I. Multiplication Table

```
int n,m,k;
cin>>n>>m>>k;
int l=1,r=n*m,ans=-1;
while(l<=r)
{
    int mid=(l+r)/2, cur=0;
    for(int i=1;i<=n;i++)
    {
        cur+=min(m, mid/i);
    }
    if(k<=cur)
    {
        ans=mid;
        r=mid-1;
    }
    else
    l=mid+1;
}
cout<<ans;
```

```
1 2 3 4 5 6
2 4 6 8 10 12
3 6 9 12 15 18
4 8 12 16 20 24
mid = 14
min(14/1,6)=6
min(14/2,6)=6
min(14/3,6)=4
min(14/4,6)=3
19 elements <= 14
```

mid=10 6 5 3 2	mid=6 6 3 2 1	mid=4 4 2 1 1

Binary Exponentiation

13
2^0 + 2^2 + 2^3

a^x * a^y = a^(x+y)

a^13 = a^(2^0 + 2^2 + 2^3) = (a^(2^0)) * (a^(2^2)) * (a^(2^3))

1101
1
a^1
a^1
a^1 * a^4
a^1 * a^4 * a^8
a^0
0000001101
a^(2^0) a^(2^1) a^(2^2) a^(2^3)
a^1 a^2 a^4 a^8

```
int power(int a, int b, int p)
    {
        if(a==0)
        return 0;
        int res=1;
        a%=p;
        while(b>0)
        {
            if(b&1)
            res=(res*a)%p;
            b>>=1;
            a=(a*a)%p;
        }
        return res;
    }
```

Modular Arithmetic

```
a %= m;
if(a<0)
     a+=m;


a>=0 && b>=0
(a+b)%m ==  ((a%m) + (b%m))%m
(a-b)%m ==  ((a%m) - (b%m)+ m)%m
(a*b)%m ==   (1ll * (a%m) * (b%m))%m
int a=(1ll * (a%m) * (b%m))%m;

a=(a+b)%m;
Assume 0 <= a,b < m here
```

Add:

```
a+=b;
if(a>=m)
a-=m;
```

Subtract:

```
a-=b;
if(a<0)
a+=m;

a=(a+b);
a+=b;

s=(s + 'a'); O(n)
s+='a';      O(1)
```

Division:

```
a/b

a*(1/b)%m

((1/b)*b)%m == 1

(a*power(b, m-2, m))%m;    m is prime here
```

Ex:
⅔ % 5
⅓ %5
3^(5-2)= 27 2

(3 * ⅓)%5 1

3*2%5
6%5 1

2*2 -> 4

```
long long power(long long a, long long b, long long MOD)
{
    if(b==0)
        return 1;
    else
    {
        long long z = power(a,b/2,MOD);
        long long ans = (z*z)%MOD;
        if(b%2==1)
            ans = (ans*a)%MOD;
        return ans;
    }
}
```

prime no:

