

Содержание

Об авторе	21
Благодарности	21
Предисловие	23
Глава 1. Предварительные сведения	25
Появление языка C	26
Причины популярности языка C	26
Конструктивные особенности	26
Эффективность	27
Переносимость	27
Мощь и гибкость	28
Ориентация на программистов	28
Недостатки	28
Происхождение языка C	29
Особенности функционирования компьютеров	30
Языки программирования высокого уровня и компиляторы	31
Стандарты языка C	32
Первый стандарт ANSI/ISO C	33
Стандарт C99	33
Стандарт C11	34
Использование языка C: семь этапов	35
Этап 1: определение целей программы	35
Этап 2: проектирование программы	35
Этап 3: написание кода	36
Этап 4: компиляция	36
Этап 5: запуск программы на выполнение	37
Этап 6: тестирование и отладка программы	37
Этап 7: сопровождение и модификация программы	38
Комментирование	38
Механика программирования	38
Файлы объектного кода, исполняемые файлы и библиотеки	39
Операционная система Unix	41
Коллекция компиляторов GNU и проект LLVM	43
Системы Linux	43
Компиляторы командной строки для PC	44
Интегрированные среды разработки (Windows)	44
Опция Windows/Linux	46
Работа с языком C в системах Macintosh	46
Как организована эта книга	47
Соглашения, принятые в этой книге	47
Шрифты и начертание	47
Вывод программы	48
Специальные элементы	49
Резюме	49
Вопросы для самоконтроля	50
Упражнения по программированию	50

Глава 2. Введение в язык C	51
Простой пример программы на языке C	52
Пояснение примера	53
Проход 1: краткий обзор	54
Проход 2: нюансы программы	55
Структура простой программы	63
Советы по обеспечению читабельности программ	64
Еще один шаг в использовании языка C	65
Документирование	65
Множественные объявления	66
Умножение	66
Вывод нескольких значений	66
Множество функций	66
Знакомство с отладкой	68
Синтаксические ошибки	68
Семантические ошибки	69
Состояние программы	70
Ключевые слова и зарезервированные идентификаторы	71
Ключевые понятия	72
Резюме	73
Вопросы для самоконтроля	73
Упражнения по программированию	74
Глава 3. Данные в языке C	77
Демонстрационная программа	78
Что нового в этой программе?	79
Переменные и константы	80
Ключевые слова для типов данных	81
Сравнение целочисленных типов и типов с плавающей запятой	82
Целые числа	82
Числа с плавающей запятой	83
Базовые типы данных языка C	84
Тип <code>int</code>	84
Другие целочисленные типы	88
Использование символов: тип <code>char</code>	92
Тип <code>_Bool</code>	98
Переносимые типы: <code>stdint.h</code> и <code>inttypes.h</code>	98
Комплексные и мнимые типы	105
За пределами базовых типов	105
Размеры типов	108
Использование типов данных	108
Аргументы и связанные с ними ловушки	109
Еще один пример: управляющие последовательности	111
Результаты выполнения программы	111
Сброс буфера вывода	112
Ключевые понятия	113
Резюме	113
Вопросы для самоконтроля	114
Упражнения по программированию	116

Глава 4. Символьные строки и форматированный ввод-вывод	117
Вводная программа	118
Введение в символьные строки	119
Массив типа char и нулевой символ	119
Использование строк	120
Функция strlen()	121
Константы и препроцессор C	123
Модификатор const	127
Работа с символическими константами	127
Исследование и эксплуатация функций printf() и scanf()	129
Функция printf()	130
Использование функции printf()	130
Использование функции scanf()	144
Ключевые понятия	151
Резюме	152
Вопросы для самоконтроля	153
Упражнения по программированию	155
Глава 5. Операции, выражения и операторы	157
Введение в циклы	158
Фундаментальные операции	160
Операция присваивания: =	160
Операция сложения: +	163
Операция вычитания: -	163
Операции знака: - и +	163
Операция умножения: *	164
Операция деления: /	166
Приоритеты операций	167
Приоритет и порядок вычисления	169
Некоторые дополнительные операции	170
Операция sizeof и тип size_t	170
Операция деления по модулю: %	171
Операции инкремента и декремента: ++ и --	172
Декрементирование: --	176
Приоритеты операций	177
Не умничайте	177
Выражения и операторы	178
Выражения	179
Операторы	179
Составные операторы (блоки)	182
Преобразования типов	184
Операция приведения	187
Функции с аргументами	188
Демонстрационная программа	190
Ключевые понятия	191
Резюме	192
Вопросы для самоконтроля	193
Упражнения по программированию	196

