

CONTENIDO

Prólogo	xv
----------------------	----

PARTE I. METODOLOGÍA DE LA PROGRAMACIÓN

Capítulo 1. Introducción a la ciencia de la computación y a la programación	2
1.1. ¿Qué es una computadora?	4
1.2. Organización física de una computadora (hardware)	4
1.2.1. Dispositivos de Entrada/Salida (E/S)	5
1.2.2. La memoria central (interna)	6
1.2.3. La Unidad Central de Proceso (UCP)	9
1.2.4. El microprocesador	10
1.2.5. Memoria auxiliar (externa)	10
1.2.6. Proceso de ejecución de un programa	12
1.2.7. Comunicaciones: módems, redes, telefonía RDSI y ADSL	12
1.2.8. La computadora personal multimedia ideal para la programación	13
1.3. Concepto de algoritmo	15
1.3.1. Características de los algoritmos	16
1.4. El software (los programas)	17
1.5. Los lenguajes de programación	19
1.5.1. Instrucciones a la computadora	20
1.5.2. Lenguajes máquina	20
1.5.3. Lenguajes de bajo nivel	21
1.5.4. Lenguajes de alto nivel	22
1.5.5. Traductores de lenguaje	22
1.5.5.1. Intérpretes	23
1.5.5.2. Compiladores	23
1.5.6. La compilación y sus fases	23
1.6. El lenguaje C: historia y características	25
1.6.1. Ventajas de C	25
1.6.2. Características técnicas de C	26
1.6.3. Versiones actuales de C	26
1.7. Resumen	27
Capítulo 2. Fundamentos de programación	28
2.1. Fases en la resolución de problemas	30
2.1.1. Análisis del problema	31
2.1.2. Diseño del algoritmo	32
2.1.3. Herramientas de la programación	33
2.1.4. Codificación de un programa	36

2.1.5. Compilación y ejecución de un programa	37
2.1.6. Verificación y depuración de un programa	38
2.1.7. Documentación y mantenimiento	38
2.2. Programación modular	49
2.3. Programación estructurada	40
2.3.1. Recursos abstractos	40
2.3.2. Diseño descendente (<i>top-down</i>)	40
2.3.3. Estructuras de control	41
2.3.4. Teorema de la programación estructurada: estructuras básicas	42
2.4. Representación gráfica de los algoritmos	42
2.4.1. Diagramas de flujo	43
2.5. Diagramas de Nassi-Schneiderman (N-S)	52
2.6. El ciclo de vida del software	53
2.6.1. Análisis	54
2.6.2. Diseño	55
2.6.3. Implementación (codificación)	55
2.6.4. Pruebas e integración	56
2.6.5. Verificación	56
2.6.6. Mantenimiento	57
2.6.7. La <i>obsolescencia</i> : programas obsoletos	57
2.6.8. Iteración y evolución del software	57
2.7. Métodos formales de verificación de programas	58
2.7.1. Aserciones	58
2.7.2. Precondiciones y postcondiciones	59
2.7.3. Reglas para prueba de programas	60
2.7.4. Invariantes de bucles	60
2.7.5. Etapas a establecer la exactitud (corrección) de un programa	62
2.7.6. Programación segura contra fallos	63
2.8. Factores en la calidad del software	64
2.9. Resumen	65
2.10. Ejercicios	65
2.11. Ejercicios resueltos	66

PARTE II. FUNDAMENTOS DE PROGRAMACIÓN EN C

Capítulo 3. El lenguaje C: elementos básicos	72
3.1. Estructura general de un programa en C	74
3.1.1. Directivas del preprocesador	76
3.1.2. Declaraciones globales	77
3.1.3. Función <code>main()</code>	78
3.1.4. Funciones definidas por el usuario	78
3.1.5. Comentarios	80
3.2. Creación de un programa	82
3.3. El proceso de ejecución de un programa en C	83
3.4. Depuración de un programa en C	86
3.4.1. Errores de sintaxis	86
3.4.2. Errores lógicos	87
3.4.3. Errores de regresión	88
3.4.4. Mensajes de error	88
3.4.5. Errores en tiempo de ejecución	88
3.5. Pruebas	89
3.6. Los elementos de un programa en C	90
3.6.1. Tokens (elementos léxicos de los programas)	90
3.6.2. Identificadores	90
3.6.3. Palabras reservadas	91
3.6.4. Comentarios	91

