

Contenido

1	Introducción a los algoritmos y a la programación de computadoras	1
1.1	Introducción.....	2
1.2	Concepto de algoritmo.....	2
1.2.1	Definición de algoritmo y problema.....	2
1.2.2	Análisis del enunciado de un problema.....	3
1.2.3	Memoria y operaciones aritméticas y lógicas.....	4
1.2.4	Teorema de la programación estructurada.....	4
1.3	Conceptos de programación.....	6
1.3.1	Lenguajes de programación.....	6
1.3.2	Codificación de un algoritmo.....	7
1.3.3	Bibliotecas de funciones.....	7
1.3.4	Programas de computación.....	7
1.3.5	Consola.....	7
1.3.6	Entrada y salida de datos.....	8
1.3.7	Lenguajes algorítmicos.....	8
1.3.8	Pseudocódigo.....	8
1.4	Representación gráfica de algoritmos.....	8
1.4.1	Representación gráfica de la estructura secuencial o acción simple.....	9
1.4.2	Representación gráfica de la estructura de decisión.....	9
1.4.3	Representación gráfica de la estructura de repetición.....	9
1.4.4	Representación gráfica de módulos o funciones.....	10
1.5	Nuestro primer programa.....	12
1.5.1	Codificación del algoritmo utilizando el lenguaje C.....	12
1.5.2	El archivo de código fuente.....	14
1.5.3	Comentarios en el código fuente.....	14
1.5.4	La compilación y el programa ejecutable.....	14
1.5.5	El entorno integrado de desarrollo (IDE).....	15
1.6	La memoria de la computadora.....	16
1.6.1	El byte.....	17
1.6.2	Conversión numérica: de base 2 a base 10.....	17
1.6.3	Dimensionamiento de los datos.....	17
1.6.4	Los números negativos.....	18
1.6.5	Los caracteres.....	19
1.7	Las variables.....	19
1.7.1	Convención de nomenclatura para variables.....	20
1.7.2	Los tipos de datos.....	20
1.7.3	Los tipos de datos provistos por el lenguaje C.....	21
1.7.4	La función de biblioteca printf.....	22
1.7.5	La función de biblioteca scanf.....	23
1.7.6	El operador de dirección &.....	24
1.7.7	Las constantes.....	24
1.7.8	Nomenclatura para las constantes.....	24
1.8	Operadores aritméticos.....	25
1.8.1	Conversión de tipos de datos (type casting).....	28
1.8.2	El operador % ("módulo" o "resto").....	28
1.8.3	Operadores relacionales.....	31
1.9	Expresiones lógicas.....	32
1.9.1	Operadores lógicos.....	32
1.10	Operadores de bits.....	33

1.10.1	Representación binaria de los tipos enteros.....	33
1.10.2	Operadores de desplazamiento de bits (>> y <<).....	34
1.10.3	Representación hexadecimal.....	34
1.10.4	Representación octal.....	35
1.10.5	Operadores lógicos de bits.....	36
1.11	Resumen.....	36
2	Estructuras básicas de control y lógica algorítmica.....	37
2.1	Introducción.....	38
2.2	Estructura secuencial.....	38
2.3	Estructura de decisión.....	38
2.3.1	Estructuras de decisión anidadas.....	40
2.3.2	Selección en línea o if-inline.....	43
2.3.3	Macros.....	44
2.3.4	Selección múltiple (switch).....	46
2.3.5	Asignación de valores alfanuméricos (función strcpy).....	48
2.4	Estructura de repetición.....	50
2.4.1	Estructuras de repetición inexactas.....	50
2.4.2	Estructuras de repetición exactas.....	52
2.4.3	Contadores.....	54
2.4.4	Acumuladores.....	56
2.4.5	Seguimiento del algoritmo y "prueba de escritorio".....	56
2.4.6	El debugger, la herramienta de depuración.....	58
2.4.7	Estructuras de repetición anidadas.....	61
2.4.8	Manejo de valores booleanos.....	63
2.4.9	Máximos y mínimos.....	64
2.5	Contextualización del problema.....	67
2.6	Resumen.....	73
3	Funciones, modularización y metodología top-down.....	75
3.1	Introducción.....	76
3.2	Conceptos iniciales.....	76
3.2.1	Metodología top-down.....	76
3.2.2	Módulos o subprogramas.....	76
3.2.3	Funciones.....	76
3.2.4	Funciones de biblioteca.....	77
3.2.5	Invocación a funciones de biblioteca.....	77
3.3	Funciones definidas por el programador.....	77
3.3.1	Prototipo de una función.....	78
3.3.2	Invocar a una función.....	78
3.3.3	Desarrollo de una función.....	79
3.3.4	Convención de nomenclatura para funciones.....	80
3.3.5	Funciones que no retornan ningún valor (tipo de datos void).....	80
3.3.6	Archivos de cabecera (.h).....	80
3.3.7	Archivos de funciones (.c).....	80
3.4	Legibilidad y reusabilidad del código.....	82
3.4.1	Abstracción.....	82
3.4.2	Argumentos y parámetros.....	84
3.5	Alcance de las variables (scope).....	87
3.5.1	Variables locales.....	87
3.5.2	Variables globales.....	87