Глава 6. Управляющие операторы C: циклы	199
Повторный обзор цикла <code>while</code>	200
Комментарии к программе	201
Цикл чтения в стиле C	202
Оператор <code>while</code>	203
Завершение цикла <code>while</code>	204
Когда цикл завершается?	204
Оператор <code>while</code> : цикл с предусловием	205
Особенности синтаксиса	205
Сравнение: операции и выражения отношений	207
Что такое истина?	208
Что еще является истинным?	209
Затруднения с понятием истины	210
Новый тип <code>_Bool</code>	212
Приоритеты операций отношений	213
Неопределенные циклы и циклы со счетчиком	215
Цикл <code>for</code>	216
Использование цикла <code>for</code> для повышения гибкости	217
Дополнительные операции присваивания: <code>+=</code> , <code>-=</code> , <code>*=</code> , <code>/=</code> , <code>%=</code>	221
Операция запятой	222
Греческий философ Зенон и цикл <code>for</code>	225
Цикл с постусловием: <code>do while</code>	226
Выбор подходящего цикла	229
Вложенные циклы	230
Анализ программы	230
Изменение поведения вложенного цикла	230
Введение в массивы	231
Использование цикла <code>for</code> с массивами	233
Пример цикла, использующего возвращаемое значение функции	235
Анализ программы	237
Использование функций с возвращаемыми значениями	238
Ключевые понятия	238
Резюме	239
Вопросы для самоконтроля	240
Упражнения по программированию	243
Глава 7. Управляющие операторы C: ветвление и переходы	247
Оператор <code>if</code>	248
Добавление к оператору <code>if</code> конструкции <code>else</code>	250
Еще один пример: знакомство с функциями <code>getchar()</code> и <code>putchar()</code>	251
Семейство функций для работы с символами <code>ctype.h</code>	254
Множественный выбор <code>else if</code>	255
Образование пар <code>else</code> и <code>if</code>	258
Другие вложенные операторы <code>if</code>	259
Давайте будем логичными	263
Альтернативное представление: заголовочный файл <code>iso646.h</code>	264
Приоритеты операций	265
Порядок вычисления выражений	265

10 Содержание

Диапазон значений	266
Программа подсчета слов	267
Условная операция ? :	270
Вспомогательные средства для циклов: <code>continue</code> и <code>break</code>	272
Оператор <code>continue</code>	272
Оператор <code>break</code>	275
Выбор из множества вариантов: операторы <code>switch</code> и <code>break</code>	277
Использование оператора <code>switch</code>	278
Чтение только первого символа строки	280
Множество меток	280
Операторы <code>switch</code> и <code>if else</code>	283
Оператор <code>goto</code>	283
Избегайте <code>goto</code>	283
Ключевые понятия	286
Резюме	287
Вопросы для самоконтроля	288
Упражнения по программированию	290
Глава 8. Символьный ввод-вывод и проверка достоверности ввода	293
Односимвольный ввод-вывод: <code>getchar()</code> и <code>putchar()</code>	294
Буферы	295
Завершение клавиатурного ввода	297
Файлы, потоки и ввод данных с клавиатуры	297
Конец файла	298
Перенаправление и файлы	301
Перенаправление в Unix, Linux и командной строке Windows	302
Создание дружественного пользовательского интерфейса	306
Работа с буферизированным вводом	306
Смешивание числового и символьного ввода	308
Проверка допустимости ввода	310
Анализ программы	315
Поток ввода и числа	315
Просмотр меню	316
Задачи	316
На пути к более гладкому выполнению	317
Смешивание символьного и числового ввода	319
Ключевые понятия	321
Резюме	322
Вопросы для самоконтроля	322
Упражнения по программированию	323
Глава 9. Функции	325
Обзор функций	326
Создание и использование простой функции	327
Анализ программы	328
Аргументы функции	330
Определение функции с аргументами: формальные параметры	331
Создание прототипа функции с аргументами	332
Вызов функции с аргументами: фактические аргументы	333