3.6.5. Signos de puntuación y separadores	92
3.6.6. Archivos de cabecera	92
3.7. Tipos de datos en C	92
3.7.1. Enteros (int)	93
3.7.2. Tipos de coma flotante (float/double)	94
3.7.3. Caracteres (char)	95
3.8. El tipo de dato LÓGICO	96
3.8.1. Escritura de valores lógicos	97
3.9. Constantes	97
3.9.1. Constantes literales	98
3.9.2. Constantes definidas (simbólicas)	101
3.9.3. Constantes enumeradas	101
3.9.4. Constantes declaradas <i>const</i> y <i>volatile</i>	101
3.10. Variables	103
3.10.1. Declaración	103
3.10.2. Inicialización de variables	105
3.10.3. Declaración o definición	105
3.11. Duración de una variable	106
3.11.1. Variables locales	106
3.11.2. Variables globales	106
3.11.3. Variables dinámicas	107
3.12. Entradas y salidas	108
3.12.1. Salida	108
3.12.2. Entrada	111
3.12.3. Salida de cadenas de caracteres	112
3.12.4. Entrada de cadenas de caracteres	112
3.13. Resumen	113
3.14. Ejercicios	113
Capítulo 4. Operadores y expresiones	114
4.1. Operadores y expresiones	116
4.2. Operador de asignación	116
4.3. Operadores aritméticos	117
4.3.1. Asociatividad	119
4.3.2. Uso de paréntesis	120
4.4. Operadores de incrementación y decrementación	120
4.5. Operadores relacionales	123
4.6. Operadores lógicos	125
4.6.1. Evaluación en cortocircuito	127
4.6.2. Asignaciones <i>booleanas</i> (lógicas)	128
4.7. Operadores de manipulación de bits	129
4.7.1. Operadores de asignación adicionales	130
4.7.2. Operadores de desplazamiento de bits (\gg , \ll)	131
4.7.3. Operadores de direcciones	131
4.8. Operador condicional	132
4.9. Operador coma	133
4.10. Operadores especiales (), []	133
4.10.1. El operador ()	133
4.10.2. El operador []	133
4.11. El operador <i>sizeof</i>	134
4.12. Conversiones de tipos	135
4.12.1. Conversión implícita	135
4.12.2. Reglas	135
4.12.3. Conversión explícita	136
4.13. Prioridad y asociatividad	136
4.14. Resumen	137

4.15. Ejercicios	137
4.16. Problemas	139
Capítulo 5. Estructuras de selección: sentencias if y switch	142
5.1. Estructuras de control	144
5.2. La sentencia if	144
5.3. Sentencia if de dos alternativas: if-else	147
5.4. Sentencias if-else anidadas	150
5.4.1. Sangría en las sentencias if anidadas	151
5.4.2. Comparación de sentencias if anidadas y secuencias de sentencias if	152
5.5. Sentencia de control switch	154
5.5.1. Caso particular de case	159
5.5.2. Uso de sentencias switch en menús	159
5.6. Expresiones condicionales: el operador ?:	159
5.7. Evaluación en cortocircuito de expresiones lógicas	161
5.8. Puesta a punto de programas	162
5.9. Errores frecuentes de programación	163
5.10. Resumen	164
5.11. Ejercicios	165
5.12. Problemas	167
Capítulo 6. Estructuras de control: bucles	168
6.1. La sentencia while	170
6.1.1. Operadores de incremento y decremento (++,--)	172
6.1.2. Terminaciones anormales de un ciclo	174
6.1.3. Diseño eficiente de bucles	174
6.1.4. Bucles while con cero iteraciones	174
6.1.5. Bucles controlados por centinelas	175
6.1.6. Bucles controlados por indicadores (banderas)	175
6.1.7. La sentencia break en los bucles	177
6.1.8. Bucles while (true)	178
6.2. Repetición: el bucle for	180
6.2.1. Diferentes usos de bucles for	184
6.3. Precauciones en el uso de for	185
6.3.1. Bucles infinitos	186
6.3.2. Los bucles for vacíos	187
6.3.3. Sentencias nulas en bucles for	188
6.3.4. Sentencias break y continue	188
6.4. Repetición: el bucle do...while	190
6.4.1. Diferencias entre while y do-while	191
6.5. Comparación de bucles while, for y do-while	193
6.6. Diseño de bucles	193
6.6.1. Bucles para diseño de sumas y productos	194
6.6.2. Fin de un bucle	194
6.6.3. Otras técnicas de terminación de bucle	196
6.6.4. Bucles for vacíos	196
6.7. Bucles anidados	197
6.8. Resumen	201
6.9. Ejercicios	201
6.10. Problemas	203
6.11. Proyectos de programación	206
Capítulo 7. Funciones	208
7.1. Concepto de función	210
7.2. Estructura de una función	211
7.2.1. Nombre de una función	213

