

C语言综合习题集 (英文版)

C YUYAN ZONGHE XITISI

赵卫东 周晶 编著



国防工业出版社

National Defense Industry Press

C 语言综合习题集

(英文版)

赵卫东 周 晶 编著

国防工业出版社

• 北京 •

内 容 简 介

本书是专为C语言程序设计课程的教学,尤其是双语教学而编写的。编写本书的目的:一是为降低直接使用国外原版教材时在C语言练习方面的难度梯度;二是为了在双语教学的同时亦能满足国内各类C语言等级考试的需求。题目根据中国学生的学习难点,侧重基本概念掌握和基本技能的训练,内容涵盖C程序设计语言的各个方面,程度由浅入深,注重学生编程能力的培养,适合C语言初学者的学习需要。所有的习题都经过了精心的编写、翻译和验算,相当数量的习题在作者的C语言双语教学过程中得以应用。题目类型有单选题、填空题、问答题、对错判断题、改错题和编程题等。其中,单选题是按国内各种等级考试的“四选一”形式给出;填空题则根据题目的不同有1个~3个空需要填充;改错题一般有两三个错误,既有语法错误,也有逻辑错误。编程题的设计考虑了我国学生在学习C语言时的思维习惯和知识范围,从他们相对熟悉的内容入手来调动学生的编程兴趣。

本书全部用英文编写,以适应双语教学的需要。此外,以16-bit PC机和Turbo C 2.0为平台给出每一道题目的参考答案,为教师的教学以及学生自学提供方便。

图书在版编目(CIP)数据

C语言综合习题集:英文版/赵卫东,周晶编著. —北京:
国防工业出版社, 2009.9
ISBN 978-7-118-06516-9

I. C... II. ①赵...②周... III. C语言-程序设计-双语
教学-高等学校-习题-英文 IV. TP312-44

中国版本图书馆CIP数据核字(2009)第142887号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路23号 邮政编码100048)

北京市李史山胶印厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 17 1/2 字数 460 千字

2009年9月第1版第1次印刷 印数 1—4000册 定价 33.00元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

前 言

随着计算机技术的飞速发展, 计算机程序设计课程在高等院校, 特别是理工科高等院校成为了必修科目。同时, 在高等院校教学中引进原版教材, 实行双语教学, 尤其是在计算机相关课程中实施双语教学, 也是教育部一直积极倡导的教改举措。笔者从 2002 年开始在各个理工科专业学生中开展“程序设计基础(C 语言)”课程的双语教学, 至今已经 7 年有余。我们选用国内影印出版的《The C Programming Language》(K&R)第二版。这本书是 C 语言的设计者编写的, 是 C 语言教材的经典之作, 被称为“C 语言圣经”, 是 C 语言双语教学的首选教科书。但美中不足的是, 书中的习题数量很少, 种类单一(只有编程题), 难度偏高, 许多题目让没有编程经验的学生无从下手。这给 C 语言双语教学带来了很大的障碍。原版 C 语言教材题目偏少、偏难、与国内学生知识面相差较大等, 是编写本书的第一动因。

此外, 国内学生学习编程语言, 还在很大程度上要应对各类的计算机等级考试, 可国内出版的 C 语言英文版教材鲜有针对国内考试的合适的练习题目, 这又给 C 语言双语教学带来了一大障碍, 因为这样的双语教学难以同时应对国内的各种等级考试。笔者所编写的本书, 内容基本上是按国内大学 C 语言的教学顺序安排的, 题目类型也与国内各种 C 语言考试相类似, 难度由浅入深。这给学生一个循序渐进的练习机会, 降低了学习 C 语言的难度梯度, 可以说是同时兼顾了 C 语言双语教学和 C 程序等级考试的共同需要。

从目前来看, 国内尚没有类似的 C 语言英文版习题集。本书的出版将在某种程度上填补这一空白, 相信会对广大学习 C 语言的学生和教师, 尤其是进行双语教学的学生和教师提供有益的帮助。