Представление в виде черного ящика	334
Возврат значения из функции с помощью <code>return</code>	334
Типы функций	337
Создание прототипов функций в ANSI C	338
Суть проблемы	338
Решение стандарта ANSI C	339
Отсутствие аргументов и неопределенные аргументы	340
Преимущество прототипов	341
Рекурсия	341
Рекурсия в действии	342
Основы рекурсии	343
Хвостовая рекурсия	344
Рекурсия и изменение порядка на противоположный	346
Преимущества и недостатки рекурсии	348
Компиляция программ, состоящих из двух и более файлов исходного кода	349
Unix	349
Linux	349
Компиляторы командной строки DOS	350
Компиляторы интегрированных сред разработки в Windows и Apple	350
Использование заголовочных файлов	350
Выяснение адресов: операция <code>&</code>	353
Изменение переменных в вызывающей функции	355
Указатели: первое знакомство	357
Операция разыменования: <code>*</code>	357
Объявление указателей	358
Использование указателей для обмена данными между функциями	359
Ключевые понятия	363
Резюме	363
Вопросы для самоконтроля	364
Упражнения по программированию	365
Глава 10. Массивы и указатели	367
Массивы	368
Инициализация	368
Назначенные инициализаторы (C99)	372
Присваивание значений элементам массива	373
Границы массива	374
Указание размера массива	376
Многомерные массивы	377
Инициализация двумерного массива	379
Большее количество измерений	380
Указатели и массивы	381
Функции, массивы и указатели	384
Использование параметров типа указателей	386
Комментарии: указатели и массивы	388
Операции с указателями	389
Защита содержимого массива	393
Использование <code>const</code> с формальными параметрами	394
Дополнительные сведения о ключевом слове <code>const</code>	395

12 Содержание

Указатели и многомерные массивы	397
Указатели на многомерные массивы	400
Совместимость указателей	401
Функции и многомерные массивы	403
Массивы переменной длины	406
Составные литералы	410
Ключевые понятия	412
Резюме	412
Вопросы для самоконтроля	414
Упражнения по программированию	416
Глава 11. Символьные строки и строковые функции	419
Введение в строки и строковый ввод-вывод	420
Определение строк в программе	421
Указатели и строки	429
Ввод строк	430
Создание пространства под строку	430
Неудачливая функция <code>gets()</code>	430
Альтернативы функции <code>gets()</code>	432
Функция <code>scanf()</code>	438
Вывод строк	440
Функция <code>puts()</code>	440
Функция <code>fputs()</code>	441
Функция <code>printf()</code>	442
Возможность самостоятельного создания функций	442
Строковые функции	445
Функция <code>strlen()</code>	445
Функция <code>strcat()</code>	446
Функция <code>strncat()</code>	447
Функция <code>strcmp()</code>	449
Функции <code>strcpy()</code> и <code>strncpy()</code>	454
Функция <code>sprintf()</code>	459
Другие строковые функции	460
Пример обработки строк: сортировка строк	462
Сортировка указателей вместо строк	464
Алгоритм сортировки выбором	465
Символьные функции <code>ctype.h</code> и строки	465
Аргументы командной строки	467
Аргументы командной строки в интегрированных средах	469
Аргументы командной строки в Macintosh	469
Преобразования строк в числа	470
Ключевые понятия	473
Резюме	473
Вопросы для самоконтроля	474
Упражнения по программированию	477
Глава 12. Классы хранения, связывание и управление памятью	479
Классы хранения	480
Область видимости	481

