|  |
| --- |
| 2的11次方：2048  2的12次方：4096  2的13次方：8192  2的14次方：16384  2的15次方：32768  2的16次方：65536  2的17次方：131072  2的18次方：262144  2的19次方：524288  2的20次方：1048576  2的21次方：2097152  2的22次方：4194304  2的23次方：8388608  2的24次方：16777216  2的25次方：33554432  2的26次方：67108864  2的27次方：134217728  2的28次方：268435456  2的29次方：536870912  2的30次方：1073741824  2的31次方：2147483648  2的32次方：4294967296  2的63次方：9223372036854775808  2的64次方：18446744073709551616 |

|  |
| --- |
| 1.绝对值  2.取整和取余  3.三角函数  4.反三角函数  5.双曲三角函数  6.指数和对数  7.标准化浮点数  8.多项式  9.数学错误计算处理  1.绝对值  函数原型: int abs(int x);  函数功能: 求整数x的绝对值  int number=-1234;  abs(number);  函数原型:double fabs(double x);  函数功能:求浮点数x的绝对值.  float  number=-1234.0;  fabs(number);  函数原型:double cabs(struct complex znum)  函数功能:求复数的绝对值  参数说明:zuum为用结构struct complex表示的复数，定义如下：                struct complex                {                  double m;                  double n;                }  #include <stdio.h>  #include <math.h>  int main()  {      struct complex z;      double val;      z.x=2.0;      z.y=1.0;      val=cabs(z);      printf("The absolute value of %.2lfi %.2lfj is %.2lf",z.x,z.y,val);      return 0;  }  2.取整和取余  函数原型:     double ceil(double num)  函数功能:     得到不小于num的最小整数  函数返回:     用双精度表示的最小整数  函数原型:     double floor(double x);  函数功能:     求出不大于x的最大整数.  函数返回:     该整数的双精度实数  函数原型:double fmod (double x, double y); 返回两参数相除x/y的余数,符号与x相同。如果y为0，则结果与具体的额实现有关    int main()  {      double number=123.54;      double down,up;      down=floor(number);      up=ceil(number);      printf("original number     %10.2lf",number);//123.54      printf("number rounded down %10.2lf",down);  //123      printf("number rounded up   %10.2lf",up);    //124      return 0;  }  @函数名称:     modf  函数原型:     double modf(double val,double \*iptr);  函数功能:     把双精度数val分解为整数部分和小数部分,把整数部分存到iptr指向的单元.  函数返回:     val的小数部分  参数说明:     val 待分解的数  所属文件:     <math.h>  使用范例：  #include <math.h>  #include <stdio.h>  int main()  {      double fraction,integer;      double number=100000.567;      fraction=modf(number,&integer);      printf("The whole and fractional parts of %lf are %lf and %lf",number,integer,fraction);      return 0;  }  3.三角函数  函数原型:     double sin(double x);  函数功能:     计算sinx的值.正弦函数  函数原型:     double cos(double x);  函数功能:     计算cos(x)的值.余弦函数.  函数原型:     double tan(double x);  函数功能:     计算tan(x)的值,即计算角度x的正切数值  @函数名称:     hypot  函数原型:     double hypot(double x,double y)  函数功能:     已知直角三角形两个直角边长度，求斜边长度  函数返回:     斜边长度  参数说明:     x,y-直角边长度  所属文件:     <math.h>  #include <stdio.h>  #include <math.h>  int main()  {      double result;      double x=3.0;      double y=4.0;      result=hypot(x,y);      printf("The hypotenuse is: %lf",result);      return 0;  }  4.反三角函数  函数原型:     double asin(double x);  函数功能:     计算sin^-1(x)的值.反正弦值函数  函数原型:     double acos(double x);  函数功能:     计算cos^-1(x)的值,反余弦函数  函数原型:     double atan(double x);  函数功能:     计算tan^-1(x)的值.  函数原型:     double atan2(double x,double y);  函数功能:     计算tan^-1/(x/y)的值.求x/y的反正切值.  5.双曲三角函数  函数原型:     double sinh(double x);  函数功能:     计算x的双曲正弦函数sinh(x)的值.  函数原型:     double cosh(double x);  函数功能:     计算x的双曲余弦cosh(x)的值.  函数原型:     double tanh(double x);  函数功能:     计算x的双曲正切函数tanh(x)的值.  #include <stdio.h>  #include <math.h>  int main()  {      double result,x=0.5;      result=sin(x);      printf("The sin() of %lf is %lf",x,result);      return 0;  }  #include <stdio.h>  #include <math.h>  int main()  {      double result;      double x=0.5;      result=cosh(x);      printf("The hyperboic cosine of %lf is %lf",x,result);      return 0;  }  6.指数和对数  函数原型:     double exp(double x);  函数功能:     求e的x次幂  函数原型:     double fmod(double x,double y);  函数功能:     求整数x/y的余数  函数原型:    double frexp(double val,int \*eptr);  函数功能:    把双精度数val分解为数字部分(尾数)x和以2为底的指数n,即val=x\*2^n,n存放在eptr指向的变量中.  @函数名称:     pow  函数原型:     double pow(double x,double y);  函数功能:     计算以x为底数的y次幂,即计算x^y的值.  函数返回:     计算结果  参数说明:     x-底数，y-幂数  所属文件:     <math.h>  使用范例：  #include <math.h>  #include <stdio.h>  int main()  {      double x=2.0,y=3.0;      printf("%lf raised to %lf is %lf",x,y,pow(x,y));      return 0;  }  函数原型:     double sqrt(double x);  函数功能:     计算x的开平方.  函数返回:     计算结果  参数说明:     x>=0  所属文件:     <math.h>  使用范例：  #include <math.h>  #include <stdio.h>  int main()  {      double x=4.0,result;      result=sqrt(x);      printf("The square root of %lf is %lf",x,result);      return 0;  }  函数原型:     double log(double x);  函数功能:     求logeX(e指的是以e为底),即计算x的自然对数(ln X)  函数返回:     计算结果  参数说明:  所属文件:     <math.h>  使用范例：  #include <math.h>  #include <stdio.h>  int main()  {      double result;      double x=8.6872;      result=log(x);      printf("The natural log of %lf is %lf",x,result);      return 0;  }  @函数名称:     log10  函数原型:     double log10(double x);  函数功能:     求log10x(10指的是以10为底).计算x的常用对数  函数返回:     计算结果  参数说明:  所属文件:     <math.h>  使用范例：  #include <math.h>  #include <stdio.h>  int main()  {      double result;      double x=800.6872;      result=log10(x);      printf("The common log of %lf is %lf",x,result);      return 0;  }  #include <stdio.h>  #include <math.h>  int main()  {      double result;      double x=4.0;      result=exp(x);      printf("'e' raised to the power of %lf(e^%lf)=%lf",x,x,result);      return 0;  }  #include <math.h>  #include <stdio.h>  int main()  {      double mantissa,number;      int exponent;      number=8.0;      mantissa=frexp(number,&exponent);      printf("The number %lf is",number);      printf("%lf times two to the",mantissa);      printf("power of %d",exponent);      return 0;  }  7.标准化浮点数  函数原型:double modf (double x, double \*ip);  函数功能:将参数的整数部分通过指针回传, 返回小数部分，整数部分保存在\*ip中  函数原型: double ldexp(double x,int exponent)  函数功能: 计算x\*2的exponent次幂，即2\*pow(2,exponent)的数值  #include <stdio.h>  #include <math.h>  int main()  {      double value;      double x=2;      value=ldexp(x,3);      printf("The ldexp value is: %lf",value);      return 0;  }  8.多项式  @函数名称:     poly  函数原型:     double poly(double x,int degree,double coeffs[])  函数功能:     计算多项式  函数返回:     多项式的计算结果  参数说明:     计算c[n]\*x^n+c[n-1]x^n-1+.....+c[1]\*x+c[0]  所属文件:     <math.h>  #include <stdio.h>  #include <math.h>  int main()  {      double array[]={-1.0,5.0,-2.0,1.0};      double result;      result=poly(2.0,3,array);      printf("The polynomial: x\*\*3 - 2.0x\*\*2 + 5x - 1 at 2.0 is %lf",result);      return 0;  }  9.数学错误计算处理  @函数名称:     matherr  函数原型:     int matherr(struct exception \*e)  函数功能:     数学错误计算处理程序  函数返回:  参数说明:     该函数不能被直接调用，而是被库函数\_matherr()调用  所属文件:     <math.h>  #include<math.h>  int matherr(struct exception \*a)  {      return 1;  } |