUMS.C. The medium screen, 403-409
- 13.5. The Group Information screen, 414-422
- 13.6. ALBUMS.C. The Musical Items entry and edit screen, 425-437
- 14.1. MENU.C. The display\_menu() routine, 445-452
- 14.2. COLORTBL.H. The color table, 452-453

- 14.3. TESTMENU.C. A test program for display\_menu(), 460-462
- 14.4. LIST1404.C. A new *Record of Records*/listing, 464-470
- 14.5. The Type of Music entry and edit menu, 472-474
- 15.1. MMNUABAR.C. The main menu's action bar functions, 488-490
- 15.2. RECOFREC.H with the action bar function prototypes, 493-494
- 15.3. MEDMABAR.C. The action bar for the Medium Code screen, 495-503
- 16.1. PRINTIT.C. Listing using startup help, 512-513
- 16.2. LIST1602.C. A new display\_medium\_help() function, 516-517
- 16.3. ABOUT.C. The *Record of Records*/about box code, 519-520
- 16.4. LIST1604.C. The context-sensitive help, 522-524
- 16.5. LIST1605.C. Replacement F1 case and new F2 case, 525
- 16.6. MEDIUM.HLP. The context-sensitive help for the medium code screen, 525
- 17.1. OPENFILE.C. Opens an index and a data file, 540-542
- 17.2. GETREC.C. Getting a record—the get\_rec() function, 543-544
- 17.3. PUTREC.C. Writing a record—the put\_rec() function, 545-546
- 17.4. UPDHDR.C. The update\_header() function, 546-547
- 17.5. CLOSFIL.C. Closing a file—the close\_files function, 548-549
- 17.6. MEDIUM.C. The medium screen's main file, 551-563
- 17.7. ADDMREC.C. Adding a medium code, 566-570
- 17.8. DELMREC.C. Deleting a medium code, 573-577
- 17.9. PROCMREC.C. Processing the next and previous records, 579-582
- 18.1. MEDMABAR.C. The medium action bar function, 588-599
- 18.2. ALBUMS.C. The Musical Items screen's main file, 605-623
- 18.3. SONGS.C. Functions needed for the Musical Items screen, 628-634

- 18.4. ADDAREC.C. Adding a medium code, 636-640
- 18.5. DELAREC.C. Deleting a musical item, 641-645
- 18.6. PROCAREC.C. Processing the next and previous records, 646-650
- 18.7. ALBMABAR.C. The Musical Items screen action bar, 650-661
- 19.1. REC\_RPTG.C. The reporting menu code, 673-677
- 19.2. LISTMED.C. The list of medium codes, 678-681
- 19.3. The convert\_str() function, 684-685
- 19.4. RPT\_GRP.C. The List of Groups, 685-689
- 19.5. RPT\_DETL.C. The detailed Musical Item Report, 691-702
- 19.6. REC\_RPTG.C with the get\_alb\_selection() function, 703-708
- 20.1. SYNTAX.C. A program with syntax errors, 719-720
- 20.2. LIST2002.C. A simple listing, 723-724
- 20.3. LIST2003.C. The simple listing updated, 725
- 20.4. ASSERT.C. Using the assert() macro, 727-728
- 20.5. PERROR.C. Using the perror() function, 729-730
- 21.1. Some assembler code, 741

**LISTLINK data type, 72**

**LISTMED.C source file, 678-682 lists, 667**

- Group Information
  - creating, 685-690
  - report prototype, 668-669
- Medium Code
  - creating, 678-685
  - report prototype, 667-668