本书共包括 12 章, 内容丰富, 基本上涵盖了国内 C 语言教学的所有内容。有单选题、填空题、对错判断题、问答题、改错题和编程题等多种题型。

本书的读者为所有 C 语言学习者, 尤其是接受 C 语言双语教学的学生。书中给出了每一道习题的参考答案, 方便教师的教学和学生自学。应该说许多习题, 尤其是编程题的答案并不是唯一的, 本书鼓励学生做出更有创意、更为简洁的习题答案。本书的习题解答均经过 Turbo C 2.0 系统的测试, 有些习题在其他系统环境下(如 VC++编译环境)可能有不同的答案, 请读者注意。

本书主要由赵卫东编写、调试和翻译, 周晶在部分习题的搜集整理和测试上做了大量有意义的工作。

本书是笔者 C 语言教学和双语教学中的一个成果, 也是进一步推广双语教学的一次有积极意义的尝试, 缺点错误在所难免, 敬请各位读者批评指正。

编著者
2009.8

目 录

Chapter 1	Fundamentals of C	1
Chapter 2	Data Types, Operators and Expressions	5
Chapter 3	The Basic Structure of C Programs	15
Chapter 4	Logical Operations, Decision Making and Branching	29
Chapter 5	Loops	46
Chapter 6	Arrays	70
Chapter 7	Functions	97
Chapter 8	Preprocessor Directives	128
Chapter 9	Pointers	136
Chapter 10	Structures and Unions	163
Chapter 11	Bitwise Operations	188
Chapter 12	File Processing	197
Solutions to All the Problems		208

Chapter 1 Fundamentals of C

1.1 Multiple Choice Questions

- 【1.1.1】** The C source code is consisted of ().
A) functions
B) functions and procedures
C) source program code
D) subroutines
- 【1.1.2】** There may be any number of functions with different names in a C program. But there must be one and only one () in a C program.
A) function
B) main function
C) include
D) procedure
- 【1.1.3】** In C, the name of the only component that is required in every C program is ().
A) stdio
B) include
C) main
D) function
- 【1.1.4】** The correct description in the following is ().
A) the main function has to be placed in the front of a C program
B) the main function must be located in the end of a C program
C) the main function can be put between any functions in a C program, but the program start its running at the beginning of the program
D) the main function can be put between any functions in a C program, but the program start its running at the main function
- 【1.1.5】** In C, the mark () denotes the termination of statements and data definitions.
A) }
B) .
C) ;
D) ,
- 【1.1.6】** In computer science, “programming” refers to ().
A) the main part of computer software
B) writing a number of instructions to form a program
C) a set of instructions that tell a computer how to carry out a task
D) hardware being driven by software
- 【1.1.7】** Computer hardware can run () directly.
A) programs written in C
B) programs written in assembly language
C) programs written in machine language
D) programs written in high level language
- 【1.1.8】** The following () is not a high level computer language.
A) Fortran
B) Pascal
C) English
D) C++
- 【1.1.9】** The C language is so named because ().
A) “C” is so simple
B) “C” is the abbreviation of word “Computer”
C) C is the third version of the language developed in Bell Lab

D) C's predecessor is called B

【1.1.10】 A portable language refers to ().

- A) the language can be moved from one place to another
- B) the language can be used in any computer
- C) the language can be used to write any kind of programs
- D) the language can be used with other languages

【1.1.11】 The C language is so popular because ().

- A) C is powerful and with many keywords
- B) C is a superset of C++
- C) a wide variety of C compilers and helpful accessories are available
- D) C is based on Java

【1.1.12】 A function consists mainly of three parts: ().

- A) function name, function list and function body
- B) #include, <stdio.h> and main function
- C) main, () and { }
- D) function name, parameter list and function body

【1.1.13】 A header file is such a file which extension name is ().

- A) .c
- B) .doc
- C) .h
- D) .xls

【1.1.14】 A function body is the main part of the function, it begins with ().

- A) {
- B) }
- C) (
- D))

【1.1.15】 A C program starts its execution from ().

- A) the **main** function in the program and ends its execution in the **main** function
- B) the first function of the program and ends its execution in the last function of the program
- C) the **main** function in the program and ends its execution in the last function of the program
- D) the first function of the program and ends its execution in the **main** function

【1.1.16】 The correct description in the following is ().

- A) the **main** function ought to be placed in the front of the program
- B) there must be one statement in a line
- C) there are not input and output statements in C
- D) the spelling mistakes in comments could be detected while compiling

【1.1.17】 The incorrect description in the following is ().

