计算机导论报告

·进制

二进制：以2为底的幂展开式

十进制：“D” 二进制：“B” 十六进制：“H”

进制对照表

ASCII编码表大小字符间隔

  ‘a’-”A”=32

·原码 反码 补码

原码：正数的符号为0，复数的符号为-1，其它位按一般方法表示数的绝对值。

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：正数的补码与原码相同，负数的补码为其反码的最末位加1。

（77）D=（0100 1101）B     （-77）D=（1100 1101）B

（-77）补=1111 1111+0000 0001-0100 1101

        =1111 1111-0100 1101（反）+0000 0001=1011 0011

注：补码可逆，即对补码再补得到原码。

·变量在使用之前首先定义它的数据类型

#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,b;  //定义变量a，b为整型数值，同类型变量可以在一条语句中定义。

  float f;   //定义变量f为单精度浮点数

  double d;  //定义变量d为双精度浮点数

  char c;  //定义变量c为单个英文字母

  scanf(“%d”，&a）； //%变量格式符，&变量名

  a=10;

  b=20;

  f=10.2;

  d=20.3;

  c=‘A’;

  printf("%d %d %f %lf %c/n",a,b,f,c);       //\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

·字符输入函数getchar，遇到回车，开始从缓冲区中接收字符

#include<stdio.h>

int main()

{

  char a;  //定义一个变量

  a=getchar(); //从键盘输入一个字符，送给字符变量

  putchar(a);   //将变量a的值输出

  putchar('/n');  //向显示器输出一个换行符

  return 0;

}

char c;c=getchar()接收输入的字符，putchar()输出一个字符

·常变量

#include<stdio.h>

#define PI 3.14 //符号常量，注意没有分号

int main()

{

      int r=123;  //整型变量

      const int a=425;  //常变量

      r=100; //合法，因为r是变量，可以随时改变它的值

      a=100;  //不合法，因为a是常变量，不能更改

      printf("%f\n",2\*PI\*r);

      return 0;

}

·标识符

标识符就是一个对象的名字，用于标识变量、符号常量、函数、数组、类型等。

以下字母或下划线开始；区分大小写；不能使用关键字；最好有含义。

#include<stdio.h>

int main()

{

      int r=123;  //整型变量

      int 3a;    //不合法的变量名

    int break;   //不合法的变量名，因为break是关键字，被系统使用

      int Radius; //变量名最好有含义

      int radius;   //与Radius是不同的变量，c语言是小写敏感的语言

      return 0;

}

 ·常用格式描述符与数据类型的对应关系

|  |  |  |
| --- | --- | --- |
| 格式符 | 对应的数据类型 | 备注 |
| %d | int |  |
| %f | float |  |
| %c | char |  |
| %lf | double |  |
| %.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 |  |
| %ld | long int |  |

例题：

1、计算球体的重量

已知铁的比重是7.86（克/立方厘米），金的比重是19.3（克/立方厘米），写一程序，分别计算给定直径的铁球和金球的质量，假定PI=3.1415926.

输入说明：输入两个整数，分别表示铁球和金球的直径（单位毫米）。

输出说明：输出两个浮点数，分别表示铁球和金球的质量9(单位克），表六小数点后三位，两个浮点数之间用空格相隔。

#include<stdio.h>

int main()

{

      int a=0,b=0;

      scanf("%d %d",&a,&b);  //不能省略，“scanf”表示赋值

      float m,n;

      m=a\*1.0/20\*a\*1.0/20\*a\*1.0/20\*4\*3.1415926/3\*7.86; //整数相除仍为整数

      n=b\*1.0/20\*b\*1.0/20\*b\*1.0/20\*4\*3.1415926/3\*19.3;

      printf("%.3f %.3f",m,n);

      return 0;

}

或写为：

#include<stdio.h>

#include<math.h>                           //函数类型

#define PI 3.1415926                       //定义一个常量，注意没有分号

int main()

{

      int a,b;

      scanf("%d %d",&a,&b);

      float m=4.0/3.0\*pow(a/2.0/10,3) \*PI; //4.0/3.0结果是浮点数，4/3结果是整数

      float n=4.0/3.0\*pow(b/2.0/10,3) \*PI;  //pow(a,n)表示次方： a的 n次方

      printf("%.3f %.3f\n",7.86\*m,19.3\*n);   //%.3f表示输出结果保留3位小数

      return 0;

}

2、求1-1/2+1/3-1/4+···+1/99-1/100。

#include<stdio.h>

int main3()

{

      int a=1,b=1;

      float c=1.0;

    while(b<=100)