3.6	Argumentos por valor y referencia.....	88
3.6.1	Punteros y direcciones de memoria.....	89
3.6.2	El operador de indirección * (asterisco).....	90
3.6.3	Argumentos por referencia.....	91
3.6.4	Funciones que mantienen su estado.....	96
3.6.5	Variables estáticas (modificador static).....	99
3.7	Resumen.....	101
4	Tipos de datos alfanuméricos.....	103
4.1	Introducción.....	104
4.2	Carácter.....	104
4.2.1	El tipo de datos char.....	104
4.2.2	Funciones para tratamiento de caracteres.....	105
4.3	Cadenas de caracteres.....	108
4.3.1	El carácter '\0' (barra cero).....	108
4.3.2	Longitud de una cadena.....	110
4.4	Tratamiento de cadenas de caracteres.....	110
4.4.1	Inicialización de una cadena de caracteres.....	110
4.4.2	Funciones para el tratamiento de cadenas de caracteres.....	111
4.5	Funciones de biblioteca para manejo de cadenas.....	120
4.5.1	Otras funciones de biblioteca.....	121
4.6	Resumen.....	122
5	Punteros a carácter.....	123
5.1	Introducción.....	124
5.2	Conceptos iniciales.....	124
5.2.1	Aritmética de direcciones.....	125
5.2.2	Prefijos y sufijos.....	126
5.3	Funciones que retornan cadenas.....	128
5.3.1	La función malloc.....	129
5.3.2	Subcadenas (función substring).....	130
5.3.3	Función de biblioteca strtok.....	137
5.4	Resumen.....	139
6	Punteros, arrays y aritmética de direcciones.....	141
6.1	Introducción.....	142
6.2	Punteros y direcciones de memoria.....	142
6.2.1	El operador de dirección &.....	142
6.2.2	Los punteros.....	143
6.2.3	El operador de indirección *.....	143
6.2.4	Funciones que reciben punteros.....	144
6.3	Arrays.....	145
6.3.1	La capacidad del array.....	145
6.3.2	Acceso a los elementos de un array.....	145
6.3.3	Dimensionamiento e inicialización de arrays.....	148
6.3.4	Crear arrays dinámicamente (funciones malloc y sizeof).....	148
6.3.5	Punteros genéricos void*.....	149
6.4	Relación entre arrays y punteros.....	149
6.4.1	Aritmética de direcciones.....	150
6.5	Código compacto y eficiente.....	150
6.5.1	Operadores de incremento y decremento (operadores unarios).....	151

6.5.2	“Pre” y “post” incremento y decremento	151
6.5.3	Operadores de asignación	152
6.5.4	Incremento de punteros	152
6.6	Arrays de cadenas	154
6.6.1	Argumentos en línea de comandos (int argc, char* argv[]).....	158
6.7	Resumen	160
7	Tipos de datos estructurados	161
7.1	Introducción	162
7.2	Acceso directo sobre arrays.....	162
7.3	Acceso indirecto sobre arrays.....	170
7.4	Operaciones sobre arrays	170
7.4.1	Capacidad vs. longitud de un array	171
7.4.2	Agregar un elemento al array	172
7.4.3	Búsqueda secuencial	173
7.4.4	Buscar y agregar	175
7.4.5	Insertar un elemento.....	179
7.4.6	Eliminar un elemento	182
7.4.7	Insertar en orden	183
7.4.8	Buscar en orden.....	186
7.4.9	Buscar e insertar en orden.....	187
7.4.10	Ordenar arrays (algoritmo de la “burbuja”).....	188
7.4.11	Búsqueda binaria o dicotómica	191
7.5	Arrays multidimensionales.....	197
7.5.1	Arrays bidimensionales (matrices).....	197
7.5.2	Arrays tridimensionales (cubos).....	201
7.6	Tipos de datos definidos por el programador	202
7.6.1	Introducción al encapsulamiento a través de TADs.....	202
7.6.2	Estructuras o registros	205
7.6.3	Representación gráfica de una estructura.....	205
7.6.4	Estructuras anidadas.....	205
7.6.5	Estructuras con campos de tipo array.....	206
7.6.6	Punteros a estructuras.....	207
7.6.7	Arrays de estructuras	208
7.6.8	Estructuras con campos de tipo “array de estructuras”.....	208
7.7	Resumen	209
8	Operaciones sobre archivos	211
8.1	Introducción	212
8.1.1	Memoria principal o memoria RAM de la computadora	212
8.1.2	Medios de almacenamiento (memoria secundaria)	212
8.2	Archivos.....	212
8.2.1	Abrir un archivo	213
8.2.2	Escribir datos en un archivo	213
8.2.3	Leer datos desde un archivo.....	214
8.2.4	El identificador de posición (puntero)	215
8.2.5	Representación gráfica.....	216
8.2.6	Valor actual del identificador de posición (función ftell).....	217
8.2.7	Manipular el valor del identificador de posición (función fseek).....	218
8.2.8	Calcular el tamaño de un archivo.....	218
8.2.9	Archivos de texto vs. archivos binarios.....	220
8.3	Archivos de registros	221