- A) a C source code is consisted of one or several functions
- B) a C source code must include one main function
- C) a C program is consisted of functions
- D) the comment in C can only be placed after a statement

【1.1.18】 Comments are used to ().

- A) explain briefly the functions of a program or some of the statements
- B) make the program look better
- C) give the program some new functions
- D) let the program work well

【1.1.19】 Any characters between () are called comment and are ignored by the compiler.

A) */ and /*

B) */ and */

C) /* and */

D) /* and /*

【1.1.20】 Comments may appear anywhere where () can appear.

A) any character

B) a letter, a number character or a punctuation

C) a keyword of C language

D) a blank, tab or newline

1.2 Filling in Blanks

【1.2.1】 A C source code can be executed by computer after __ 【1】 __ and __ 【2】 __.

【1.2.2】 In C source code, a variable is a _____ assigned to a data storage location.

【1.2.3】 A C source code file's extension name is __ 【1】 __, the file's extension name after compiling is __ 【2】 __, the file's extension name after linking is __ 【3】 __.

【1.2.4】 A comment in C is enclosed by __ 【1】 __ and __ 【2】 __.

【1.2.5】 In C, most data inputting is implemented by __ 【1】 __, and most data outputting is implemented by __ 【2】 __.

1.3 Quizzes

【1.3.1】 Give three reasons why C is the best choice of programming language.

【1.3.2】 What does the compiler do?

【1.3.3】 What are the steps in the program development cycle?

【1.3.4】 What extension should you use for your C source files?

【1.3.5】 If you execute a program that you have compiled and it doesn't work as you expected, what should you do?

【1.3.6】 What is machine language?

【1.3.7】 What does the linker do?

【1.3.8】 What is the term for a group of one or more C statements enclosed in braces?

【1.3.9】 What is the one component that must be present in every C program?

【1.3.10】 How do you add program comments, and why are they used?

【1.3.11】 What is a function?

【1.3.12】 C offers two types of functions. What are they, and how are they different?

【1.3.13】 What is the **#include** directive used for?

【1.3.14】 Can comments be nested?

【1.3.15】 Can comments be longer than one line?

【1.3.16】 What is another name for an include file?

【1.3.17】 What is an include file?

【1.3.18】 What does portability mean in the context of programming?

【1.3.19】 Explain the difference between a source code file, object code file, and executable file.

【1.3.20】 What does a compiler do?

【1.3.21】 What does a linker do?

【1.3.22】 What is the difference between a statement and a block?

【1.3.23】 How can I find out what library functions are available?

1.4 Programming Exercises

【1.4.1】 Input the following source code in your computer. Then, compile, link and run it, to see what will happen.

```
#include <stdio.h>
main()
{    printf("Hello, World!\n");
    return 0; }
```

【1.4.2】 Experiment with leaving out parts of the above program, to see what error messages you get.

【1.4.3】 Experiment to find what happens when *printf*'s argument string contains *\t*, *\b*, *\r* and *\w*.

