题意兼解析

输入两个数字B和N。要求找一个A,使得A^N最接近B。

一开始想到的是暴力，不过想想或许可以换个方式来做这道题的。充分利用math.h里的pow()，先对B开n次方，就是用pow(b, 1.0/n).得到一个值right，然后分别对这个值取上整high和下整low，然后再计算pow(high,n)，pow(low, n)，比较这两个值中最小的那个就是答案。

#include <iostream>

#include<stdio.h>

#include<cmath>

using namespace std;

int main()

{

int B,N;

while(~scanf("%d%d",&B,&N) &&(B||N))

{

double x=pow(B,1.0/N);

int low=(int)x;

int high=(int)x+1;

int t1=pow(low,N);

int t2=pow(high,N);

if(abs(B-t1)<=abs(B-t2))

printf("%d\n",low);

else printf("%d\n",high);

}

return 0;

}

暴力写法

题意应该很好看懂，给你B,K，求A^K最接近B的那个A是多少，当时想了一会，没什么思路，，差点用快速幂求了，不过认真看题发现数据范围并不大，B才在百万级内，完全可以用暴力解决， 要知道2^20等于1024\*1024>1000000，即最大只需for遍历到20就可以了；

#include<cstdio>

#include<cstring>

#include<cstdlib>

#include<iostream>

#include<algorithm>

#include<cmath>

using namespace std;

int main()

{

int B,N,i;

while(~scanf("%d%d",&B,&N)&&B&&N)

{

if(N==1)

printf("%d\n",B);

else if(N>=B)//如果相等，2^k次方肯定远大于x;

printf("1\n");

else

{

for(i=1;;i++)//遍历求出满足条件A的范围；

if((int)pow(i\*1.0,N)<=B && B<=(int)pow((i+1)\*1.0,N))

break;

if(abs((int)pow(i\*1.0,N)-B)<=abs((int)pow((i+1)\*1.0,N)-B))

printf("%d\n",i);

else

printf("%d\n",i+1);

}

}

}