- 7.2.2. Tipo de dato de retorno 213
- 7.2.3. Resultados de una función 214
- 7.2.4. Llamada a una función 215
- 7.3. Prototipos de las funciones 217
 - 7.3.1. Prototipos con un número no especificado de parámetros 220
- 7.4. Parámetros de una función 222
 - 7.4.1. Paso de parámetros por valor 222
 - 7.4.2. Paso de parámetros por referencia 223
 - 7.4.3. Diferencias entre paso de variables por valor y por referencia 224
 - 7.4.4. Parámetros const de una función 225
- 7.5. Funciones en línea, macros con argumentos 226
 - 7.5.1. Creación de macros con argumentos 227
- 7.6. Ámbito (alcance) 228
 - 7.6.1. Ámbito del programa 229
 - 7.6.2. Ámbito del archivo fuente 230
 - 7.6.3. Ámbito de una función 230
 - 7.6.4. Ámbito de bloque 230
 - 7.6.5. Variables locales 231
- 7.7. Clases de almacenamiento 231
 - 7.7.1. Variables automáticas 231
 - 7.7.2. Variables externas 231
 - 7.7.3. Variables registro 232
 - 7.7.4. Variables estáticas 232
- 7.8. Concepto y uso de funciones de biblioteca 234
- 7.9. Funciones de carácter 234
 - 7.9.1. Comprobación alfabética y de dígitos 235
 - 7.9.2. Funciones de prueba de caracteres especiales 236
 - 7.9.3. Funciones de conversión de caracteres 236
- 7.10. Funciones numéricas 237
 - 7.10.1. Funciones matemáticas 237
 - 7.10.2. Funciones trigonométricas 238
 - 7.10.3. Funciones logarítmicas y exponenciales 238
 - 7.10.4. Funciones aleatorias 239
- 7.11. Funciones de fecha y hora 240
- 7.12. Funciones de utilidad 243
- 7.13. Visibilidad de una función 244
 - 7.13.1. Variables locales frente a variables globales 245
 - 7.13.2. Variables estáticas y automáticas 247
- 7.14. Compilación separada 249
- 7.15. Variables registro (register) 250
- 7.16. Recursividad 251
- 7.17. Resumen 254
- 7.18. Ejercicios 255
- 7.19. Problemas 255

- Capítulo 8. Arrays (listas y tablas) 258**
 - 8.1. Arrays 260
 - 8.1.1. Declaración de un array 260
 - 8.1.2. Subíndices de un array 261
 - 8.1.3. Almacenamiento en memoria de los arrays 262
 - 8.1.4. El tamaño de los arrays 263
 - 8.1.5. Verificación del rango del índice de un array 264
 - 8.2. Inicialización de un array 264
 - 8.3. Arrays de caracteres y cadenas de texto 266
 - 8.4. Arrays multidimensionales 269
 - 8.4.1. Inicialización de arrays multidimensionales 270