Связывание	483
Продолжительность хранения	484
Автоматические переменные	486
Регистровые переменные	490
Статические переменные с областью видимости в пределах блока	491
Статические переменные с внешним связыванием	492
Статические переменные с внутренним связыванием	496
Множество файлов	497
Спецификаторы классов хранения	498
Классы хранения и функции	501
Выбор класса хранения	501
Функция генерации случайных чисел и статическая переменная	502
Игра в кости	505
Выделенная память: <code>malloc()</code> и <code>free()</code>	509
Важность функции <code>free()</code>	513
Функция <code>calloc()</code>	514
Динамическое распределение памяти и массивы переменной длины	514
Классы хранения и динамическое распределение памяти	515
Квалификаторы типов ANSI C	517
Квалификатор типа <code>const</code>	517
Квалификатор типа <code>volatile</code>	519
Квалификатор типа <code>restrict</code>	520
Квалификатор типа <code>_Atomic (C11)</code>	521
Новые места для старых ключевых слов	522
Ключевые понятия	523
Резюме	523
Вопросы для самоконтроля	525
Упражнения по программированию	526
Глава 13. Файловый ввод-вывод	531
Взаимодействие с файлами	532
Понятие файла	532
Текстовый режим и двоичный режим	532
Уровни ввода-вывода	534
Стандартные файлы	534
Стандартный ввод-вывод	535
Проверка наличия аргумента командной строки	536
Функция <code>fopen()</code>	537
Функции <code>getc()</code> и <code>putc()</code>	538
Конец файла	538
Функция <code>fclose()</code>	540
Указатели на стандартные файлы	540
Бесхитростная программа уплотнения файла	540
Файловый ввод-вывод: <code>fprintf()</code> , <code>fscanf()</code> , <code>fgets()</code> и <code>fputs()</code>	542
Функции <code>fprintf()</code> и <code>fscanf()</code>	542
Функции <code>fgets()</code> и <code>fputs()</code>	544
Произвольный доступ: <code>fseek()</code> и <code>ftell()</code>	544
Работа функций <code>fseek()</code> и <code>ftell()</code>	545
Сравнение двоичного и текстового режимов	547

14 Содержание

Переносимость	547
Функции <code>fgetpos()</code> и <code>fsetpos()</code>	548
“За кулисами” стандартного ввода-вывода	548
Другие стандартные функции ввода-вывода	549
Функция <code>int ungetc(int c, FILE *fp)</code>	549
Функция <code>int fflush()</code>	550
Функция <code>int setvbuf()</code>	550
Двоичный ввод-вывод: <code>fread()</code> и <code>fwrite()</code>	551
Функция <code>size_t fwrite()</code>	552
Функция <code>size_t fread()</code>	553
Функции <code>int feof(FILE *fp)</code> и <code>int ferror(FILE *fp)</code>	553
Пример использования <code>fread()</code> и <code>fwrite()</code>	553
Произвольный доступ с двоичным вводом-выводом	556
Ключевые понятия	558
Резюме	558
Вопросы для самоконтроля	559
Упражнения по программированию	561
Глава 14. Структуры и другие формы данных	565
Учебная задача: создание каталога книг	566
Объявление структуры	567
Определение переменной типа структуры	568
Инициализация структуры	570
Доступ к членам структуры	570
Инициализаторы для структур	571
Массивы структур	571
Объявление массива структур	574
Идентификация членов в массиве структур	574
Анализ программы	575
Вложенные структуры	576
Указатели на структуры	577
Объявление и инициализация указателя на структуру	579
Доступ к членам по указателю	579
Сообщение функциям о структурах	580
Передача членов структуры	580
Использование адреса структуры	581
Передача структуры в качестве аргумента	582
Дополнительные возможности структур	583
Символьные массивы или указатели на <code>char</code> в структурах	587
Структура, указатели и <code>malloc()</code>	588
Составные литералы и структуры (C99)	591
Члены с типами гибких массивов (C99)	592
Анонимные структуры (C11)	594
Функции, использующие массив структур	595
Сохранение содержимого структур в файле	596
Пример сохранения структуры	597
Анализ программы	600
Структуры: что дальше?	601
Объединения: краткое знакомство	602