8.3.1	Archivos de estructuras.....	221
8.3.2	Acceso directo a registros.....	225
8.3.3	Calcular la cantidad de registros que tiene un archivo	228
8.4	Lectura y escritura en bloques (buffers).....	228
8.5	Archivos de texto	230
8.5.1	Apertura de un archivo de texto.....	231
8.5.2	Leer y escribir caracteres (funciones getc y putc)	231
8.5.3	Escribir líneas (función fprintf)	232
8.5.4	Leer líneas (función fgets).....	233
8.5.5	Leer datos formateados (función fscanf).....	233
8.6	Operaciones lógicas sobre archivos	234
8.6.1	Limitaciones de los archivos secuenciales.....	234
8.6.2	Ordenamiento de archivos en memoria	235
8.6.3	Relación entre el número de byte y el número de registro	239
8.6.4	Búsqueda binaria sobre archivos	239
8.6.5	Indexación	241
8.6.6	Indexación de archivos.....	241
8.6.7	Eliminar registros en un archivo (bajas lógicas)	246
8.6.8	Bajas lógicas con soporte en un archivo auxiliar	248
8.7	Resumen	249
9	Tipo Abstracto de Dato (TAD)	251
9.1	Introducción	252
9.2	Capas de abstracción.....	252
9.3	Tipos de datos.....	253
9.3.1	Tipo Abstracto de Dato (TAD)	254
9.3.2	Interfaz e implementación de un TAD	254
9.3.3	El TAD Fecha	255
9.3.4	El TAD XFile (implementación de bajas lógicas).....	257
9.3.4.1	Análisis de la estrategia.....	257
9.4	Resumen	267
10	Análisis de ejercicios integradores	269
10.1	Introducción.....	270
10.2	Problemas con corte de control.....	270
10.2.1	Archivos de novedades vs. archivos maestros	276
10.2.2	Uso de arrays auxiliares	280
10.2.3	Mantener archivos (pequeños) en memoria.....	281
10.3	Apareo de archivos	287
10.3.1	Apareo de archivos con corte de control	292
10.4	Resumen	298
11	Estructuras de datos dinámicas lineales	299
11.1	Introducción.....	300
11.2	Estructuras estáticas	301
11.3	Estructuras dinámicas	301
11.3.1	El nodo	301
11.4	Listas enlazadas.....	302
11.4.1	Estructuras de datos dinámicas lineales.....	302
11.4.2	Estructuras de datos dinámicas no lineales.....	302
11.4.3	Punteros por referencia.....	303

11.5 Operaciones sobre listas enlazadas	304
11.5.1 Agregar un elemento nuevo al final de una lista	304
11.5.2 Recorrer una lista para mostrar su contenido.....	309
11.5.3 Liberar la memoria que utilizan los nodos de una lista enlazada.....	309
11.5.4 Determinar si la lista contiene un valor determinado	311
11.5.5 Eliminar un elemento de la lista.....	314
11.5.6 Insertar un valor respetando el ordenamiento de la lista	316
11.5.7 Insertar un valor solo si la lista aún no lo contiene.....	319
11.6 Estructura Pila (LIFO)	320
11.6.1 Implementación de la estructura pila	320
11.6.2 Operaciones poner (push) y sacar (pop).....	320
11.6.3 Determinar si la pila tiene elementos o no	323
11.7 Estructura Cola (FIFO)	323
11.7.1 Lista enlazada circular	324
11.7.2 Implementar una cola sobre una lista circular.....	324
11.7.3 Operaciones encolar y desencolar.....	326
11.8 Lista doblemente enlazada	328
11.9 Nodos que contienen múltiples datos	329
11.9.1 Nodo con múltiples campos	329
11.9.2 Nodo con un único valor de tipo struct	330
11.10 Estructuras de datos combinadas	332
11.10.1 Lista y sublista	332
11.10.2 Arrays de colas.....	335
11.10.3 Matriz de pilas	343
11.11 Resumen	347
Bibliografía	348