# 问题求解和算法

**算法的特征：**

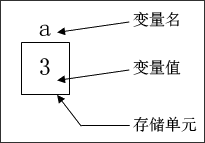
1. **有穷性**
2. **确切性**
3. **可行性**
4. **一个算法有零个或多个输入**
5. **一个算法有一个或多个输出**

# 数据如何在计算机中表示

**变量的定义：**

程序中通过数据类型定义变量来表示自己需要的数据，可以给内存中**一块连续的空间**起一个名字，通过这个名字来操作它，这个名字就是变量名，也叫做**标识符**。

**变量定义必须放在变量使用之前。一般放在函数体的开头部分。**

**变量声明的基本语法：**

**数据类型 变量名1, 变量名2 …, 变量名n;**

**变量命名的规范：**

**变量名由数字，字母和下划线组成**

**数字不能打头**

**不能使用C语言中的关键字**

**C语言的变量名中的字母区分大小写**

**变量名应该尽量有意义，参考编程规范**

**变量的赋值：**

变量可以先定义再赋值，也可以在定义的同时进行赋值；在定义变量的同时赋初值称为**初始化**。

**注意，在定义中不允许连续赋值，如a=b=c=5是不合法的。**

**变量的使用规则**

**先定义，后使用**

**先赋值，后参加运算**

**常考试题：**

**（D）是合法的用户自定义标识符。**

**A. case（关键字） B.123f（数字） C. #html5（符号） D.\_int**

# 3、数据类型

## 整型数据

### 整型数据的分类

**无符号整数**

**Unsigned short（2）**

**Unsigned int（4）**

**Unsigned long（4）**

**Unsigned long long（8）**

**有符号整数**

**short (2)**

**int (4)**

**long (4)**

**long long (8)**

**不同的系统必须满足sizeof(short) ≤sizeof(int) ≤sizeof(long) ≤sizeof(long long)**

### 整型数据所分配的字节数及表示范围

|  |  |  |
| --- | --- | --- |
| **类型说明符** | **数的范围** | **字节数** |
| int | -32768~32767，即 -231~(231-1) | 4 |
| unsigned int | 0~65535，即 0~(232-1) | 4 |
| short int | -32768~32767，即 -215~(215-1) | 2 |
| unsigned short int | 0~65535，即 0~(216-1) | 2 |
| long int | -2147483648~2147483647，即 -231~(231-1) | 4 |
| unsigned long | 0~4294967295，即0~(232-1) | 4 |

**注意到：**有符号数表示的范围是**-2n-1~(2n-1-1)**

无符号数表示的范围是**0~(2n-1)**

### 整型数据在内存中存放的方式

在内存中数值是以**补码**的形式存储的

有符号整数在内存中的存储

正数：原码、补码、反码相同

负数： **原码**：数值的二进制表示 **反码**：符号位不变，数值的二进制按位取反

**补码**：数值的反码加1

### 整型数据的表示方法

**十进制数**

十进制数**没有前缀**。其数码为0～9。

以下是合法的十进制数：237、-568、65535、1627

以下是不合法的十进制数：023 (不能有前导0)、23D (非十进制数码)。2) **八进制数**

八进制数**必须以0开头，即以0作为八进制数的前缀**。数码取值为0～7。八进制数通常是无符号数。

以下是合法的八进制数：015(十进制为13)、0101(十进制为65)、0177777(十进制为65535)；

以下是不合法的八进制数：256(无前缀0)、03A2(包含了非八进制数码)、-0127(出现了负号)。3)