**LOCALE.H header file, 782**

**logic errors, 721**

**logical design of computer programs/systems, 358-361**

**long data type, 32, 34**

**long int data type, maximum/minimum values, 33**

**long keyword, 774**

**long variables, 144**

**LONG\_MAX constant, 147**

**LONG\_MIN constant, 147**



## LPSTR data type

**lowercase()** function, 157  
**LPCSTR** data type, 34  
**LPSTR** data type, 34  
**.LST** file extension, 219

## M

**macros**, **assert()**, 726-729  
**MagnalComm C** (SoftDesign International, Inc.), 747  
**main** screen, returning control to, 506  
**main()** function  
    in **recofrec.c** source file, 401  
    **TESTMENU.C** source file, 463  
**main-menu** action bars, 488-492  
**maintainability**, programming for, 134-138  
**malloc()** function, 10-12, 74  
**manuals**, 528-529  
**math.h** header file, 783  
**matrices**, **Record of Records!** application, 377-378  
**Medium Code**  
    entry and edit screen, 403-413  
        *access requirements*, 377  
        *adding functions to action bars*, 588-601  
        *adding records*, 566-573  
        *deleting records*, 578-579  
        *extended help*, 516-518  
        *keys*, 376  
        *multi-menu action bars*, 492-506  
        *processing next/previous records*, 579-583  
        *prototype screen*, 376  
        *updating files for*, 550-566  
    list  
        *creating*, 678-685  
        *prototype*, 667-668  
    matrix, 378  
**medium memory model**, 757  
**MEDIUM.C** source file, 551-563  
**MEDIUM.HLP** help file, 525  
**MEDIUM\_REC** structure, 400  
**MEDIUMS.C** source file, 403-409  
**MEDMABAR.C** source file, 495-503, 588-599  
**member (.)** operator, 55  
**memcpy()** function, 627  
**memory**  
    accessing directly, 231-232

allocating  
    *at compile time*, 6-9  
    *dynamically*, 9-12  
    *far (over 64K)*, 21-26  
    *initializing and*, 17-21  
deallocating (freeing), 12-15  
reallocating (resizing) blocks, 15-17  
segmented architecture, 754  
variable requirements, 4-6  
video, accessing  
    *ANSI functions*, 105-128  
    *direct*, 128  
    *through BIOS*, 128-129

### memory models

64K limit, 760  
compact, 757-758  
comparing, 759-760  
components, 754-755  
huge, 758-759  
large, 758  
medium, 757  
small, 756  
tiny, 755-756

### memset() function, 628

### MENU.C source file, 445-452

### menus, 444-445

action bars, 482  
    *multi-menu*, 492-506  
    *single (main) menu*, 488-492  
    *standards for menu items*, 483-487  
adding to **Record of Records!** application, 464-471  
color table for, 452-454  
displaying, 445-458, 460-464  
entry and edit screens, temporary system, 390-402  
non-scan keys, 459-460  
Reporting, creating, 672-678  
scan codes, 458-459  
Type of Music (**Record of Records!** application), 472-475

### message boxes, displaying, 401

### messages

displaying for system errors, 729-731  
printing when predefined conditions occur, 726-729  
*see also* error messages

### methodologies, 354-355

for reports, 666  
program specifications, 365-366  
prototyping, 364-365

rapid or accelerated systems analysis  
and design, 364  
Structured Systems Analysis and  
Design, 355-356  
    *Construction*, 361-362  
    *Preliminary Analysis*, 356-358  
    *Systems Analysis (logical design)*,  
    358-361  
    *Systems Design (physical design)*, 361  
    *Wrap up*, 362-363  
**Microsoft**  
    compiler-specific functions  
        *BIOS*, 256  
        *far memory*, 22  
    LIB library program, 208  
**minus sign (-) subtraction operator**, 220  
**MMNUABAR.C source file**, 488-490  
**mnemonic keys**, 444-445  
**models**  
    64K limit, 760  
    compact, 757-758  
    comparing, 759-760  
    components, 754-755  
    huge, 758-759  
    large, 758  
    medium, 757  
    small, 756  
    tiny, 755-756  
**modes**  
    binary, 534  
    text, 534  
    video  
        *controls*, ANSI codes, 108  
        *getting/setting*, 273-275  
**modifiers, static**, 491, 775  
**monitors, see video display**  
**monochrome text**, 264  
**move\_cursor\_down() function**, 114  
**move\_cursor\_left() function**, 114  
**move\_cursor\_right() function**, 114  
**move\_cursor\_up() function**, 113  
**moving**  
    cursor, 269-272  
        *BIOS functions*, 237-241  
        *escape sequences for*, 113-115, 117-  
        121  
    functions from libraries, 221  
**\_MSC\_VER constant**, 164  
**multi-menu action bars**, 492-506  
**Multiple declaration for 'var1' in**

**function main error message**, 141  
**multiple source files**, 203-204  
**music\_rpt() function**, 702  
**Musical Information report, prototype**,  
669-672  
**Musical Items entry and edit screen**,  
425-438, 604-625  
    access requirements, 376-377  
    action bar, 374, 650-661  
    adding/changing and calculating songs,  
    628-635  
    capturing data, 625-626  
    keys/function keys, 372-373  
    page up and page down functions,  
    626-628  
    prototype screen, 371  
    records  
        *adding*, 635-640  
        *deleting*, 641-646  
        *processing next/previous*, 646-650  
    updating files, 602-604