【1.4.4】 Input and run the following source program, to see what happens.

```
#include <stdio.h>
main()
{
    int fahr, celsius;
    int lower, upper, step;
    lower = 0;
    upper = 300;
    step = 20;
    fahr = lower;
    while (fahr <= upper) {
        celsius = 5 * (fahr-32) / 9;
        printf("%d\t%d\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

Chapter 2 Data Types, Operators and Expressions

2.1 Multiple Choice Questions

- 【2.1.1】 The smallest executable entity within C programming language is ().
A) a function B) a procedure C) a statement D) an operator
- 【2.1.2】 In programming language, we use () to store values used during the computation.
A) operators B) functions C) escape sequences D) variables
- 【2.1.3】 C uses a pair of () to mark up a character string (or string constant).
A) parentheses B) double quotes C) single quotes D) braces
- 【2.1.4】 The first argument of *printf* function when calling it must be ().
A) a character string B) a variable C) a constant D) an operator
- 【2.1.5】 An escape sequence is used to denote ().
A) a newline B) a hard-to-type or invisible character
C) a printable character D) a character other than English letters
- 【2.1.6】 An escape sequence consists of a mark () followed by a single character.
A) “\” B) “-” C) “\” D) “/”
- 【2.1.7】 In C, the “address of” operator is ().
A) # B) * C) @ D) &
- 【2.1.8】 Choose a correct description in the following: ().
A) an address operator has to be put in front of a variable parameter in *printf* function call
B) an arithmetic operator has to be put in front of a variable parameter in *printf* function call
C) an address operator has to be put in front of a variable parameter in *scanf* function call
D) a logical operator has to be put in front of a variable parameter in *scanf* function call
- 【2.1.9】 In C, ().
A) some variables must be declared before they are used
B) all variables doesn't need to be declared before they are used
C) all variables will be declared automatically when they are used first time
D) all variables must be declared before they are used
- 【2.1.10】 In C, a declaration section ().
A) is usually at the end of a function
B) is usually at the beginning of a function before any executable statements
C) is at liberty wherever you put it
D) must be put after the statement that you use the variable
- 【2.1.11】 A variable declaration ().
A) announces the names of variables and the properties of variables
B) tells the compiler how to compile the program

- C) announces functions to be used
- D) tells computer what kind of numbers to be used

【2.1.12】 The following () is a legal variable declaration.

- A) INT A, B, C; B) float x, y, z; C) int a=345; D) long int b

【2.1.13】 I want to declare a variable to hold number 1.234, what variable type should I use? ()

- A) long B) int C) float D) char

【2.1.14】 Which of the following data type occupies 4 bytes in memory?()

- A) int B) char C) short D) long

【2.1.15】 Which of the following is not a legal data type? ()

- A) long B) short C) double D) single

【2.1.16】 Which of the following description is not true?()

- A) In C, all names(function name, variable name and so on) must be an identifier.
- B) In C, function names must be an identifier.
- C) In C, variable names must be an identifier.
- D) In C, an identifier is merely a C's keyword.

【2.1.17】 Which of the following is not a legal identifier? ()

- A) _a_34 B) 3w C) _345 D) ABCD

【2.1.18】 Which of the following description is not true? ()

- A) An identifier is a sequence of letters, digits and underscores. But the first character must be a letter.
- B) An identifier is a sequence of letters, digits and underscores. But the last character must be a letter.
- C) An identifier is a sequence of letters, digits and underscores. But underscore can not be the first character.
- D) An identifier is a sequence of letters, digits and underscores. But digit must be the first character.

【2.1.19】 In C, if I want to input a floating point number to variable 'celsius', which of the following statements should I use? ()

- A) scanf("%d",&celsius); B) scanf("%f",&celsius);
- C) scanf("%f",celsius); D) printf("%f",celsius);

【2.1.20】 The correct description in the following is ().

- A) in C, both 'define' and 'if' can be defined as identifiers
- B) in C, 'if' can be defined as an identifier, but 'define' can not be used as an identifier
- C) in C, 'define' can be defined as an identifier, but 'if' can not be used as an identifier
- D) in C, neither 'define' and 'if' can be used as an identifier

【2.1.21】 The correct description in the following is ().

- A) in C, there can be more than one function with the same name in a program
- B) in C, a statement must be written in one line
- C) comment can appear in any position where a space, a tab or a new line character can appear
- D) the smallest C program can be written without main function

【2.1.22】 The correct assignment statement in the following is ().

- A) $b + c = a + 3$; B) $3 = a + b$; C) $a = b + c$; D) $a + 3 = c$;

【2.1.23】 Suppose that a and b are both of type *int*, and let $a=3$, $b=7$, the value of expression b/a is ().

- A) 2.3333... B) 2.3 C) 2 D) 3

【2.1.24】 The value of expression " $c = 3 + 5 * 2$ " is ().

- A) $c = 13$ B) $c = 30$ C) 13 D) 30

【2.1.25】 What does expression " $c = \text{getchar}()$ " evaluate when data 5 is keyed in? ().

- A) 5 B) 53 C) 69 D) 101

【2.1.26】 After the execution of $x = 10$; $y = x++$; , the values of x and y are ().

- A) 10 and 11 B) 11 and 10 C) 11 and 11 D) 10 and 10

【2.1.27】 The conversion specification *%ld* tells *printf* that ().

- A) the corresponding argument is a long integer