**十六进制数**

**十六进制数的前缀为0X或0x**。其数码取值为0~9，A~F或a~f。

以下是合法的十六进制数：0X2A(十进制为42)、0XA0 (十进制为160)、0XFFFF (十进制为65535)；

以下是不合法的十六进制数：5A (无前缀0X)、0X3H (含有非十六进制数码)。

## 字符型数据

### 字符数据的表示

字符型数据是**用单引号括起**来的一个字符。

例如：'a'、'b'、'='、'+'、'?'都是合法字符型数据。

### 字符型数据的特点

（1）字符型数据**只能用单引号括**起来，不能用双引号或其它括号。

（2）字符型数据**只能是单个字符**，不能是字符串。

（3）字符可以是字符集中任意字符。但数字被定义为字符型之后就不能参与数值运算。如'5'和5 是不同的。'5'是字符型数据，不能参与运算。

### 转义字符

**转义字符是一种特殊的字符。转义字符以反斜线"\"开头**，后跟一个或几个字符。转义字符具有特定的含义，不同于字符原有的意义，故称“转义”字符。

**转义字符主要用来表示那些用一般字符不便于表示的控制代码。**

|  |  |  |
| --- | --- | --- |
| 常用的转义字符及其含义 | | |
| 转义字符 | 转义字符的意义 | ASCII代码 |
| \n | 回车换行 | 10 |
| \t | 横向跳到下一制表位置 | 9 |
| \b | 退格 | 8 |
| \r | 回车 | 13 |
| \f | 走纸换页 | 12 |
| \\ | 反斜线符"\" | 92 |
| \' | 单引号符 | 39 |
| \” | 双引号符 | 34 |
| \a | 鸣铃 | 7 |
| \ddd | 1～3位八进制数所代表的字符 |  |
| \xhh | 1～2位十六进制数所代表的字符 |  |

### 字符类型说明符

字符变量的类型说明符是**char**。字符变量类型定义的格式和书写规则都与整型变量相同。例如：**char a,b;**

**每个字符变量被分配一个字节的内存空间，因此只能存放一个字符**。字符值是以**ASCII码**的形式存放在变量的内存单元之中的。

**由此可以看出：**

**（1）小写的ASCII码比大写的ASCII码大32**

**（2）后面的ASCII码比前面的ASCII码大**

**（3）'3'要变化成3需要‘3’-‘0’=3**

### 字符串

**字符串是由一对双引号括起的字符序列**。例如： "CHINA" ，“C program” ， "$12.5" 等都是合法的字符串。

**字符串和字符不同，它们之间主要有以下区别：**

（1）字符由**单引号**括起来，字符串由**双引号**括起来。

（2）字符只能是**单个字符**，字符串则可以含**一个或多个**字符。

**（3）可以把一个字符型数据赋予一个字符变量，但不能把一个字符串赋予一个字符变量。**

**字符占一个字节的内存空间。字符串占的内存字节数等于字符串中字节数加1。**增加的一个字节中存放字符**"\0"** (ASCII码为0)。这是字符串结束的标志。

## 实型数据

实型数据类型指的就是浮点数或实数

### 实型数据类型的分类

|  |  |  |  |
| --- | --- | --- | --- |
| 类型说明符 | 比特数（字节数） | 有效数字 | 数的范围 |
| float | 32(4) | 6~7 | -10的38次方~10的38次方 |
| double | 64(8) | 15~16 | -10的308次方~10的308次方 |

**由以上内容可以看出：**

**（1）实型数据没有办法精确的表示所有数字，因此表示的范围大**

**（2）相邻两个数据的距离是不等距的**

**（3）每一种实型数据都有自己的有效位数和精度**

### 实型数据的表示

**1) 十进制数形式**

由数码0~ 9和小数点组成。 例如：0.0、25.0、5.789、0.13、5.0、300.、-267.8230 等均为合法的实数。

**注意，必须有小数点。**

**2) 指数形式**

由十进制数，加**阶码标志“e”或“E”**以及阶码（只能为整数，可以带符号）组成。其一般形式为：

    a E n（a为十进制数，n为十进制整数）

其值为 a\*10n。如：2.1E5 (等于2.1\*105) 3.7E-2 (等于3.7\*10-2) 0.5E7 (等于0.5\*107) -2.8E-2 (等于-2.8\*10-2)

### [C语言](http://lib.csdn.net/base/c)中float和double数据在内存中的表示



前面的符号位很容易判断为0或者是1，指数部分称之为阶码，这个阶码部分是由偏置值（127或1111111）和阶码真值相加得到的，阶码真值符合IEEE-75标准，这个标准规定了是这个数一定能写成1.--\*2的多少次方的形式，尾数部分的存储是原码的存储。

# 4、常量和字面值

## 常量的分类

|  |  |
| --- | --- |
| **常量** | **说明** |
| 直接常量（字面量） | 可以立即拿来用，无需任何说明的量，例如：   * 整型常量：12、0、-3； * 实型常量：4.6、-1.23； * 字符常量：‘a’、‘b’。 |
| 符号常量 | 用标识符代表一个常量。在C语言中，可以用一个标识符来表示一个常量，称之为符号常量。 |
| Const常量 | Const int x=3.14 |

