Power in c

Power in c

Given two basic and exhibiting numbers, pow () function finds x high to y power i.e. xy. Basically, in the value representing C is calculated using the pow () function. Example: Entry: 2.0, 5.0 Output: 32 Explanation: pow (2.0, 5.0) runs 2.0 increased to power 5.0, which equates to 32 Entry: 5.0, 2.0 Output: 25 Explanation: pow (5.0, 2.0) runs 5.0 increased to power 2.0, which equates to 25 Syntax: double pow (double x, double y); Settings: The method takes two arguments: x: base value floating point y: floating point power value Program: The program below takes two whole user (a basic number and an exhibitor) and calculates the power. For example: In the case of 23 2 is the base number 3 is the exhibitor And, the power is equal to 2 - 2 - 2 Power of a number using the time Loop #include main () base int, exp; long result long ' 1; printf (Enter a basic number:); scanf (%d; printf (Enter an exhibitor:); scanf (%d, while (exp! - 0) - result - base; --

exp; printf (Response -%lld, result); return 0; Exit Enter a basic number: 3 Enter an exhibitor: 4 Answer - 81 The above technique only works if the exhibitor is a positive whole. If you need to find the power of a number with any real number as an exhibitor, you can use the pow function. Power Using pow() Function #include 'math.h'#include 'stdio.h' main () - double base, exp, result; printf (Enter a basic number:); scanf (%lf; printf (Enter an exhibitor:); scanf (%lf, calculates the power result - pow (base, exp); printf (%1lf.1lf -2lf, base, exp, result); return 0; Exit Enter a basic number: 2.3 Enter an exhibitor: 4.5 2.3-4.5 - 42.44 C-ProgrammingServer Lateral programming The power function is used to find the power given to two numbers that are the base and the exhibitor. The result is the high base to the power of the exhibitor. One example that demonstrates this is the following 'Base' 2 Exhibitor '5 2'5 '32 Therefore, 2 increased to power 5 is 32.A program that demonstrates the power function in C' is given as follows 'Example Live Demo' to include using namespace std; int main() int x, y, years - 1; cost 'lt; enter the base' value: i-) - years - x-x-1st-to-the-power-y-It's is-up 'examplethe' output' of the above-program is as'follows', 'enter' the base' value: '3' enter the exponent 4= 3= raised= to= the= power= 4= is= 81now= let= us= understand= the= above= program.the= values= of= base= and= exponent= are= obtained= from= the= user.= the= code= snippet= that= shows= this= is= as= follows= −cout=&gt;&lt;/&amp;;lt;ans;&gt; &lt; enter= the= base= value:= ;= cin=&gt; &gt; x; cout &lt; enter= the= exponent= value:= ;= cin=&gt; &gt; y; La puissance est calculée à l'aide de la boucle for qui s'exécute jusqu'à la valeur de l'exposant. Dans chaque passage, la valeur de base est&lt;/stdio.h&gt; &lt;/math.h&gt; &lt;/stdio.h&gt; &lt;/stdio.h&gt; avec ans. Après l'achèvement de la boucle for, la valeur finale de la puissance est stockée dans les ans variables. L'extrait de code qui montre ceci est le suivant −for(int i=0; i &lt;y; i++)= ans= *=x;Finally, the= value= of= the= power= is= displayed.= the= code= snippet= that= shows= this= is= as= follows= −cout=&gt;&lt;/y;&gt; &lt; x=&gt; &lt; raised= to= the= power=&gt;&lt;/&gt; &lt; y=&gt; &lt; is=&gt;&lt;/&gt; &lt;ans; published= on= 06-nov-2018= 14:03:30= 等级= 丫的，这个是excel里的函数吧。 = c里只有pow。 = power在c语言中没有特别的bai含义，既不是du关键字也不zhi是库函数的函数名。 可能是dao编程人员回自定义的一个变量名或函数名答。 举例如下：int power; // 定义一个int类型的变量，变量名为powerdouble power; // 定义一个double类型的变量，变量名为powerint power(int a, int b) // // 自定义一个函数，函数名为power{ return a+b;} c语言中只有pow库函数，没有power库函数。 函数原型：double= pow(double= x,= double= y);功= = = = 能：计算x^y返= 回= 值：计算结果如：double x = 2, y;y = pow(3.14, x); // 计算3.14的平方 p是保存上一次结果的变量，表示i个base相乘，循环结束就是base的n次方。 i表示循环计数变量，n次方，就是n个base相乘。 p=p*base,经过for循环结束，就表示n个base相乘。 #include=&gt;&lt;/ans;&gt; &lt;stdio.h&gt; int power(int n1, int n2); int main() { int base, a, result; printf (« Entrer le numéro de base: «); scanf (« %d », &amp;base); printf (« Enter power number(positive integer): «); scanf (« %d », &amp;a); résultat = puissance (base, a); printf (« %d^%d = %d », base, a, résultat); retour 0; } puissance int (base int, int a) { si (a != 0) retour (base * puissance (base, a - 1)); autre retour 1; } Numéro de base d'entrée de sortie : 3 Entrez le numéro de puissance (entier positif) : 4 3^4 = 81 Vous pouvez également calculer la puissance d'un nombre à l'aide d'une boucle. Si vous devez calculer la puissance d'un nombre augmenté à une valeur décimale, vous pouvez utiliser la fonction pow() bibliothèque. utiliser la fonction pow() dans la bibliothèque cmath, tgmath ou math.h. #include #include &lt;iostream&gt; &lt;cmath&gt;utilisant namespace std; int &lt; pow(a,b)=&gt; &lt; endl; // this calculates a^b return 0; } do note that if you give input to power as any data type other than long double then the answer will be promoted to that of double. that is it will take input and give output as double. for long double inputs the return type is long double. for changing the answer to int use, int c=(int)pow(a,b) But, do keep in mind for some numbers this may result in a number less than the correct answer. so for example you have to calculate 5^2, then the answer can be returned as 24.99999999999 on some compilers. on changing the data type to int the answer will be 24 rather than 25 the correct answer. So, do this int c=(int)(pow(a,b)+0.5) Now, your answer will be correct. also, for very large numbers data is lost in changing data type double to endl;= this= calculates= a^b= return= 0;= }= do= note= that= if= you= give= input= to= power= as= any= data= type= other= than= long= double= then= the= answer= will= be= promoted= to= that= of= double.= that= is= it= will= take= input= give= output= as= double.= for= long= double= inputs= the= return= type= is= long= double.= for= changing= the= answer= to= int= use,= int= c=(int)pow(a,b) but,= do= keep= in= mind= for= some= numbers= this= may= result= in= a= number= less= than= the= correct= answer.= so= for= example= you= have= to= calculate= 5^2,= then= the= answer= can= be= returned= as= 24.99999999999= on= some= compilers.= on= changing= the= data= type= to= int= the= answer= will= be= 24= rather= than= 25= the= correct= answer.= so,= do= this= int= c=(int)(pow(a,b)+0.5) now,= your= answer= will= be= correct.= also,= for= very= large= numbers= data= is= lost= in= changing= data= type= double= to=&gt;&lt;/ endl; // this calculates a^b return 0; } do note that if you give input to power as any data type other than long double then the answer will be promoted to that of double. that is it will take input and give output as double. for long double inputs the return type is long double. for changing the answer to int use, int c=(int)pow(a,b) But, do keep in mind for some numbers this may result in a number less than the correct answer. so for example you have to calculate 5^2, then the answer can be returned as 24.99999999999 on some compilers. on changing the data type to int the answer will be 24 rather than 25 the correct answer. So, do this int c=(int)(pow(a,b)+0.5) Now, your answer will be correct. also, for very large numbers data is lost in changing data type double to &gt; main() { int a,b; cin &gt;&gt; a &gt;&gt; b; cout&lt;/cmath&gt; &lt;/iostream&gt; &lt;/stdio.h&gt; &lt;/stdio.h&gt; long int. for example, you write long int (long int long) (pow (a,b)-0.5); and give the entry a-3 and b-38 then the result will be 1350851717672992000 while the correct answer is 1350851717672992089, this happens because pow () function back 1.35085th '18 which is promoted int as 1350851717672992000. I suggest writing a custom power function for such scenarios, such as:- long int __pow (long int a, long int b) - long long int q-1; for (long int i-0;i-1;i-) - return q and then call it when you want, int principal() - long int a,b; cin a long int c-__pow (a) (b) cost return 0; For numbers above the long int long range, use the boost library or chains. Strings.