8.4.2.	Acceso a los elementos de los arrays bidimensionales	271
8.4.3.	Lectura y escritura de arrays bidimensionales	272
8.4.4.	Acceso a elementos mediante bucles	272
8.4.5.	Arrays de más de dos dimensiones	274
8.4.6.	Una aplicación práctica	274
8.5.	Utilización de arrays como parámetros	276
8.5.1.	Precauciones	279
8.5.2.	Paso de cadenas como parámetros	281
8.6.	Ordenación de listas	282
8.6.1.	Algoritmo de la burbuja	282
8.7.	Búsqueda en listas	284
8.7.1.	Búsqueda secuencial	285
8.8.	Resumen	289
8.9.	Ejercicios	290
8.10.	Problemas	291
Capítulo 9.	Estructuras y uniones	294
9.1.	Estructuras	296
9.1.1.	Declaración de una estructura	297
9.1.2.	Definición de variables de estructuras	297
9.1.3.	Uso de estructuras en asignaciones	298
9.1.4.	Inicialización de una declaración de estructuras	299
9.1.5.	El tamaño de una estructura	300
9.2.	Acceso a estructuras	300
9.2.1.	Almacenamiento de información en estructuras	300
9.2.2.	Lectura de información de una estructura	302
9.2.3.	Recuperación de información de una estructura	302
9.3.	Estructuras anidadas	303
9.3.1.	Ejemplo de estructuras anidadas	304
9.4.	Arrays de estructuras	307
9.4.1.	Arrays como miembros	308
9.5.	Utilización de estructuras como parámetros	309
9.6.	Uniones	310
9.7.	Enumeraciones	311
9.7.1.	sizeof de tipos de datos estructurados	314
9.7.2.	typedef	314
9.8.	Campos de bit	315
9.9.	Resumen	319
9.10.	Ejercicios	320
9.11.	Problemas	321
Capítulo 10.	Punteros (apuntadores)	322
10.1.	Direcciones en memoria	324
10.2.	Concepto de puntero (apuntador)	325
10.2.1.	Declaración de punteros	326
10.2.2.	Inicialización (iniciación) de punteros	327
10.2.3.	Indirección de punteros	328
10.2.4.	Punteros y verificación de tipos	329
10.3.	Punteros null y void	330
10.4.	Punteros a punteros	331
10.5.	Punteros y arrays	332
10.5.1.	Nombres de arrays como punteros	332
10.5.2.	Ventajas de los punteros	333
10.6.	Arrays de punteros	334
10.6.1.	Inicialización de un array de punteros a cadenas	335
10.7.	Punteros de cadenas	335
10.7.1.	Punteros <i>versus</i> arrays	335

