思维 比例扩大+下取整

将原有的每个雕塑的坐标位置，映射在一个总长为n+m的数轴上，设第一个点的坐标为0，（新的等分点必然有至少有一个和原来n等分的等分点重合，因为等分点可以等距的绕圆周旋转，总可以转到有至少一个重合的，不妨就让这个重合的点是坐标为0的点）从0到n+m-1的每个整数端点为添加雕塑之后每个雕塑的正确位置。pos[i]代表原来的第i个点在新数轴上的坐标，i/n是在总长为1的线段上n等分的第i个点所占的比例，那么在总长为n+m的线段上它的坐标pos[i]=i/n\*(n+m).由于第一个雕塑的坐标保持为0，从第二个雕塑开始枚举，判断当前雕塑的坐标距离哪个整数的端点最近（用四舍五入判断，这又是比较精彩实用的技巧），较近的这段距离，即为它所需要移动的距离，用一个变量来累加结果。在这里不可能出现两个雕塑都距离同一个整数端点较近的情况，因为现在有 m+n 个雕塑，每个雕塑之间的间隔为1，而之前只有 n 个雕塑，那么之前的雕塑之间的间隔一定大于1，所以不可能都靠近同一个整数端点。最后将总的移动距离ans转化为在周长为10000的圆上，用 ans/(m+n)\*10000 即可。

#include<stdio.h>

#include<string.h>

#include<iostream>

#include<algorithm>

#include<set>

#include<iomanip>

#include<bits/stdc++.h>

using namespace std;

#define ll long long

int main()

{

// freopen("input.txt","r",stdin);

int n,m;

while(cin>>n>>m)

{

double pos;

double ans=0;

for(int i=1;i<n;++i)

{

pos=1.0\*i/n\*(n+m);

ans+=fabs(pos-floor(pos+0.5));

}

ans/=n+m;

cout<<fixed<<setprecision(4)<<ans\*10000<<endl;

}

return 0;

}