## N

**network programming**, 739-740  
**nodes, trees**, 89  
**non-DOS platforms, programming  
for**, 744  
**non-interactive tutorials**, 527  
**non-scan keys, menus**, 459-460  
**Novell, Inc.'s Btrieve**, 748  
**null characters**, 38  
**NULL pointers in linked lists**, 69-70  
**NULL string**, 17  
**Null terminator**, 7  
**number systems**  
    ASCII Character Table, displaying,  
    170-176  
    binary, 183-189  
    decimal, 179-183  
    hexadecimal, 190-194  
    octal, 194-196  
    translating characters into multiple,  
    176-179  
**numbers**  
    function, 233  
    integers  
        *allocating memory for*, 18-20  
        *converting to strings*, 684-685  
    zeros, padding fields with, 401-402  
**numeric variables**



## **.OBJ file extension**

---

determining type sizes, 143-145  
setting maximum/minimum values,  
145-151

## **O**

---

**.OBJ file extension, 202, 217**  
**object-oriented programming lan-  
guages, 742-744**  
**objective list testing, 715**  
**objectives of programs**  
    overview statements, 356  
    setting, 356-357  
**octal number system, 194-196**  
    translating characters into, 176-179  
**OCTDUMP.C source file, 194-195**  
**online tutorials, 527**  
**open\_files() function, 540-543**  
**open\_songs() function, 625**  
**OPENFILE.C source file, 540-542**  
**opening files, 540-543**  
**operators**  
    & (address of), 45  
    \* (dereference), 37  
    \* (indirection), 55  
    + (addition), 217  
    - (subtraction), 220  
    -> (indirect membership), 55  
    . (member), 55  
    order of precedence, 778-779  
    sizeof, 6, 21, 775  
**option parameter, getline() function,  
303-304**  
**Options menu, standards, 486**  
**order of precedence, operators, 778-779**

## **P**

---

**page breaks, 690**  
**page down key**  
    Medium Code entry and edit  
        screen, 412  
    Musical Items entry and edit screen,  
        372, 626-628  
    standards for navigating entry and edit  
        windows, 387  
**page up key**  
    Medium Code entry and edit  
        screen, 412  
    Musical Items entry and edit screen,  
        372, 626-628

**pages, video, 274**

### **parameters**

    command line, startup help for,  
        511-514  
    display\_menu() function, 454-456  
    getline() function, 304-305  
        *option, 303-304*

**passing by reference, 52**

**passing by value, 52**

**patches, 732-733**

**pause() function, 288-289**

**PAUSE.C source file, 288-289**

### **pausing**

    computers, 322-323  
    for user input, 288-289

**PCURSOR.C source file, 238, 269-270**

**PCX ToolKit (Genus Micro program-  
ming), 746**

**period (.) member operator, 55**

**perror() function, 729-731**

**PERROR.C source file, 729-736**

**physical design of computer**

    programs/systems, 361

**pixel graphics, 265**

**planning methodologies, *see*  
    methodologies**

**platforms (non-DOS), programming  
for, 744**

**plus sign (+) addition operator, 217**

**POINTER.C source file, 46-47**

### **pointers**

    declaring, 44-45  
    finding variable addresses, 45-47  
    head, 69-70  
    linked lists, 69  
    tail, 88-89  
    to functions, 48-52  
    to other pointers, 47-48  
    to structures, 52-55

