

第二章 数据类型及基本运算

第 1 题：03e2 是合法的浮点数吗？

答案：是的，0 表示 8 进制，03 就是八进制的 3，当然也等于十进制的 3，只要是 e 前面有数字，e 后面有整数，那么表达式就是正确的。

第 2 题：以下选项中不属于 C 语言的类型的是 (D)

- A) signed short int B) unsigned long int
C) unsigned int D) long short

第 3 题：以下选项中属于 C 语言的数据类型是 (C)

- A) 复数型 B) 逻辑型 C) 双精度型 D) 集合型

第 4 题：以下关于 long、int 和 short 类型数据占用内存大小的叙述中正确的是 (D)

- A) 均占 4 个字节 B) 根据数据的大小来决定所占内存的字节数
C) 由用户自己定义 D) 由 C 语言编译系统决定

第 5 题：以下不正确的 C 语言标识符是 (A)

- A) int B) a_1_2 C) ab1exe D) _x

第 6 题：以下是正确的 C 语言标识符是 (B)

- A) #define B) _123 C) %d D) \n

第 7 题：C 语言程序从 main() 函数开始执行，所以这个函数要写在 (D)

- A) 程序文件的开始 B) 程序文件的最后
C) 它所调用的函数的前面 D) 程序文件的任何位置

第 8 题：下列关于 C 语言的叙述错误的是 (A)

- A) 大写字母和小写字母的意义相同
B) 不同类型的变量可以在一个表达式中
C) 在赋值表达式中等号(=)左边的变量和右边的值可以是不同类型
D) 同一个运算符在不同的场合可以有不同的含义

第 9 题：在 C 语言中，错误的 int 类型的常数是 (A)

- A) 32768 B) 0 C) 037 D) 0xAF

第 10 题：已知：char a；int b；float c；double d；

执行语句 "c=a+b+c+d ;" 后 , 变量 c 的数据类型是(C)

A) int B) char C) float D) double

第三章 顺序程序设计

第 1 题 : printf("%08d\n%8d",1234,1234); 其中 %08d 中 08 表示什么 ?

答 : 在 printf 中 %08d 表示输出宽度为 8 , 不足 8 位的补 0 , %8d 表示输出宽度为 8 , 不足 8 位的补空格。本题输出 :

00001234

空空空空 1234

第二章 部分课后题解析 (P26)

2.1 中的错误为 : B: 01a 其中 0 表示 8 进制 , 那么 a 不合要求

C: 0668 其中 0 表示 8 进制 , 那么 8 不合要求

D: 0X 0X 为 16 进制 , 后面需要有内容

2.2 中的错误为 : A: 5e-4.9 其中 e 后面只能跟整数

C: 123e 后面必须有数 , 1.2e-.4 其中 e 后面必须为整数。

2.3 (1) x 的值为 15 , 逗号表达式的值为 20.

2.4 (2) $e(-4)=\exp(-4)$, $\log ab=\log_{10}(a)/\log_{10}(b)$

2.6(1) $1/2=0$, $5/(\text{float})2=2.5$

2.7 (5) 6 空空空空空空 6.000000 空空空空空空 12

(6) 580,90,590, 其中过程需要用到二进制转化来进行运算。

第三章 部分课后题解析 (P38)

3.1(1) main(){

int x=-2;

printf("dec:%d,oct:%o,hex:%x,unsigned:%u\n" ,x,x,x,x);}

分析：o-8 进制，x-16 进制，u-无符号整型

第一步：将-2 看成 2 转成二进制 0000 0000 0000 0010

第二步：取反加 1 1111 1111 1111 1110

第三步：每三位一组转成 8 进制（每四位一组转成 16 进制）

所以得到 $(-2)_{10} = (177776)_8 = (fffe)_{16}$

第四步：转成无符号型即第二步的值，即 $(-2)_{10} = (65534)_u$

注： $2^{16}-1 = (65535)_{10} = (1111\ 1111\ 1111\ 1111)_2$

3.1(2) main(){char c= 'B' ;

printf("dec:%d,oct:%o,hex:%x,ASCII:%c\n" ,c,c,c,c);

printf("c=%c\n" ,c+10);

分析：我们知道字母 'a' =97（让你唯一记住的一个 ASCII 码值哦），所以 'A' =97-32=65，'B' = 'A' +1=66；所以%d 对应的值为 66，又因为可以通过计算得到 $(66)_{10} = (102)_8 = (42)_{16}$ ；因为 c= 'B'，所以 c+10=' B' +10= 'L'。

3.1(3) # 对%f (%c, %s, %d, %u) 格式无影响，

对%o 格式，输出时加前缀 0（零），

对%x 格式，输出时候加前缀 0x。

float a=99.9;double b=98765.4567;

printf("(1)%-14.2f\n" ,a); 输出 (1) 99.90 空---空 (9 个空)

printf("(2)%#20.3f\n" ,a); 输出 (2) 空---空 99.900 (14 个空)

printf("(3)%6.4f\n" ,a); 输出 (3) 99.9000

printf("(4)%-lf\n" ,b); 输出 (4) 98765.456700

printf("(5)%#14.3lf\n" ,b); 输出 (5) 空---空 98765.457 (5 个空)

3.2 scanf("_%c_%c" ,&c1,&c2);格式特别注意 ,因为空格也是字符 ,所以含有空格的时候 ,必须老老实实的带空格输入 ,即_A_a 回车。

如果改为 scanf("%c%c" ,&c1,&c2);那么输入方法最好是 :Aa ,但是如果是下面的输入方法就是错误的 :

A 回车

a 回车 ,结束输入。

如果这时候你输入 A_a ,那么 c1=A,c2=_ ,而不是 c2=a ,其中_ 表示空格。

3.3 int x=10; float pi=3.1416;

printf("(1)%d\n" ,x); 输出为 (1) 10 ;

printf("(2)%6d\n" ,x); 输出为 (2) 空空空空 10 ;

printf("(3)%f\n" ,56.1); 输出为 (3) 56.100000 ;

printf("(4)%14f\n" ,pi); 输出为 (4) 空空空空空空 3.141600 ;

printf("(5)%e\n" ,568.1); 输出为 (5) 5.68100e+02 或 5.681000e+002 都对 ,我们统一输出格式为前一个 ;

printf("(6)%14e\n" ,x); 输出为 (6) 空空空 3.14160e+00 或 空 3.141600e+000 都对 ,我们统一输出格式为前一个 ;

printf("(7)%g\n" ,x); 输出为 (7) 3.1416 ; g 为实数输出 ,比 float 精度少一位 (即最多小数点后面保留 5 位 ,多了四舍五入 ,少了原值输出) 。

printf("(8)%12g\n" ,x); 输出为 (8) 空空空空空空 3.1416 ;
输出分析 :

```
main(){ float a;  
    scanf("%3f",&a);  
    printf("%f\n",a);  
}
```

如果输入为: 12.345 回车,那么返回值为: 12.000000 即只保留 12.作为 3 个长度,因为是浮点类型,所以后面保留 6 为小数。