# 2020上半学期计算机导论学习报告

## 20049200354-周志诚

1. 十，二，十六进制以及数值表示原理

1.十进制: 以 10 为底的幂展开式:

(123) 10 = 1 × 10 2 + 2 × 10 1 + 3 × 10 0 ;

自低到高各位数 (除 10 取余至商为 0): 3 = 123%10, 2 = 123/10%10 = 12%10,

1 = 123/10/10%10 = 1%10

二进制: 以 2 为底的幂展开式:

(77) 10 = (0100 1101) 2 = 0 × 2 7 + 1 × 2 6 + 0 × 2 5 + 0 × 2 4

+ 1 × 2 3 + 1 × 2 2 + 0 × 2 1 + 1 × 2 0

十六进制：以16为底的幂展开式

(77) 10 = (4D) 16 = 4 × 16 1 + 13 × 16 0

2.原码：正数的符号为 0，负数的符号为 1，其它位按一般的方法表示数的绝对

值。

x = (+103) 10 [x] 原 = (01100111) 2

x = (−103) 10 [x] 原 = (11100111) 2

反码: 正数的反码与原码相同；负数的反码是符号位不变，其他位按位取反

补码: 正数的补码与其原码相同；负数的补码为其反码最末位加 1. 即,

负数补码 = 反码 +1 = 2 n − 该数的绝对值, n 是编码二进制位数.

(77) 10 = (0100 1101) 2 , (−77) 10 = (1100 1101) 2

(−77) 补 = 2 8 − 77 = 1111 1111 + 0000 0001 − 0100 1101= 1111 1111 − 0100 1101

+ 0000 0001= 1011 0010 +0000 0001 = 1011 001

1.对于正数，原码 = 反码 = 补码

2.对于负数，补码 = 反码 + 1

反码 = 符号位不变, 其他位按位取反

3.补码是可逆的，即再对补码求补得到原码。

4.引入补码后，使减法统一为加法。

(+77) 补 + (−77) 补 = 0100 1101 + 1011 0011 = 0000 0000

1. 简单的c语言语句

#include<stdio.h> // standard input/output编译预处理指令

int main() // 主函数

{ // 函数开始标志

int i,p; // p，i表示整数，同类型变量可在一条语句中编译

float t；//t表示单精度浮点数

double q；// q表示双精度浮点数

char c；//c表示字符，如：\*，&， ，！等等

p=1;

|  |  |  |
| --- | --- | --- |
| 格式 | 对应的数据类型 | 备注 |
| %d | Int | 整数，十进制 |
| %f | Float | 单精度浮点数 |
| %c | Char | 字符 |
| %lf | Double | 双精度浮点数 |
| %.2f | Float | 保留两位小数，四舍五入，不适用于scanf |
| %.2lf | Double | 保留两位小数，四舍五入，不适用于scanf |
| %x | Int | 十六进制 |

|  |  |  |
| --- | --- | --- |
| %ld | Long int | 长整数 |

i=2;

p=2.0；

q=3.0；

c=‘A’；

}

printf(”%d\n”,p);输出语句//输入语句scanf，之中的语句需要使用取值符号&

return 0; // 函数执行完毕返回函数值0

} // 函数结束标志

在简单的代码之中有以下几点错误易范，在此处指出：

1. 每一条语句后面要加上‘；’
2. 使用输入变量时要使用取址符
3. 注意使用不同类型的数时要考虑四舍五入
4. 按照题目要求输入和输出值是选择，或者是空格
5. /n为换行符
6. 字符输出函数putchar（）
7. 字符输入函数 getchar（）如a=getchar（）//表示用键盘输入一个字符赋给a//遇到回车从缓冲区中接受字符
8. 简单的条件语句
   1. 1. if(条件表达式)

{ 表达式为真 (非 0) 时执行语句; }

#include<stdio.h> // standard input/output编译预处理指令

int main() // 主函数

{ // 函数开始标志

int a=10; // 定义变量a为整型数值, 定义变量时，可以指定变量的初值

if(a>=10)

{

printf(”a>=10\n”); // \n为换行符

}

else

{

printf(”a<10\n”); // \n为换行符

}

return 0; // 函数执行完毕返回函数值0

} // 函数结束标志

Ps：在使用if语句中判定符号为“==”而不是“=”

在if的条件里||表示或和&&表示且

在if语句里若要判别多种情况则需要使用else if语句，else等等

②while(条件表达式)

{ 表达式为真 (非 0) 时执行的语句;}

#include<stdio.h> // standard input/output编译预处理指令

int main() // 主函数

{ // 函数开始标志

int a=10; // 定义变量a为整型数值, 定义变量时，可以指定变量的初值

while(a>=0)

{

printf(”a=%d\n”,a); // \n为换行符

a--; // a= a - 1

}

return 0; // 函数执行完毕返回函数值0

} // 函数结束标志

Ps：while在编程中的意思为当满足条件是一直循环该语句，所以在使用时一定要限制其进入与输出条件。

条件那一条语句后面不需要加上‘；

* 1. For（表达式1；表达式2；表达式3）

表达式1：设置初始条件，只执行一次。可以为零个、一个或多个变量 (逗号隔开) 设置初值

表达式2：是循环条件表达式，用来判定是否继续循环。在每次执行循环体前先执行此表达式 (包括第 1 次循环)，决定是否继续执行循环

表达式3：作为循环的调整，例如使循环变量增值，它是在执行完循环体后才进行的。

* 1. Switch
  2. (int 或 char 型表达式)

int a;

scanf(”%d”,&a)

switch(a)