**polymorphism, 743**

**pop\_stack() function, 87**

**portability, 138-139**

    ANSI standards, 139

    case

*converting, 156-157*

*sensitivity, 140-141*

    character sets, 142

    classifying variables, 151-156

    enforcing ANSI compatibility, 142-143

    keywords, 139-140

    non-ANSI features, 162-164

- numeric variables, 143-145
  - setting maximum/minimum values, 145-151*
- predefined constants, 159-162
- preprocessor directives, 159
- structures and unions, 157-159
- post-order access, 90-99**
- pre-order access, 90-99**
- precedence, operators, 778-779**
- predefined constants, 159-162**
- Preliminary Analysis, 356-358**
  - reviewing, 359
- preliminary estimates, 358**
- preprocessor directives, 159**
  - #include, 204-207
- print statements, debugging with, 723-726**
- print\_group() function, 690**
- print\_grp\_header() function, 690**
- print\_med\_hdr() function, 683**
- printf() function, 178-179**
- printing messages when predefined conditions occur, 726-729**
- PRINTIT.C source file, 512-513**
- proc\_med\_rec() function, 579-583**
- PROCAREC.C source file, 646-650**
- procedural languages, 742**
- process\_med\_list() function, 683-684**
- PROCMREC.C source file, 579-582**
- programming**
  - for efficiency, 134-138
  - for maintainability, 134-138
  - for non-DOS platforms, 744
  - for portability, 138-164
  - in assembler language, 740-741
  - in C++ language, 742-744
  - specialized, 738-740
- programs**
  - Bounds Checker, 732
  - commercial, 744-749
  - debuggers, 731-732
  - design methodologies
    - program specifications, 365-366*
    - prototyping, 364-365*
    - rapid or accelerated systems analysis and design, 364*
    - Structured Systems Analysis and Design, 355-363*
  - LIB library, 208
  - patches/updates/upgrades, 732-733
  - Record of Records!, *see* Record of

- Records! application
- TLIB library, 209
- versus computer systems, 354
- see also* source files
- prompts, drawing**
  - field, 410
  - group, 424
- prototypes**
  - creating
    - program specifications, 365-366*
    - rapid or accelerated systems analysis and design, 364*
    - Structured Systems Analysis and Design, 361*
  - functions, consolidating in header files, 400
  - reports, 666
    - Group Information list, 668-669*
    - Medium Code list, 667-668*
    - Musical Information, 669-672*
- prototyping, 355**
  - as methodology, 364-365
- push\_stack() function, 87**
- put\_cursor() function, 113**
- put\_rec() function, 545-546**
- PUTREC.C source file, 545-546**

## Q

- qsort() function, 49-52**
- queues, 87-89**

## R

- rapid systems analysis and design, 364**
- reading records from files, 543-545**
- realloc() function, 15-17, 21**
- REC\_RPTG.C source file, 673-677, 703-708**
- RECOFREC.C source file, 391-396, 550, 602, 678**
- RECOFREC.H header file, 396-397, 493-494, 550, 602-603, 677-678**
- Record of Records! application**
  - about boxes, 519-520
  - database access requirements, 376-379
  - file structures, 538-539
  - Group Information entry and edit screen, 414-425
  - keys, 375*
  - prototype screen, 375*





