**[Chinese Remainder Theorem](http://lightoj.com/volume_problemcategory.php?category=Chinese%20Remainder%20Theorem), O(log (n)^2), where n=mod1\*mod2\*…modn**

**Tested on LOJ 1319 - Monkey Tradition**

Finds lowest x such that

X % mod1= rem1

X % mod2 = rem2

………………………… (upto n)

All mod have to be relatively prime to each other, adding tot to ret gives a different x.

long long chnRem(vector<int>mod, vector<int>rem)

{

int n=mod.size();

long long tot=1, ret=0;

for(int i=0;i<n;i++)

tot\*=mod[i];

for(int i=0;i<n;i++)

{

long long tmp=tot/mod[i];

while(tmp%mod[i]!=rem[i])

tmp+=tot/mod[i];

ret=(ret+tmp)%tot;

}

return ret;

}