B) the corresponding argument is a short integer
C) the corresponding argument is a single precision floating point number
D) the corresponding argument is a double precision floating point number

【2.1.28】 Which of the following is correct?()

- A) '\n' is an escape sequence. B) '\n' is a character constant.
C) 'A' is symbolic constant. D) A is a character constant.

【2.1.29】 After the execution of the following program segment, the value of $c3$ is ().

int c1=1,c2=2,c3;

*c3=1.0/c2*c1;*

- A) 0 B) 0.5 C) 1 D) 2

【2.1.30】 Which of the following expression is the equivalent to $k=n++$?().

- A) $k=n$, $n=n+1$ B) $n=n+1$, $k=n$ C) $k=++n$ D) $k+=n+1$

【2.1.31】 The basic data objects manipulated in a program are ().

- A) variables and functions B) integers and floating numbers
C) expressions and statements D) variables and constants

【2.1.32】 The expressions ().

- A) combine variables and constants with one or several operators to produce new values
B) combine parameters and constants with one or several operators to produce new values
C) combine variables and arguments with one or several operators to produce new values
D) combine arguments and parameters with one or several operators to produce new values

【2.1.33】 There are () types of constants in C language.

- A) 1 B) 2 C) 3 D) 4

【2.1.34】 what is a literal constant? ()

- A) A literal constant is a value that is typed directly into the source code wherever it is needed.
B) A literal constant is a name that is directly written in the source code wherever it is needed.
C) A literal constant is an argument that is typed directly into the source code wherever it is needed.
D) A literal constant is a value that is calculated from an expression.

【2.1.35】 Which of the following description is not correct? ().

A) $3 + 4.5$ is a literal constant

B) $0.85e-4$ is a literal constant

C) 123456789L is a literal constant

D) 0XAUL is a literal constant

【2.1.36】 The default type of a floating point constant is ().

A) float

B) double

C) long double

D) long float

【2.1.37】 There are () notations for an integer constant.

A) 1

B) 2

C) 3

D) 4

【2.1.38】 There are () notations for a floating point constant.

A) 1

B) 2

C) 3

D) 4

【2.1.39】 With a scientific notation floating point constant " $0.85e-4$ ", which of the following is an identical one? ().

A) " $0.085e-5$ "

B) " $8.5e-5$ "

C) " $85e-8$ "

D) " $0.852E-4$ "

【2.1.40】 Which of the following is a decimal integer constant? ().

A) 12.34

B) 0110110

C) 01234

D) 1234

【2.1.41】 Which of the following is a legal octal integer constant? ().

A) 12.35

B) 0110110

C) 0129

D) 1234

【2.1.42】 Which of the following is a legal hexadecimal integer constant? ().

A) 0XYZ

B) 0XAUL

C) X123

D) 129H

【2.1.43】 Which of the following is a character constant? ().

A) '007'

B) '\A'

C) '\XF'

D) "A"

【2.1.44】 The value of '\077' is ().

A) 77

B) 0x77

C) 129

D) 63

【2.1.45】 The value of '\x70' is ().

A) 70

B) 56

C) 070

D) 112

【2.1.46】 The following () is not a string constant.

A) "He is tall!"

B) ""

C) "abcdefg"

D) "hello," "world"

【2.1.47】 Given the declaration statements as follows:

const int a, b, c; int i, j;

which of the following is a constant expression? ().

A) $i++$

B) $20+j$

C) $a+b*c+4$

D) $abc+5$

【2.1.48】 Which of the following is not a string constant? ().

A) "This is an apple."

B) ""

C) "That is not" "an apple"

D) 'This is a banana.'

【2.1.49】 Which symbol in the following is either a binary or unary operator? ().

A) %

B) +

C) >=

D) &&

【2.1.50】 Which of the following is correct? ().

A) The precedence of arithmetic operators is higher than that of relational operators.

B) The precedence of logical operators is higher than that of arithmetic operators.

C) The precedence of logical operators is higher than that of relational operators.

D) The precedence of relational operators is higher than that of arithmetic operators.

【2.1.51】 What is the data type of expression $3 + 4.5$? ().

A) int

B) float

C) long

D) double

【2.1.52】 If variable *c* holds a digit character, which of the following expression is evaluated to the value that the digit character stands for? ()

- A) 'c'-'0' B) c-'A' C) 'c'-'a' D) c-'0'

【2.1.53】 Conversions take place across assignments: longer integers are converted to shorter ones or to *chars* by ().

- A) dropping the excess low-order bits B) dropping the excess high-order bits
C) adding the excess high-order bits D) adding the excess low-order bits

【2.1.54】 Conversions take place across assignments: ().

- A) the value of the right side of the assignment operator is converted to the type of the left, which is the type of the result
B) the type of the left side of the assignment operator is converted to the type of the right, which is the type of the result
C) the value of the right side of the assignment operator is kept the type of its own, which is the type of the result
D) the value of the right side of the assignment operator is converted to the type of the left, which is not the type of the result

【2.1.55】 A cast is ().

- A) a variable type B) a constant type
C) something enclosed in parentheses D) an explicit type conversion operator

【2.1.56】 The expression *(float)5.3* means ().

- A) converting 5.3 from int to float
B) converting 5.3 from float to double
C) converting 5.3 from double to float
D) changing 5.3 by dropping its fractional part

【2.1.57】 Either side of a logical operator ().

- A) can only be 1 or 0 B) can only be 0 or a positive integer
C) can be a datum of type *int* or *char* D) can be a datum of any type

【2.1.58】 Assume all variables are of type *int*, what's the output of the following program segment? ()