## Record of Records! application

Medium Code entry and edit screen,  
403-413

*adding functions to action bars,*  
588-601

*adding records, 566-573*

*deleting records, 578-579*

*extended help, 516-518*

*keys, 376*

*processing next/previous records,*  
579-583

*prototype screen, 376*

*updating files for, 550-566*

menus

*adding, 464-471*

*multi-menu action bars, 492-506*

*Reporting, creating, 672-678*

*single (main) menu action bar,*  
488-492

*temporary system, 390-402*

*Type of Music, 472-475*

Musical Items entry and edit screen,  
425-438, 604-625

*action bar, 374, 650-661*

*adding records, 635-640*

*adding/changing and calculating*  
*songs, 628-635*

*capturing data, 625-626*

*deleting records, 641-646*

*keys/function keys, 372-373*

*page up and page down functions,*  
626-628

*processing next/previous records,*  
646-650

*prototype screen, 371*

*updating files, 602-604*

objectives, 370

purpose, 370

rules, 371

**records, 533**

adding

*to Medium Code entry and edit*  
*screen, 566-573*

*to Musical Items entry and edit screen,*  
635-640

deleting

*from Medium Code entry and edit*  
*screen, 578-579*

*from Musical Items entry and edit*  
*screen, 641-646*

processing next/previous, 579-583,

646-650

reading from files, 543-545

updating header information, 546-548

writing to files, 545-546

**RECORDS.H header file, 397-399, 551,**  
**603-604**

**recursive errors, 722**

**reference guides, 528**

**register keyword, 774**

**registers, 755**

**REGS union, 234-236**

**repeat\_char() function, 279-281**

**REPEATCH.C source file, 279**

**repeating characters, 278-279**

**Reporting menu, creating, 672-678**

**reports**

customizing, 709-710

design methodologies, 666

Detailed Information, creating,  
691-709

Group Information list

*creating, 685-690*

*prototype, 668-669*

Medium Code list

*creating, 678-685*

*prototype, 667-668*

Musical Information, prototype,  
669-672

prototypes, 666

**reset\_memory() function, 635**

**/\*resident\_C\*/ (South Mountain**  
**Software, Inc.), 749**

**resizing memory blocks, 15-17**

**resources, 104-129, 230**

**RESSCRN.C source file, 345-346**

**restore\_cursor\_position() function, 115**

**restore\_screen\_area() function, 345-347**

**restoring**

saved cursor positions, 115

screens, 341-349

**retrieving data, 411-413, 424-425**

Musical Items entry and edit screen,  
625-626

**return keyword, 774**

**reviewing Preliminary Analysis, 359**

**rewrite\_menu\_items() function,**  
**458-459**

**right arrow key**

getline() function, 320

menu scan codes, 459

standards for navigating entry and edit



windows, 387  
 ROM BIOS, software interrupts  
   (listing), 244-252  
 roots, trees, 89  
 RPT\_DETL.C source file, 691-702  
 RPT\_GRP.S.C source file, 685-689  
 RSTRCURS.C source file, 115

## S

save\_cursor\_position() function, 115  
 save\_screen\_area() function, 341-347  
 SAVECURS.C source file, 115  
 SAVESCRN.C source file, 341-343  
 saving  
   current cursor positions, 115  
   screens and restoring, 341-349  
 SBRDCLR.C source file, 275-276  
 scan codes, 318  
   menus, 458-459  
 scan key, getline() function, 320  
 SCHAR\_MAX constant, 146  
 SCHAR\_MIN constant, 146  
 scheduling work on computer  
   programs/systems, 358  
 Scientific Endeavors Corporation's  
   GraphiC, 746  
 scope of computer  
   programs/systems, 357  
 screens  
   clearing, 335-338  
   colors  
     borders, 275-276  
     escape sequences for, 121-126  
     text, 276-278  
   entry and edit, *see* entry and edit  
   screens  
   erasing, escape sequences for, 115-121  
   main, returning control to, 506  
   saving and restoring, 341-349  
   setting up, 410-411  
 script testing, 715  
 SCURS.C source file, 241-242  
 search\_list() function, 99  
 search\_med\_rec() function, 600-601  
 segmented memory architecture, 754  
 Sequiter Software, Inc.'s CodeBase, 748  
 set-up manuals, 528  
 set\_border\_color() function, 275-276  
 set\_cursor\_size() function, 241-243  
 set\_video() function, 274-275  
 setjmp.h header file, 783  
 setup\_exit\_keys() function, 321  
 setup\_today() function, 683  
 SEX.C source file, 210-211  
 shadows, creating, 125  
 shift+tab keys  
   Medium Code entry and edit  
     screen, 412  
   Musical Items entry and edit  
     screen, 372  
 short data type, 32-34  
 short int data type, maximum/minimum  
   values, 33  
 short keyword, 774  
 short variables, 143  
 show\_list() function, 82  
 SHRT\_MAX constant, 146  
 SHRT\_MIN constant, 146  
 signal.h header file, 783  
 signed char data type, maximum/  
   minimum values, 33  
 signed int data type, maximum/  
   minimum values, 33  
 signed integer variables, 144  
 signed keyword, 775  
 signed long int data type, maximum/  
   minimum values, 33  
 signed short int data type, maximum/  
   minimum values, 33  
 SilverWare, Inc.'s "C" EMM  
   Library, 748  
 single-linked lists, 69-73  
 sizeof operator, 6, 21, 775  
 sizes  
   cursor, BIOS functions, 241-243  
   numeric variable types, determining,  
     143-145  
   scope of computer programs/systems,  
     357  
   source files, 203-204  
   variables, 4-6  
 small memory model, 756  
 SMLCURS.C source file, 242  
 SoftDesign International, Inc.'s  
   MagnaComm C, 747  
 software interrupts, 129, 232  
   creating, 255-256  
   listing of, 244-252  
 SONG\_REC structure, 400  
 songs  
   adding, 635-640



