

Contenido

Prefacio ix

1. Fundamentos de C 1

- 1.1. Comprensión de los componentes de un programa en C 2
- 1.2. Creación y compilación de un programa 5
- 1.3. Declaración de variables y asignación de valores 7
- 1.4. Introducción de números desde el teclado 11
- 1.5. Realización de cálculos utilizando expresiones aritméticas 13
- 1.6. Cómo añadir comentarios a un programa 15
- 1.7. Escriba sus propias funciones 17
- 1.8. Utilización de funciones para devolver valores 20
- 1.9. Utilización de argumentos de función 23
- 1.10. Repaso a las palabras clave de C 25

2. Introducción a las sentencias de control de programa de C 27

- 2.1. Familiarización con la sentencia `if` 28
- 2.2. Cómo añadir la sentencia `else` 30
- 2.3. Creación de bloques de código 32
- 2.4. Utilización del bucle `for` 34
- 2.5. Sustitución de operadores de incremento y de decremento de C 37
- 2.6. Extensión de las capacidades de `printf()` 40
- 2.7. Programación con operadores lógicos y relacionales de C 43

3. Más sentencias de control de programa de C 47

- 3.1. Introducción de caracteres 48
- 3.2. Anidamiento de sentencias `if` 51
- 3.3. Estudio de las variaciones del bucle `for` 54
- 3.4. Comprensión del bucle `while` de C 57

- 3.5. Utilización del bucle **do** 58
- 3.6. Creación de bucles anidados 60
- 3.7. Utilización de **break** para salir de un bucle 62
- 3.8. Cómo saber cuándo utilizar la sentencia **continue** 64
- 3.9. Selección de caminos con la sentencia **switch** 66
- 3.10. Comprensión de la sentencia **goto** 71

4. Un estudio más profundo de los tipos de datos, variables y expresiones 75

- 4.1. Utilización de los modificadores de tipo de datos de C 76
- 4.2. Aprendizaje de dónde se declaran las variables 80
- 4.3. Una mayor aproximación a las constantes 85
- 4.4. Inicialización de variables 88
- 4.5. Comprensión de las conversiones de tipo en expresiones 89
- 4.6. Comprensión de las conversiones del tipo en asignaciones 92
- 4.7. Programación con moldes de tipo 94

5. Exploración de arrays y cadenas 97

- 5.1. Declaración de arrays unidimensionales 98
- 5.2. Utilización de cadenas 103
- 5.3. Creación de arrays multidimensionales 108
- 5.4. Inicialización de arrays 111
- 5.5. Construcción de arrays de cadenas 114

6. Utilización de punteros 119

- 6.1. Comprensión de los fundamentos de los punteros 120
- 6.2. Aprendizaje de las restricciones de las expresiones de punteros 125
- 6.3. Utilización de punteros con arrays 128
- 6.4. Utilización de punteros a constantes de cadena 134
- 6.5. Creación de arrays de punteros 135
- 6.6. Familiarización con la indirección múltiple 137
- 6.7. Utilización de punteros como parámetros 139

7. Una mayor aproximación a las funciones 143

- 7.1. Creación de funciones que devuelven valores 144
- 7.2. Utilización de prototipos de función 148
- 7.3. Comprensión de la recursividad 151
- 7.4. Un mayor acercamiento a los parámetros 155
- 7.5. Paso de argumentos a **main()** 158
- 7.6. Comparación de las declaraciones de parámetros de función clásicas y modernas 162

8. Entrada/Salida por consola	167
8.1. Aprendizaje de otra directiva del preprocesador	168
8.2. Examen de la entrada y salida de caracteres y de cadenas	172
8.3. Examen de algunas funciones de consola no estándar	174
8.4. Un mayor acercamiento a gets() y puts()	176
8.5. Dominio de printf()	178
8.6. Dominio de scanf()	182
9. Entrada/Salida por archivos	189
9.1. Comprensión de los fundamentos de los flujos	190
9.2. Dominio de los fundamentos del sistema de archivos	191
9.3. Comprensión de feof() y ferror()	199
9.4. Aprendizaje de algunas funciones de texto de más alto nivel	202
9.5. Aprendizaje de lectura y escritura de datos binarios	206
9.6. Comprensión del acceso aleatorio	211
9.7. Aprendizaje de diversas funciones del sistema de archivos	215
9.8. Aprendizaje de los flujos estándar	217
10. Estructuras y uniones	221
10.1. Dominio de los fundamentos de la estructura	222
10.2. Declaración de punteros a estructuras	233
10.3. Trabajo con estructuras anidadas	236
10.4. Comprensión de los campos de bits	241
10.5. Creación de uniones	244
11. Tipos de datos y operadores avanzados	251
11.1. Utilización de los especificadores de clase de almacenamiento	252
11.2. Utilización de los modificadores de acceso	260
11.3. Definición de enumeraciones	263
11.4. Comprensión de typedef	266
11.5. Utilización de los operadores a nivel de bit	268
11.6. Dominio de los operadores de desplazamiento	272
11.7. Comprensión del operador ?	274
11.8. Más aplicaciones del operador de asignación	276
11.9. Comprensión del operador coma	278
11.10. Conocimiento del resumen de prioridades	279
12. El preprocesador de C y algunos temas avanzados	283
12.1. Un mayor aprendizaje de #define e #include	284
12.2. Comprensión de la compilación condicional	288
12.3. Aprendizaje #error , #undef , #line y #pragma	293

- 12.4. Examen de las macros incorporadas de C 295
- 12.5. Utilización de los operadores # y ## 297
- 12.6. Comprensión de los punteros a función 298
- 12.7. Dominio de la asignación dinámica 304

A. Algunas funciones comunes de la biblioteca de C 311

- A.1. Funciones de caracteres y de cadenas 312
- A.2. Las funciones matemáticas 322
- A.3. Funciones de hora y fecha 330
- A.4. Asignación dinámica 335
- A.5. Otras funciones 338

B. Resumen de palabras clave de C 349

C. Una visión general de la programación en Windows 361

- C.1. ¿Qué es Windows? 362
- C.2. ¿Cómo interactúa Windows y su programa? 364
- C.3. Windows es multitarea 365
- C.4. El API 365
- C.5. Los componentes de una ventana 366
- C.6. Algunos fundamentos de una aplicación de Windows 367
- C.7. Un esqueleto de Windows 369
- C.8. El archivo de definición del esqueleto 378
- C.9. Compilación de un programa de Windows 379
- C.10. Cómo dar nombre a convenios 380
- C.11. Para aprender más 380

D. Respuestas 383

Índice 497

7.2

7.3

7.4 uno

7.5 Pasap

7.8. C

7.8.1