Использование объединений	603
Анонимные объединения (C11)	604
Перечислимые типы	605
Константы enum	606
Стандартные значения	606
Присвоенные значения	606
Использование enum	606
Совместно используемые пространства имен	608
Средство typedef: краткое знакомство	609
Причудливые объявления	611
Функции и указатели	612
Ключевые понятия	619
Резюме	620
Вопросы для самоконтроля	620
Упражнения по программированию	623
Глава 15. Манипулирование битами	627
Двоичные числа, биты и байты	628
Двоичные целые числа	629
Целые числа со знаком	629
Двоичные числа с плавающей запятой	630
Другие основания систем счисления	631
Восьмеричная система счисления	631
Шестнадцатеричная система счисления	631
Побитовые операции	632
Побитовые логические операции	633
Случай применения: маски	634
Случай применения: включение (установка) битов	635
Случай применения: выключение (очистка) битов	636
Случай применения: переключение битов	636
Случай применения: проверка значения бита	637
Побитовые операции сдвига	637
Пример программы	639
Еще один пример	640
Битовые поля	642
Пример с битовыми полями	644
Битовые поля и побитовые операции	647
Средства выравнивания (C11)	653
Ключевые понятия	655
Резюме	655
Вопросы для самоконтроля	656
Упражнения по программированию	658
Глава 16. Препроцессор и библиотека C	661
Первые шаги в трансляции программы	662
Символические константы: #define	663
Лексемы	666
Переопределение констант	667
Использование аргументов в директиве #define	667

16 Содержание

Создание строк из аргументов макроса: операция #	670
Средство слияния препроцессора: операция ##	671
Макросы с переменным числом аргументов: . . . и __VA_ARGS__	672
Выбор между макросом и функцией	673
Включение файлов: директива #include	674
Пример заголовочного файла	675
Случаи применения заголовочных файлов	677
Другие директивы	678
Директива #undef	678
Определение с точки зрения препроцессора	678
Условная компиляция	679
Предопределенные макросы	684
Директивы #line и #error	685
Директива #pragma	685
Обобщенный выбор (C11)	686
Встраиваемые функции (C99)	688
Функции _Noreturn (C11)	690
Библиотека C	690
Получение доступа к библиотеке C	691
Использование описаний библиотеки	692
Библиотека математических функций	693
Немного тригонометрии	694
Варианты типов	695
Библиотека tgmath.h (C99)	697
Библиотека утилит общего назначения	698
Функции exit() и atexit()	698
Функция qsort()	700
Библиотека утверждений	704
Использование assert()	704
_Static_assert (C11)	706
Функции memcpy() и memmove() из библиотеки string.h	707
Переменное число аргументов: файл stdarg.h	709
Ключевые понятия	711
Резюме	711
Вопросы для самоконтроля	712
Упражнения по программированию	713
Глава 17. Расширенное представление данных	717
Исследование представления данных	719
От массива к связанному списку	721
Использование связанного списка	725
Дополнительные соображения	728
Абстрактные типы данных	729
Получение абстракции	730
Построение интерфейса	731
Использование интерфейса	735
Реализация интерфейса	737
Создание очереди с помощью ADT	744
Определение абстрактного типа данных для представления очереди	744

Определение интерфейса	744
Реализация представления данных интерфейса	745
Тестирование очереди	753
Моделирование реальной очереди	755
Сравнение связного списка и массива	761
Двоичные деревья поиска	764
Создание абстрактного типа данных для двоичного дерева	765
Интерфейс двоичного дерева поиска	766
Реализация двоичного дерева	768
Тестирование пакета для древовидного представления	782
Соображения по поводу дерева	786
Другие направления	787
Ключевые понятия	788
Резюме	788
Вопросы для самоконтроля	788
Упражнения по программированию	789
Приложение А. Ответы на вопросы для самоконтроля	791
Ответы на вопросы для самоконтроля из главы 1	792
Ответы на вопросы для самоконтроля из главы 2	792
Ответы на вопросы для самоконтроля из главы 3	794
Ответы на вопросы для самоконтроля из главы 4	796
Ответы на вопросы для самоконтроля из главы 5	798
Ответы на вопросы для самоконтроля из главы 6	801
Ответы на вопросы для самоконтроля из главы 7	804
Ответы на вопросы для самоконтроля из главы 8	807
Ответы на вопросы для самоконтроля из главы 9	808
Ответы на вопросы для самоконтроля из главы 10	810
Ответы на вопросы для самоконтроля из главы 11	812
Ответы на вопросы для самоконтроля из главы 12	816
Ответы на вопросы для самоконтроля из главы 13	817
Ответы на вопросы для самоконтроля из главы 14	820
Ответы на вопросы для самоконтроля из главы 15	823
Ответы на вопросы для самоконтроля из главы 16	824
Ответы на вопросы для самоконтроля из главы 17	826
Приложение Б. Справочные материалы	829
Раздел I. Дополнительные источники информации	830
Онлайновые ресурсы	830
Книги по языку C	831
Книги по программированию	831
Справочные руководства	832
Книги по C++	832
Раздел II. Операции в языке C	832
Арифметические операции	833
Операции отношений	834
Операции присваивания	834
Логические операции	835
Условная операция	835