## **SORT.C source file**

---

adding/changing and calculating, 628-635  
deleting, 641-646  
**SONGS.C source file, 628-634**  
**SORT.C source file, 49-51**  
**sorted index files, 537**  
**sorting**  
elements, with binary trees, 90-99  
in ascending/descending orders, 49-52  
**sounds, beeping, 321-322**  
**source files, 202-203**  
A\_CURSOR.C, 117-118  
ABOUT.C, 519-520  
ADDAREC.C, 636-640  
ADDMREC.C, 566-570  
ALBMABAR.C, 650-661  
ALBUMS.C, 425-437, 605-623  
ARRAY.C, 35-36  
ASSERT.C, 727-728  
BDATE.C, 212-213  
BIGCURS.C, 242  
BINDUMP.C, 184-186  
BOOP.C, 321-322  
BOX.C, 284-285  
CDOWN.C, 114  
CLEARCOL.C, 116  
CLEARSCN.C, 336  
CLEFT.C, 114  
CLOFILE.C, 548-549  
CLRSCRN.C, 116  
CONVSTR.C, 684-685  
CPUT.C, 113  
CRIGHT.C, 114  
CUP.C, 113  
CURSOFF.C, 332-333  
CURSON.C, 333-334  
DATE.C, 211-212  
DECDUMP.C, 182-183  
DELAREC.C, 641-645  
DELMREC.C, 573-577  
DRAWBOX.C, 286  
G\_CURSOR.C, 238-239, 270-271  
GETLINE.C, 306-316  
GETREC.C, 543-544  
GETTICKS.C, 324  
GL\_TEST.C, 325-326  
GRID.C, 339  
GROUPS.C, 414-422  
GVIDEO.C, 273  
header files, including, 204-207  
HEX.C, 196-198  
HEXDUMP.C, 192-193  
KBWAIT.C, 254  
KEYHIT.C, 253-254  
LIST0101.C, 5-6  
LIST0102.C, 7-8  
LIST0103.C, 10-11  
LIST0104.C, 12-14  
LIST0105.C, 16-17  
LIST0106.C, 18-20  
LIST0107.C, 23-25  
LIST0205.C, 48  
LIST0207.C, 53-54  
LIST0208.C, 55-57  
LIST0209.C, 61-63  
LIST0301.C, 70-72  
LIST0302.C, 77-81  
LIST0304.C, 91-98  
LIST0403.C, 119-120  
LIST0405.C, 123-124  
LIST0406.C, 126-127  
LIST0501.C, 135-136  
LIST0502.C, 136-137  
LIST0503.C, 140-141  
LIST0504.C, 144  
LIST0505.C, 148-149  
LIST0506.C, 150-151  
LIST0507.C, 152-153  
LIST0508.C, 154-156  
LIST0509.C, 158  
LIST0510.C, 161  
LIST0511.C, 162-163  
LIST0601.C, 171  
LIST0602.C, 177-178  
LIST0703.C, 205-207  
LIST0709.C, 214  
LIST0711.C, 222-223  
LIST0802.C, 236-237  
LIST0805.C, 239-240  
LIST0811.C, 254-255  
LIST0907.C, 278  
LIST0909.C, 280-281  
LIST1103.C, 334-335  
LIST1110.C, 347-348  
LIST1404.C, 464-470  
LIST1405.C, 472-474  
LIST1602.C, 516-517  
LIST1604.C, 522-524  
LIST1605.C, 525  
LIST2002.C, 723-724  
LIST2003.C, 725-726  
LISTMED.C, 678-682

MEDIUM.C, 551-563  
 MEDIUMS.C, 403-409  
 MEDMABAR.C, 495-503, 588-599  
 MENU.C, 445-452  
 MMNUABAR.C, 488-490  
 multiple, 203-204  
 OCTDUMP.C, 194-195  
 OPENFILE.C, 540-542  
 PAUSE.C, 288-289  
 PCURSOR.C, 238, 269-270  
 PERROR.C, 729-736  
 POINTER.C, 46-47  
 PRINTIT.C, 512-513  
 PROCAREC.C, 646-650  
 PROCMREC.C, 579-582  
 PUTREC.C, 545-546  
 REC\_RPTG.C, 673-677, 703-708  
 RECOFREC.C, 391-396, 550,  
     602, 678  
 REPEATC.C, 279  
 RESSCRN.C, 345-346  
 RPT\_DETL.C, 691-702  
 RPT\_GRP.C, 685-689  
 RSTRCURS.C, 115  
 SAVECURS.C, 115  
 SAVESCRN.C, 341-343  
 SBRDCLR.C, 275-276  
 SCURSOR.C, 241-242  
 SEX.C, 210-211  
 SMLCURS.C, 242  
 SONGS.C, 628-634  
 SORT.C, 49-51  
 STACK.C, 83-86  
 STATE.C, 209-210  
 STRUCT.C, 38-39  
 SVIDEO.C, 274-275  
 SYNTAX.C, 719-720  
 TESTCLR.C, 337  
 TESTGRID.C, 340  
 TESTMENU.C, 460-462  
 UNION.C, 42-44  
 UPDHDR.C, 546-547  
 WAITSEC.C, 323  
 WRITECH.C, 276-277  
 WRITESTR.C, 282