10.8. Aritmética de punteros	336
10.8.1. Una aplicación de punteros: conversión de caracteres	338
10.9. Punteros constantes frente a punteros a constantes	339
10.9.1. Punteros constantes	339
10.9.2. Punteros a constantes	339
10.9.3. Punteros constantes a constantes	340
10.10. Punteros como argumentos de funciones	341
10.11. Punteros a funciones	343
10.11.1. Inicialización de un puntero a una función	343
10.11.2. Aplicación de punteros a función para ordenación	347
10.11.3. Arrays de punteros de funciones	348
10.11.4. Una aplicación de punteros de funciones	349
10.12. Punteros a estructuras	350
10.13. Resumen	351
10.14. Ejercicios	352
10.15. Problemas	353
Capítulo 11. Asignación dinámica de memoria	354
11.1. Gestión dinámica de la memoria	356
11.1.1. Almacén libre (<i>free store</i>)	357
11.2. Función <code>malloc ()</code>	357
11.2.1. Asignación de memoria de un tamaño desconocido	361
11.2.2. Uso de <code>malloc ()</code> para arrays multidimensionales	362
11.3. Liberación de memoria, función <code>free ()</code>	363
11.4. Funciones de asignación de memoria <code>calloc ()</code> y <code>realloc ()</code>	364
11.4.1. Función <code>calloc ()</code>	364
11.4.2. Función <code>realloc ()</code>	365
11.5. Asignación de memoria para arrays	368
11.5.1. Asignación de memoria interactivamente	369
11.5.2. Asignación de memoria para un array de estructuras	371
11.6. Arrays dinámicos	373
11.7. Reglas de funcionamiento de la asignación de memoria	374
11.8. Resumen	376
11.9. Ejercicios	376
11.10. Problemas	377
Capítulo 12. Cadenas	378
12.1. Concepto de cadena	380
12.1.1. Declaración de variables de cadena	381
12.1.2. Inicialización de variables de cadena	381
12.2. Lectura de cadenas	382
12.2.1. Función <code>getchar ()</code>	385
12.2.2. Función <code>putchar ()</code>	385
12.2.3. Función <code>puts ()</code>	386
12.2.4. Funciones <code>getch ()</code> y <code>getche ()</code>	387
12.3. La biblioteca <code>string.h</code>	387
12.3.1. La palabra reservada <code>const</code>	389
12.4. Arrays y cadenas como parámetros de funciones	389
12.5. Asignación de cadenas	391
12.5.1. La función <code>strncpy ()</code>	391
12.6. Longitud y concatenación de cadenas	392
12.6.1. La función <code>strlen ()</code>	392
12.6.2. Las funciones <code>strcat ()</code> y <code>strncat ()</code>	393
12.7. Comparación de cadenas	394
12.7.1. La función <code>strcmp ()</code>	395
12.7.2. La función <code>strcmp ()</code>	395

12.7.3. La función <code>strncmp ()</code>	396
12.7.4. La función <code>strnicmp ()</code>	396
12.8. Inversión de cadenas	397
12.9. Conversión de cadenas	397
12.9.1. Función <code>strupr ()</code>	397
12.9.2. Función <code>strlwr ()</code>	398
12.10. Conversión de cadenas a números	399
12.10.1. Función <code>atoi ()</code>	399
12.10.2. Función <code>atof ()</code>	400
12.10.3. Función <code>atol ()</code>	400
12.10.4. Entrada de números y cadenas	400
12.11. Búsqueda de caracteres y cadenas	401
12.11.1. Función <code>strchr ()</code>	401
12.11.2. Función <code>strrchr ()</code>	402
12.11.3. Función <code>strspn ()</code>	402
12.11.4. Función <code>strcspn ()</code>	403
12.11.5. Función <code>strpbrk ()</code>	403
12.11.6. Función <code>strstr ()</code>	404
12.11.7. Función <code>strtok ()</code>	404
12.12. Resumen	405
12.13. Ejercicios	406
12.14. Problemas	407

PARTE III. ESTRUCTURA DE DATOS

Capítulo 13. Entradas y salidas por archivos	410
13.1. Flujos	412
13.2. Puntero <code>FILE</code>	412
13.3. Apertura de un archivo	413
13.3.1. Modos de apertura de un archivo	414
13.3.2. <code>NULL</code> y <code>EOF</code>	415
13.3.3. Cierre de archivos	415
13.4. Creación de un archivo secuencial	416
13.4.1. Funciones <code>putc ()</code> y <code>fputc ()</code>	416
13.4.2. Funciones <code>getc ()</code> y <code>fgetc ()</code>	417
13.4.3. Funciones <code>fputs ()</code> y <code>fgets ()</code>	418
13.4.4. Funciones <code>fprint ()</code> y <code>fscanf ()</code>	419
13.4.5. Función <code>feof ()</code>	421
13.4.6. Función <code>rewind ()</code>	421
13.5. Archivos binarios en C	422
13.5.1. Función de salida <code>fwrite ()</code>	423
13.5.2. Función de lectura <code>fread ()</code>	424
13.6. Funciones para acceso aleatorio	426
13.6.1. Función <code>fseek ()</code>	426
13.6.2. Función <code>ftell ()</code>	431
13.7. Datos externos al programa con argumentos de <code>main ()</code>	432
13.8. Resumen	434
13.9. Ejercicios	435
13.10. Problemas	436
Capítulo 14. Listas enlazadas	438
14.1. Fundamentos teóricos	440
14.2. Clasificación de las listas enlazadas	441
14.3. Operaciones en listas enlazadas	442
14.3.1. Declaración de un nodo	442

