#include<bits/stdc++.h>

#define llINF 9223372036854775807

#define pi 3.141592653589793//23846264338327950254

#define endl "\n"

#define mp make\_pair

#define pb push\_back

#define ll long long

#define int long long

#define IOS ios::sync\_with\_stdio(0); cin.tie(0); cout.tie(0);

**using** **namespace** std;

**const** ll maxn=1e3+7;

**const** **double** eps=1e-10;

**const** ll mod=1e9+7;

一开始的班级人数就会存在一个最大值Max

求最小的所有班级人数的最大值x，分配的时候要让他尽可能小

我们的策略必然是让其他还没到达Max这个值的班级优先分配学生进去，直到等于Max，都不会改变x

因此我们用一个sum记录所有班级都打到Max人数需要安排多少人，如果m<=sum就代表最小的x就是Max

如果max>sum

也就是所有班级都放到Max人后，还剩下temp=m-sum个人代放

我们采取平均放的方式，一个一个依次分配到班级里

如果temp恰好被n整除，那么每个班级是平均的，增加的人数为temp/n

如果temp无法被n整除，那么temp%n个班级会比其他班级多一个人，增加的人数为temp/n+1

求最小的所有班级人数的最大值x,我们只需要把所有的m个新学生全部分配到一开始的最大班级人数Max的那个班去就行了

int32\_t main()

{

    IOS;

    ll n,m;

    cin>>n>>m;

    vector<ll>num(n);

    ll Min=llINF,Max=-llINF;        //Min为这些班级一开始人数最小的值，Max为最大值的值

**for**(ll i=0;i<n;i++)

    {

        cin>>num[i];

        Min=min(Min,num[i]);

        Max=max(Max,num[i]);

    }

    ll sum=0;

**for**(ll i=0;i<n;i++) sum+=Max-num[i];

**if**(m<=sum) cout<<Max<<' ';

**else**

    {

        ll temp;

**if**((m-sum)%n) temp=(m-sum)/n+1;

**else** temp=(m-sum)/n;

        cout<<temp+Max<<' ';

    }

    cout<<Max+m<<endl;

}