      {

            a=(-1)/a;

            b=b+1;

            c=c+a\*1.0/b;

}

      printf("%f",c);

      return 0;

}

3、m是一个已知的三位数，从左到右用a，b，c表示各位数字。

（1）、求数bac的值。

（2）、计算m最后一个字节。

#include<stdio.h>

int main()

{

      int m=0;

      scanf("%d",&m);

      int a,b,c;

      b=m/10%10;

      a=m/100;

      c=m%10;

      printf("a=%d,b=%d,c=%d\n",a,b,c);

      printf("%d\n",100\*b+10\*a+c);

      char k;

      k=m%16;

      printf("%x",k);

return 0;

}

注：1、“%”格式转换符，“\n”换行。

2、deno表示分母，int表示整数类型，float单精度浮点数，double双精度浮点数，sizeof所占字节数，char字符类型。

3、“float c=10.0”表示浮点数，一般写为此形式。

4、%x表示用十六进制表示，%d表示整数类型，%u按无符号表示，无十六进制中的正负号（“1、0”）。

5、a++表示先使用在加1，++a表示先加1再使用。

6、5%2表示取5/2的余数。

·使用函数时要加“#include<math.h>”

注意事项：（-1）^n用d=pow（-1,b)表示但不推荐，可以使用a=(-1)\*a。

·循环语句

while循环语句：

while(条件）   //不加“；”，否则陷入死循环。

{

}

if循环语句：（表达式非0时执行语句）

//case1

If（）

{

}

//case2

If（ )

{

}

else

{  
}

//case3

If（）

{  
}

else if(）  //else if可以有多个

{

}

else  //else最多有一个

{  
}

for 语句

for(初始化表达式；条件表达式；改变循环变量）

{

}

switch语句实现多分支选择结构

int a;

scanf(“%d”,&a)

switch（）

{

   case 10：多条语句1；

          break;

   case 20: 多条语句2；

          break;

   case30: 多条语句3；

          Break;

   default: 多条语句4；

}

·优先级

算数运算符>关系运算符>赋值运算符

注：1、“==”表示是否等于，“！=”表示不等于。

2、“a&&b”表示a和b，“a||b”表示a或b。

**循环语句的运用示例**

1.switch语句

#include<stdio.h>

int main()

{

      unsigned char a=0; //表示没有最前面的正负号（0或1）

      switch(a)   //a必须是整数表达式，也可写a+b;

      {

            case 0:

                  printf("x=0\n");

                  break;   //程序在这里停止，没有break，程序向下进行

            case 1:

                  printf("x=1/n");

                  break;

            case 10:

                  printf("x=10\n");

                  break;

            default:     //最后，相当于else。

                  printf("other");

      }

      return 0;

}

2.while语句

#include<stdio.h>

int main()

{

      float s,sum=0.0;

      int total1=10,i=1;

      while(i<=total1)  //i表示循环变量

      {                //循环体，有一系列语句组成（复合语句）

            scanf("%f",&s);

            sum=sum+s;   //也可写为sum+=s

            i++;         //i=i+1 必须要有改变循环变量的语句

      }

      printf("%f\n",sum/total1);

      return 0;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main()

{

      float s,sum=0.0;

      int total1=10,i=1;

      do

      {

            scanf("%f",&s);

            sum=sum+s;

            i++;

      }while(i<total1);

      printf("%f\n",sum/total1);

      return 0;

}

3.for循环语句

#include<stdio.h>

int main()

{

      float s,sum=0.0;

      int i=1,total1;

      for(i=1;i<total1;i++) //初始化表达式；条件表达式；改变循环变量

      {

            scanf("%f",&s);

            if(s<60)

              continue;    //不执行循环语句的后续部分

            sum=sum+s;

      }

      printf("%f\n",sum);

      return 0;

}

4.比较while语句和for语句

#include<stdio.h>

int main()

{

      int i=0;

      while(i<4)

      {

            if(i==2)

            {

                  i++;    //忽略此句，是常见错误，必须要有改变循环变量的语句

                  continue; //终止本轮循环，开始下轮循环

            }

      }

      printf("\nend i=%d\n",i);

      return 0;

}

#include<stdio.h>

int main()

{

      int i=0;

      for(i=0;i<4;i++)   //表达式3是continue的部分

      {

            if(i==2)

            {

                  i++;

                  continue;  //先执行表达式3，终止本轮循环，开始下轮循环

            }              //如果用break代替continue，表示终止整个循环，也不会执行表达式3

      }

      printf("\nend i=%d\n",i);

      return 0;

}