14.3.2. Puntero de cabecera y cola	443
14.3.3. El puntero nulo	444
14.3.4. El operador \rightarrow de selección de un miembro	445
14.3.5. Construcción de una lista	445
14.3.6. Insertar un elemento en una lista	447
14.3.7. Búsqueda de un elemento	453
14.3.8. Supresión de un nodo en una lista	454
14.4. Lista doblemente enlazada	456
14.4.1. Declaración de una lista doblemente enlazada	457
14.4.2. Insertar un elemento en una lista doblemente enlazada	458
14.4.3. Supresión de un elemento en una lista doblemente enlazada	459
14.5. Listas circulares	462
14.5.1. Insertar un elemento en una lista circular	462
14.5.2. Supresión de un elemento en una lista circular	463
14.6. Resumen	467
14.7. Ejercicios	468
14.8. Problemas	468
Capítulo 15. Pilas y colas	470
15.1. Concepto de pila	472
15.1.1. Especificaciones de una pila	473
15.2. El tipo pila implementado con arrays	473
15.2.1. Especificación del tipo pila	475
15.2.2. Implementación de las operaciones sobre pilas	477
15.2.3. Operaciones de verificación del estado de la pila	478
15.3. Colas	481
15.4. El tipo cola implementada con arrays	483
15.4.1. Definición de la especificación de una cola	483
15.4.2. Especificación del tipo cola	483
15.4.3. Implementación del tipo cola	484
15.4.4. Operaciones de la cola	486
15.5. Realización de una cola con una lista enlazada	487
15.5.1. Declaración del tipo cola con listas	488
15.5.2. Codificación de las operaciones del tipo cola con listas	489
15.6. Resumen	492
15.7. Ejercicios	493
15.8. Problemas	494
Capítulo 16. Árboles	496
16.1. Árboles generales	498
16.1.1. Representación de un árbol	502
16.2. Resumen de definiciones	504
16.3. Árboles binarios	505
16.3.1. Equilibrio	507
16.3.2. Árboles binarios completos	508
16.4. Estructura de un árbol binario	511
16.4.1. Diferentes tipos de representaciones en C	512
16.5. Operaciones en árboles binarios	513
16.6. Árboles de expresión	514
16.6.1. Reglas para la construcción de árboles de expresión	516
16.7. Recorrido de un árbol	518
16.7.1. Recorrido <i>preorden</i>	519
16.7.2. Recorrido <i>enorden</i>	521
16.7.3. Recorrido <i>postorden</i>	522
16.7.4. Profundidad de un árbol binario	524
16.8. Árbol binario de búsqueda	525

16.8.1. Creación de un árbol binario de búsqueda	525
16.8.2. Implementación de un nodo de un árbol binario de búsqueda	527
16.9. Operaciones en árboles binarios de búsqueda	528
16.9.1. Búsqueda	528
16.9.2. Insertar un nodo	529
16.9.3. Función insertar()	530
16.9.4. Eliminación	531
16.9.5. Recorridos de un árbol	535
16.9.6. Determinación de la altura de un árbol	535
16.10. Aplicaciones de árboles en algoritmos de exploración	536
16.10.1. Visita a los nodos de un árbol	537
16.11. Resumen	537
16.12. Ejercicios	539
16.13. Problemas	540
16.14. Referencias bibliográficas para lecturas posteriores	542
APÉNDICES	543
Apéndice A. Lenguajes C ANSI. Guía de referencia	545
Apéndice B. Códigos de caracteres ASCII	575
Apéndice C. Palabras reservadas de C++	579
Apéndice D. Guía de sintaxis ANSI/ISO estándar C++	599
Apéndice E. Biblioteca de funciones ANSI C	649
Apéndice F. Recursos (Libros/Revistas/URL de Internet de C/C++)	713
ÍNDICE	727