说明：符号常量在使用之前必须先定义，其一般形式为：**#define 标识符 常量**

## 字面值类型

### 整型字面值

**整型字面值表示**:

–十进制整数：123, -456, 0

–八进制整数：0123, 011

–十六进制整数：0x123, 0Xff

**整型常量的类型**

一般**默认整型常量为int型**。也可以根据其值所在范围或后缀确定其数据类型

长整型字面值 123L 23l,

无符号整型字面值 123U 123u

无符号长整型字面值 123UL 123ul

long long型字面值 123ll 123LL

### 浮点型字面值

**浮点型字面值得表示**

**– 科学计数法：5e3 , 12.3e3 ,123E2, 1.23e4**

**– 自然数计数：5.3 ,0.123, .123, 123.0**

**默认为double型，也可以使用后缀f，F，l，L确定类型。**

### 字符型字面值

**字符型字面值是用单引号括起来单个普通字符或转义字符。**

### 字符串字面值

**字符串字面值是由一对双引号括起来的字符序列（以’\0’结束）。**

**注意：1.字符串字面值会自动在字符串最后加上‘\0’，占据一个字节的内存**

**2.求字符串长度的时候注意转义字符占据一个字节，例如：’\xdd’等**

# 5、数据的输入和输出

## Printf函数

**printf函数称为格式输出函数，其关键字最末一个字母f即为“格式”(format)之意。其功能是按用户指定的格式，把指定的数据显示到显示器屏幕上。**

### printf函数调用的一般格式。

printf函数是一个标准库函数，它的函数原型在头文件“stdio.h”中。printf函数调用的一般形式为：**printf(“格式控制字符串”, 输出表列)**

其中**格式控制字符串用于指定输出格式**。格式控制串可由**格式字符串和非格式字符串两种组成**。格式字符串是以%开头的字符串，在%后面跟有各种格式字符，以说明输出数据的类型、形式、长度、小数位数等。

### printf函数格式中的格式字符串。

**格式字符串的一般形式为： [标志][输出最小宽度][.精度][长度]类型。**

#### （1）最小宽度

用十进制整数来表示输出的最少位数。

/\*

当转换值的字符数（含前缀）小于最小宽度说明时，则使用填充符（空格）将数值填充到最小宽度.

当转换值的字符数（含前缀）大于最小宽度说明时，最小宽度说明失效。

\*/

#### （2）精度

**精度格式符以“.”开头，后跟十进制整数。**

**本项的意义是：**

**如果输出数字，则表示小数的位数；**

**如果输出的是字符，则表示输出字符的个数；**

**若实际位数大于所定义的精度数，则截去超过的部分。**

#### （3）长度

长度格式符为h、l两种，h表示按短整型量输出，l表示按长整型量输出。

#### （4）类型

|  |  |
| --- | --- |
| 格式字符 | 意义 |
| d | 以十进制形式输出带符号整数(正数不输出符号) |
| o | 以八进制形式输出无符号整数(不输出前缀0) |
| x,X | 以十六进制形式输出无符号整数(不输出前缀Ox) |
| u | 以十进制形式输出无符号整数 |
| f | 以小数形式输出单、双精度实数 |
| e,E | 以指数形式输出单、双精度实数 |
| g,G | 以%f或%e中较短的输出宽度输出单、双精度实数 |
| c | 输出单个字符 |
| s | 输出字符串 |

## Scanf函数

**scanf函数称为格式输入函数，即按用户指定的格式从键盘上把数据输入到指定的变量之中。**

scanf函数是一个标准库函数，它的函数原型在头文件“stdio.h”中。scanf函数的一般形式为：**scanf(“格式控制字符串”, 地址表列)**

### 1、scanf格式控制字符串

**格式字符串的一般形式为：  %[\*][输入数据宽度][长度]类型**

#### （1）类型

|  |  |
| --- | --- |
| 格式 | 字符意义 |
| d | 输入十进制整数 |
| o | 输入八进制整数 |
| x | 输入十六进制整数 |
| u | 输入无符号十进制整数 |
| f或e | 输入实型数(用小数形式或指数形式) |
| c | 输入单个字符 |
| s | 输入字符串 |

#### （2）\* 符

**用以表示该输入项，读入后不赋予相应的变量，即跳过该输入值。**

#include<stdio.h>

int main(void)

{

     scanf("%d %\*d %d",&a,&b);

     return 0;

}

当输入为：1  2   3时，把1赋予a，2被跳过，3赋予b。

#### （3）宽度

#include<stdio.h>

int main(void)

{

    scanf("%5d",&a);//输入12345678只把12345赋予变量a，其余部分被截去。

    scanf("%4d%4d",&a,&b);//输入12345678将把1234赋予a，而把5678赋予b。

    return 0;