```
sum = pad = 5; pAd = sum++, pAd++, ++pAd;  
printf("'%d\n'", pad);
```

- A) 7 B) 6 C) 5 D) 4

【2.1.59】 What's the output of the following: ()

```
printf("'%d, '", 'A' + '5' - '3');  
printf("'%c'", 'A' + '5' - '3');"
```

- A) 67, C B) B, C C) C, 67 D) C, c

【2.1.60】 Given the declaration:

```
int a = 7; float x=2.5, y=4.7;
```

the following expression's value is ().

```
x + a%3*(int)(x + y)%2/4
```

- A) 2.500000 B) 2.750000 C) 3.500000 D) 0.000000

2.2 Filling in Blanks

【2.2.1】In C language (assuming for PC or compatibles), a *char* uses ____【1】____ byte for storage, and an *int* uses ____【2】____ bytes for storage.

【2.2.2】In C language (take PC for example), a *float* uses ____【1】____ byte for storage, and a *double* uses ____【2】____ bytes for storage.

【2.2.3】Given the declaration as follows:

int m=5,y=2;

the value of *y* after the evaluation of *y+=y-=m*=y* is _____.

【2.2.4】Suppose that a variable of type *int* takes up two bytes in memory, so the range of values the variable can take on is _____.

【2.2.5】There are two kinds of variables of real type, they are ____【1】____ and ____【2】____.

【2.2.6】The basic data type in C includes float, double, ____【1】____, ____【2】____ and ____【3】____.

【2.2.7】If *s* is a variable of type *int*, the value of the following expression is _____.

s%2+(s+1)%2

【2.2.8】Assume that *x* and *a* are both variables of type *int*, the value of *x* after the execution of expression (1) is ____【1】____ and the value of *x* after the execution of expression (2) is ____【2】____.

(1) *x=(a=4,6*2)*

(2) *x=a=4,6*2*

【2.2.9】Assume *a*, *b*, and *c* are all of type *int*, after the evaluation of the following expression,

a=(b=4)+(c=2)

the *a*'s value is ____【1】____, *b*'s value is ____【2】____, *c*'s value is ____【3】____.

【2.2.10】Assume *a* is a variable of type *int* and initialized to 6, after the evaluation of the following expression,

*a+=a-=a*a*

the value of *a* is _____.

【2.2.11】Assume *a* is a variable of type *int*, after the evaluation of the following expression, the value of *a* is _____.

a=25/3%3

【2.2.12】Assume *x* and *n* are variables of type *int* and are both initialized to 5, after the evaluation of the following expression,

x+=n++

the value of *x* is ____【1】____, the value of *n* is ____【2】____.

【2.2.13】Given variable declarations: *int b=7; float a=2.5, c=4.7;*

the value of the following expression is _____.

a+(int)(b/3(int)(a+c)/2)%4*

【2.2.14】Given variable declarations as follows:

int a=2, b=3; float x=3.5, y=2.5;

the value of the following expression is _____.

(float)(a+b)/2+(int)x%(int)y

【2.2.15】Given the declaration:

`char c='\010';`

the number of characters stored in variable `c` is _____.

【2.2.16】 With the variable declarations as follows:

`int x=3, y=2; float a=2.5, b=3.5;`

the value of the following expression is _____.

`(x+y)%2+(int)a/(int)b`

【2.2.17】 Assume `x` and `n` are variables of type `int`, and `x` is initialized to 12, `n` is initialized to 5, after the evaluation of the following expression,

`x%=(n%=2)`

the value of `x` is ____ 【1】 ____, the value of `n` is ____ 【2】 ____.