18 Содержание

Операции, связанные с указателями	836
Операции со знаком	836
Операции структур и объединений	836
Побитовые операции	837
Прочие операции	838
Раздел III. Базовые типы и классы хранения	838
Сводка: базовые типы данных	838
Сводка: объявление простой переменной	840
Сводка: квалификаторы	842
Раздел IV. Выражения, операторы и поток управления программой	843
Сводка: выражения и операторы	843
Сводка: оператор while	844
Сводка: оператор for	844
Сводка: оператор do while	845
Сводка: использование операторов if для реализации выбора	845
Сводка: множественный выбор с помощью switch	846
Сводка: переходы в программе	847
Раздел V. Стандартная библиотека ANSI C с дополнениями C99 и C11	848
Диагностика: assert.h	848
Комплексные числа: complex.h (C99)	849
Обработка символов: ctype.h	851
Сообщение об ошибках: errno.h	851
Среда плавающей запятой: fenv.h (C99)	852
Характеристики среды плавающей запятой: float.h	854
Преобразование формата целочисленных типов: inttypes.h (C99)	856
Альтернативное написание: iso646.h	857
Локализация: locale.h	857
Математическая библиотека: math.h	860
Нелокальные переходы: setjmp.h	864
Обработка сигналов: signal.h	865
Выравнивание: stdalign.h (C11)	866
Переменное количество аргументов: stdarg.h	866
Поддержка атомарности: stdatomic.h (C11)	867
Поддержка булевских значений: stdbool.h (C99)	867
Общие определения: stddef.h	868
Целочисленные типы: stdint.h	868
Стандартная библиотека ввода-вывода: stdio.h	871
Общие утилиты: stdlib.h	874
_Noreturn: stdnoreturn.h	879
Обработка строк: string.h	879
Математические функции для обобщенных типов: tgmath.h (C99)	882
Потоки: threads.h (C11)	883
Дата и время: time.h	883
Утилиты Unicode: uchar.h (C11)	887
Утилиты для работы с многобайтными и широкими символами: wchar.h (C99)	887
Утилиты классификации и отображения широких символов: wctype.h (C99)	893
Раздел VI. Расширенные целочисленные типы	895
Типы с точной шириной	895
Типы с минимальной шириной	896

Самые быстрые типы с минимальной шириной	896
Типы максимальной ширины	897
Целые, которые могут хранить указатели	897
Расширенные целочисленные константы	898
Раздел VII. Расширенная поддержка символов	898
Триграфы	898
Диграфы	899
Альтернативное написание: <code>iso646.h</code>	899
Многобайтные символы	899
Универсальные имена символов (UCN)	900
Широкие символы	901
Широкие и многобайтные символы	903
Раздел VIII. Расширенные вычислительные средства C99/C11	903
Стандарт плавающей запятой IEC	903
Заголовочный файл <code>fenv.h</code>	907
Прагма <code>STDC_FP_CONTRACT</code>	908
Дополнения библиотеки <code>math.h</code>	908
Поддержка комплексных чисел	909
Раздел IX. Отличия между C и C++	911
Прототипы функций	911
Константы <code>char</code>	912
Модификатор <code>const</code>	913
Структуры и объединения	914
Перечисления	914
Указатель на <code>void</code>	915
Булевские типы	915
Альтернативное написание	915
Поддержка широких символов	915
Комплексные типы	915
Встраиваемые функции	916
Средства C99/C11, которых нет в C++11	916
Приложение В. Набор символов ASCII	917
Предметный указатель	922