#### **South Mountain Software, Inc.**

C Utility Library, 748-749  
 Essential B-Tree, 748  
 Essential Communications, 747  
 Hold Everything, 749  
 /\*resident\_C\*/, 749

**specialized programming, 738-740**  
**specifications, programs, 365-366**  
**sprintf() function, 338**

**STACK.C source file, 83-86**

**stacks, 83-87**

**standards, 385**

    action-bar menu items, 483-487

    ANSI, 139

*case sensitivity, 140-141*

*character sets, 142*

*classifying variables, 151-156*

*converting case, 156-157*

*enforcing compatibility, 142-143*

*keywords, 139-140*

*numeric variables, 143-151*

*predefined constants, 159-162*

*preprocessor directives, 159*

*structures and unions, 157-159*

    benefits of following, 385

    entry and edit, 386-388

    function keys, 388-390

**startup help, 511-514**

**STATE.C source file, 209-210**

**static modifier, 491, 775**

**STDARG.H header file, 783**

    \_\_STDC\_\_ predefined constant,  
     142, 160

**STDDEF.H header file, 783**

**STDIO.H header file, 783**

**STDLIB.H header file, 783**

**STRING.H header file, 783**

**strings**

    converting integers to, 684-685

    entering, 325-327

    in color, 282-283

    NULL, 17

    trash, 12

    versus character arrays, 37-38

**struct keyword, 775**

**STRUCT.C source file, 38-39**

**Structured Systems Analysis and  
 Design, 355-356**

    Construction, 361-362

    Preliminary Analysis, 356-358

    Systems Analysis (logical design),  
     358-361

    Systems Design (physical design), 361

    Wrap up, 362-363

**structures, 38-41**

    ALBUM\_INDEX, 604

    ALBUM\_REC, 400



## structures

---

- arrays of, 55-58
- BYTEREGS, 233-235
- color\_table, 400-401, 452-454
- DATE, 400
- fields, clearing, 410
- GROUP\_REC, 400
- HREG, 235-236
- MEDIUM\_REC, 400
- pointers to, 52-55
- portability, 157-159
- SONG\_REC, 400
- unions, 41-44
- variable-length, 58-63
- WORDREGS, 233-235
- XREG, 235-236

**subtraction (-) operator, 220**

**SVIDEO.C source file, 274-275**

**switch keyword, 775**

**switch\_globals() function, 702**

**syntax errors, 719-721**

**SYNTAX.C source file, 719-720**

**system**

- clock, tick counter value, returning, 323-324
- errors, displaying messages about, 729-731
- resources, 104, 230
  - video display, 104-129*

**Systems Analysis (logical design), 358-361**

**Systems Design (physical design), 361**

## T

---

**tab key**

- Medium Code entry and edit screen, 411
- Musical Items entry and edit screen, 372
- standards for navigating entry and edit windows, 387

**tail pointers, 88-89**

**teams for projects, creating, 359**

**TeraTech**

- E-MEM, 748
- Virt-Win, 746

**TEST file, 181**

**TESTCLR.C source file, 337**

**TESTGRID.C source file, 340**

**testing**

- alpha, 718
- beta, 718-719
- complexity, 714-715
- computer programs/systems, 362
- detailed checklist, 715
- free-form, 715
- integration, 717
- objective list, 715
- script, 715
- unit, 716-717

**TESTMENU.C source file, 460-462**

**text**

- colored, 264
- colors, 303-304
- controls, ANSI codes, 107-108
- monochrome, 264
- repeating characters, 278-279
- screen colors, 276-278
- strings
  - converting integers to, 684-685*
  - entering, 325-327*
  - in color, 282-283*
  - NULL, 17*
  - trash, 12*
  - versus character arrays, 37-38*
- undeleting, 577