【2.2.18】 Assume all variables are of type `int`, then the value of expression `(a=2,b=5,a++,b++,a+b)` is _____.

【2.2.19】 In C, an identifier consists of three kinds of characters, they are ____ 【1】 ____, ____ 【2】 ____ and ____ 【3】 ____.

【2.2.20】 Suppose letter `a`'s ASCII code is 97 in decimal, let `ch` be a variable of type `char`, the value of expression `ch='a'+'8'-'3'` is _____.

【2.2.21】 Assume `x` and `y` are variables of type `double`, and `x` is initialized to 3.0, `y` is initialized to 2.0, then the value of expression `pow(y,fabs(x))` is _____.

【2.2.22】 Given the declarations:

`int e=1, f=4, g=2; float m=10.5, n=4.0, k;`

the value of expression `k=(e+f)/g+sqrt((double)n)*1.2/g+m` is _____.

【2.2.23】 The data type of expression `8/4*(int)2.5/(int)(1.25*(3.7+2.3))` is _____.

【2.2.24】 The data type of expression `pow(2.8,sqrt((double)x))` is _____.

【2.2.25】 Given `m` is a three-digit integer and the digits are `a`, `b`, `c` from left to right respectively. Then, to be evaluated as an integer with digits sequence `bac`, the expression using `m` as a variable is _____.

【2.2.26】 Identify the data type (as used in declaration statements) and the `printf()` format specifier for each of the following constants:

	Constant	Type	Specifier		Constant	Type	Specifier
a.	12			e.	'\040'		
b.	0X3			f.	7.0		
c.	'C'			g.	6L		
d.	2.34E07			h.	6.0f		

【2.2.27】 Identify the data type (as used in declaration statements) and the `printf()` format specifier for each of the following constants (assume a 16-bit int):

	Constant	Type	Specifier		Constant	Type	Specifier
a.	012			e.	'\n'		
b.	2.9e05L			f.	20.0f		
c.	's'			g.	0x44		
d.	100000						

【2.2.28】 Suppose a program begins with these declarations:

```
int imate = 2;
long shot = 53456;
char grade = 'A';
float log = 2.71828;
```

Fill in the blanks with specifiers of proper type in the following *printf()* statements:

```
printf("The odds against the % 【1】 were % 【2】 to 1.\n", imate, shot);
printf("A score of % 【3】 is not an % 【4】 grade.\n", log, grade);
```

2.3 Quizzes

【2.3.1】 Which data type would you use for each of the following kinds of data?

- A. *the number of students of a class*
- B. *the cost of a movie on DVD*
- C. *the most common letter in this paper*
- D. *the number of times that the letter occurs in this book*

【2.3.2】 Why would you use a type *long* variable instead of type *int*?

【2.3.3】 What portable types might you use to get a 32-bit signed integer, and what would the rationale be for each choice?

【2.3.4】 Identify the type and meaning, if any, of each of the following constants:

- A. `'\b'`
- B. `1066`
- C. `99.44`
- D. `0XAA`
- E. `2.0e30`

【2.3.5】 Mr. Zhang has concocted an error-laden program. Help him find the mistakes.

```
1: include <stdio.h>
2: main
3: (
4:   float g; h;
5:   float tax, rate;
6:
7:   g = e21;
8:   tax = rate*g;
9: )
```

【2.3.6】 Suppose that *ch* is a type *char* variable. Show how to assign the carriage-return character to *ch* by using an escape sequence, a decimal value, an octal character constant, and a hexadecimal character constant. (Assume ASCII code values.)

【2.3.7】 Correct the following silly program.

```
1: void main(int) / this program is perfect /
2: {
```