}

#### （4）长度

**h表示按短整型量输出，l表示按长整型量输出。**

**1》**%L修饰符(注意大写的L）：和浮点转换说明符一起使用，表示一个long double值。  
　　 示例：“%Lf”和“%10.4Le”  
　　 注：%l的修饰符(小写的l)：和整数说明符一起使用，表示一个long int 或者unsigned long int 类型值。  
　　 示例：“%ld”和“%8lu”。

**2》**%h修饰符：和整数转换说明符一起使用，表示一个short int 或者 unsigned short int 类型数值。  
　　 示例：“%hu”、“%hx”和“%6.4hd”

### 2、地址列表

例如：&a、&b分别表示变量a和变量b的地址。

**这个地址就是编译系统在内存中给a、b变量分配的地址。**

在赋值表达式中给变量赋值，如： a=567;

a为变量名，567是变量的值，&a是变量a的地址。但在赋值号左边是变量名，不能写地址，而scanf函数在本质上也是给变量赋值，但要求写变量的地址，如&a。

**这两者在形式上是不同的。&是一个取地址运算符，&a是一个表达式，其功能是求变量的地址。**

### 3、注意事项

**（1）scanf函数中没有精度控制，如：scanf("%5.2f",&a);是非法的。不能企图用此语句输入小数为2位的实数。**

**（2）scanf中要求给出变量地址，如给出变量名则会出错。如 scanf("%d",a);是非法的，应改为scnaf("%d",&a);才是合法的。**

**（3）在输入多个数值数据时，若格式控制串中没有非格式字符作输入，数据之间的间隔则可用空格，TAB或回车作间隔。C编译在碰到空格，TAB，回车或非法数据(如对“%d”输入“12A”时，A即为非法数据)时即认为该数据结束。**

**（4）在输入字符数据时，若格式控制串中无非格式字符，则认为所有输入的字符均为有效字符。**

# 6、实例

## （1）变量的初始化

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

int a = 123;//初始化

printf("%d\n",a);

a = 456;//赋值

printf("%d",a);

system("pause");

return 0;

}

## （2）利用scanf进行加法运算

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

int x;

int y;

int z;

printf("请输入两个整数x,y（数据间以空格分隔开）：");

scanf("%d %d",&x,&y);

z=x+y;

printf("%d+%d=%d\n",x,y,z);

system("pause");

return 0;

}

## （3）进行1—n的加法运算

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

unsigned int a;

unsigned int b;

printf("请输入你的进行运算的n的值：");

scanf("%d",&a);

b=a\*((a+1)/2);

printf("1+2+3+4+...+n(n=%d)=%d",a,b);

system("pause");

return 0;

}

## （4）精度的说明

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

double x=8.1234;

printf("%.4g",x);

system("pause");

return 0;

}

## （5）输入三个字符，输出这三个字符

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

char a;

char b;

char c;

a=getchar();

b=getchar();

c=getchar();

putchar(a);

putchar(b);

putchar(c);

system("pause");

return 0;

}

## 标识符

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

int x=34,y=-34;

printf("x=%8d,y=%8d\n",x,y);//表明最小宽度为8,用空格进行填充

printf("x=%08d,y=%08d\n",x,y);//用0而不是用空格进行填充

printf("x=% 08d,y=% 08d\n",x,y);//空格总是产生一个-号或者是空格

printf("x=%-8d,y=%-8d\n",x,y);//-号表示的是左对齐

printf("x=%- 8d,y=%- 8d\n",x,y);

printf("x=%-+8d,y=%-+8d\n",x,y);//+总是产生一个+或-号

printf("x=%8.4d,y=%8.4d\n",x,y);//精度.4表示是四位有效数字

printf("x=%-8.4d,y=%-8.4d\n",x,y);

system("pause");

return 0;

}

## 字符转换

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

char a,b,c;

printf("请用户输入三个小写字母，输入时字母之间使用空格隔开：");

scanf("%c %c %c",&a,&b,&c);

printf("%c %c %c相对应的ASCII码值为：%d %d %d\n",a,b,c,a,b,c);

printf("%c %c %c相对应大写字母为：%c %c %c\n",a,b,c,a-32,b-32,c-32);

system("pause");

return 0;

}

## （8）求圆柱的体积和表面积

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

float r,h,s,v;

printf("请输入您要求的圆柱的半径：");

scanf("%f",&r);

printf("请输入您要求的圆柱的高度：");

scanf("%f",&h);

s=2\*3.14\*r\*h+2\*3.14\*r\*r;

v=3.14\*r\*r\*h;

printf("您所要求的的圆柱所得到的表面积为：%.2f\n",s);

printf("您所要求的的圆柱所得到的体积为：%.2f\n",v);

system("pause");

return 0;

}