**text mode, 534**

**TIME.H header file, 783**

**\_\_TIME\_\_ predefined constant, 160**

**timer (system), returning tick counter value, 323-324**

**tiny memory model, 755-756**

**TLIB library program, 209**

**toupper() function, 157**

**tracking user questions during program/system design, 366**

**trash string, 12**

**tutorials, 527-528**

**TYAC.H header file, 266-268, 539-540**

- for getline() function, 297-302
- updating, 289-291

**TYAC.LIB library file, updating, 289-291**

**Type of Music menu (Record of Records! application), creating, 472-475**

**typedef keyword, 34, 775**

## U

**UCHAR\_MAX** constant, 146  
**UINT** data type, 34  
**UINT\_MAX** constant, 147  
**ULONG\_MAX** constant, 147  
 Undefined symbol ‘\_xxxx\_’ in  
   function main error message, 160  
 undeleting, 577  
 underlining `getline()` data-retrieval  
   fields, 410-411  
**union** keyword, 775  
**UNION.C** source file, 42-44  
**unions**, 41-44  
   portability, 157-159  
   REGS, 234-236  
 unit testing, 362, 716-717  
**unsigned** char data type, maximum/  
   minimum values, 33  
**unsigned int** data type, maximum/  
   minimum values, 33  
**unsigned integer** variables, 144  
**unsigned** keyword, 775  
**unsigned long int** data type, maximum/  
   minimum values, 33  
**unsigned short int** data type, maximum/  
   minimum values, 33  
**up** arrow key  
   Medium Code entry and edit  
   screen, 412  
   menu scan codes, 459  
   standards for navigating entry and edit  
   windows, 387  
**update\_header()** function, 546-548  
**update\_songs()** function, 635  
**updates**, 732-733  
**updating**  
   functions, 271-272  
   header information in records, 546-548  
   library functions, 221  
   songs, 628-635  
   TYAC.H header file, 289-291  
   TYAC.LIB library file, 289-291  
**UPDHDR.C** source file, 546-547  
**upgrades**, 732-733  
**user** guides, 528  
**user** interfaces, 382-384  
   action bars, 482-506  
   menus, 444-475  
   standards for creating, 384-390  
**users**

input, pausing for, 288-289  
 questions, tracking during program/  
   system design, 366

**USHRT\_MAX** constant, 146

## V

### values

ASCII Character Table, displaying,  
 170-176  
 basic data types,  
   maximum/minimum, 33  
 numeric variable types, setting  
   maximum/minimum, 145-151  
 variable types, determining classifica-  
   tions, 151-156

### variable-length structures, 58-63

### variables, 33

Borland, directvideo, 128  
 classifying, 151-156  
 finding addresses, 45-47  
 numeric  
   *determining type sizes, 143-145*  
   *setting maximum/minimum values,*  
   *145-151*  
   types and sizes, 4-6

### verify\_mdata() function, 565

### verifying birthdate entries, 222-223

### Vermont Views Plus (Vermont Creative Software), 749

### video display

ANSI escape sequences, *see* ANSI,  
 escape sequences  
 BIOS, accessing memory with,  
   128-129, 232-237  
 direct memory access, 128, 231-232  
 erasing screens, 115-121  
 moving cursor, 113-121, 269-272  
   *BIOS functions, 237-241*  
 pausing, 288-289  
 redefining keys, 126-127  
 screen colors, 121-126  
 sizing cursor, BIOS functions, 241-243

### video mode

controls, ANSI codes, 108  
 getting/setting, 273-275

### video pages, 274

### View menu, standards, 485-486

### Virt-Win (TeraTech), 746

### void keyword, 776

### volatile keyword, 776



### W

---

**waitsec() function, 322-323**  
**WAITSEC.C source file, 323**  
**walk-through debugging, 722-723**  
**while keyword, 776**  
**white space, writing code with, 135-138**  
**word alignment in structures, 40-41**  
**WORD data type, 34**  
**WORDREGS structure, 233-235**  
**Wrap up of designing computer programs/systems, 362-363**  
**write\_char() function, 276-278**  
**write\_string() function, 282-283, 458-459**  
**WRITECH.C source file, 276-277**  
**WRITESTR.C source file, 282**  
**writing**  
    code with white space, 135-138  
    compilers, 739  
    records to files, 545-546

### X-Z

---

**XREG structure, 235-236**  
**yes\_no\_box() function, 401-402**  
**zero\_fill\_field() function, 401-402, 424-425**