{

case 10: 多条语句1;

break;

case 20: 多条语句2;

break;

case 30: 多条语句3;

break;

default: 多条语句4;

}

等价于

int a;

scanf(”%d”,&a)

if(a == 10)

{ 多条语句1; }

else if(a == 20 )

{ 多条语句2; }

else if(a == 30)

{ 多条语句3; }

else

{ 多条语句4; }

⑤改变循环执行的状态：break，continue

while(表达式)

{

printf(”语句1”);

if(条件表达式) break; //提前终止循环

printf(”语句1”);

}//**提前终止循环**

while(表达式)

{

printf(”语句1”);

if(条件表达式) continue; //结束本次

循环, 进入下轮循环

printf(”语句1”);

}//提前结束本次循环

4.简单的运算符号以及数学函数

算术运算符 +,−,∗,/,%,++,−−

1. 整数 = 整数/整数, 结果不会四舍五入。
2. 浮点数=整数/浮点数

3. 余数 r=a%b, a,b 必须是整数

4.++i, - -i: 先加 (减)1, 再使用。i++, i- -: 先使用, 再加 (减)1

5.数学函数库

|  |  |
| --- | --- |
| Abs（x） | 对整数x求其绝对值 |
| Fabs（x） | 求浮点数x的绝对值 |
| Sqrt（x） | 计算根号x |
| Rand（void）//void表示空 | 产生-9032767的随机整数 |
| Pow（x，y） | 求x的y次方 |
| Log（x） | 求lnx |
| Log10（x） | 求log10x |

5数组的简单介绍

1. 引用数组: int i=0; a[i]
2. int a[10]; // 10个元素的整型数组
3. 数组元素的下标从 0 开始，int a[10]; 10 个整型元素，则最大下标值为 9，不存在数组元素 a[10]
4. int a[10]={1,2,3,4,5,6,7,8,9,10}; // 在定义数组时对全部数组元素赋予初值。
5. char c[10]={’a’,’b’}; // 可以只给数组中的一部分元素赋值。其他元素的值不确定
6. double d[]={10.0,10.2,10.3}; // 等效于 double d[3]={10.0,10.2,10.3}
7. d[2] = 20.2; // 修改第3个元素

二维数组

1. int a[5][10]; // 定义5行10列的整型二维数组
2. float pay[3][6]; // 定义3行6列的单精度浮点型二维数组
3. double pay1[3][6]; // 定义3行6列的双精度浮点型二维数组
4. #define M 5 // 行数

#define N 10 // 列数

int a[M][N]; // 定义M行N列的整型二维数组

1. 同一维数组是从[0][0]开始
2. 字符串的简单介绍

两种方式初始化字符组

1. s[0]=’a’, s=’b’, s=’c’, s[3]=’d’, s[4]以后的字符未赋值

char s[81]={’a’,’b’,’c’,’d’}; // 4个有效字符

2. str[0]=’a’, str[1]=’b’, str[2]=’c’, str[3]=’d’, str[4]=’\0’自动追加’\0’, 表示字符串结束)

char str[]=”abcd”; // 5个有效字符

3. 在printf，使用格式描述符%s, 输出以’\0’结尾的字符串，否则只能使用%c的格式描述符

4.在scanf，注意字符数组前没有取地址符号’&’， 以空格隔开，输入3个字符串。自动追加字符串结束字符’\0’， 遇空格或回车结束, 自动追加字符串结束字符’\0’

5.可以使用gets和puts输入和输出字符串

6. 字符串处理函数 #include<string.h>

7连接和复制字符串 strcat(s1,s2); strcpy(s1,s2)

8. strncpy(s1,”1234”,2); // 复制s2的前n个字符给s1,覆盖s1相应位置的字符, n要小于

s1的数组长度

9.str（s1，s2）从左到右逐字符比较串s1和s2，大于返回1, 小于返回-1, 同时到达’\0’返回0表示相等//仅比较前n个则同上

10.获取字符串的长度 strlen(s)

11. strlwr(s1); // 大写转小写，strupr(s2); // 小写转大写

例题解: 例 6.8 输入一行字符, 统计其中有多少个单词, 单词之间用空格 (可能多个) 分隔开。

char string[81];

int i,num=0,word=0;

char c;

gets(string); //输入一个字符串给字符数组string

for(i=0;(c=string[i])!=’\0’;i++) //只要字符不是’\0’就循环,条件表达式:第i个字

符赋值给c, 并且c̸=’\0’

{

if(c==’ ’) word=0; //若是空格字符，使word置0

else if(word==0) //如果不是空格字符且word原值为0

{

word=1; //使word置1

num++; //num累加1，表示增加一个单词

}

}

printf(”%d words in this line.\n”,num); //输出单词数