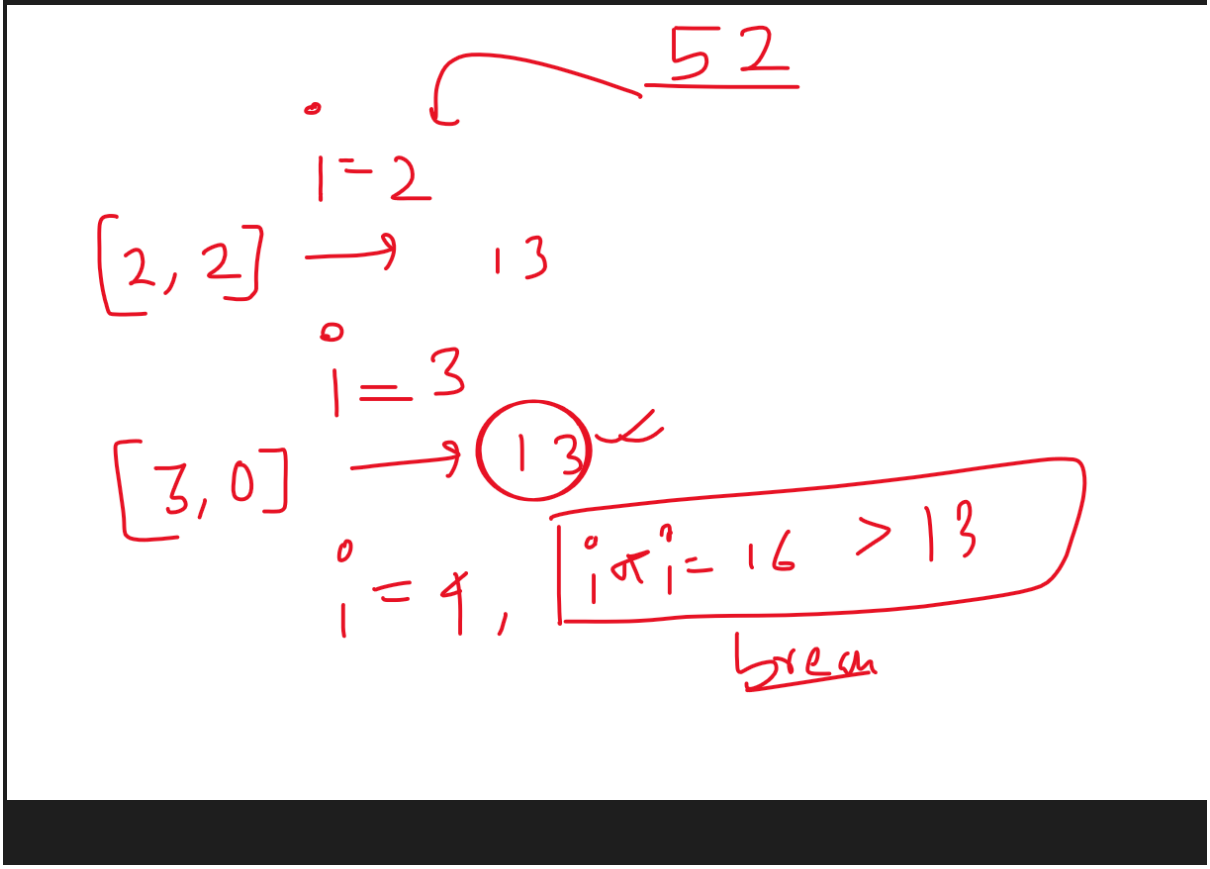
13 -> 1 and itself

40
-> 2 (5, 3)
-> 3 (5, 0)
-> 4 (5, 0)
-> 5 (1, 1)

```cpp
bool is_prime(int n) {
    int div = 0;

    for(int i=1;i<=n;i++) { // O(n)
        if(n%i==0) div++;
    }

    return (div == 2);
}
```

```cpp
    for(int i=1;i*i<=n;i++) { // O(sqrt(n))
        if(n%i==0) {
            div ++;
            if(i != n/i) div ++;
        }
    }
    // 25 -> 5*5

    vector<int> divisors;

    for(int i=1;i*i<=n;i++) { // O(sqrt(n))
        if(n%i==0) {
            divisors.push_back(i);
            if(i != n/i) {
                divisors.push_back(n/i);
            }
        }
    }

    map<int, int> m;
    for(int i=2;i<=n;i++) { // O(n + logn)
        if(n%i==0) {
            int count = 0;
            while(n%i == 0) {
                count ++;
                n/= i;
            }

            m[i] = count;
```

```
    }
}

for(pair<int,int> x:  m) {
    cout << x.first <<" "<< x.second << '\n';
}
```



Almost All Divisors

https://codeforces.com/problemset/problem/1165/D

assume x = correct;

x = min * max;

generate all div of x -> match it given arr;

ex : a[] = {3, 4}

let x = 12 = 3*4

b[] = {1, 2, 3, 4, 6, 12}

```cpp
    vector<int> a(n);

    for(int i=0;i<n;i++) cin >> a[i];

    sort(a.begin(), a.end()); // 3, 4

    int x = a[0] * a.back(); // 3*4 = 12

    vector<int> b;

    for(int i=2;i*i<=n;i++) {
        if(n%i==0) {
            b.push_back(i);
            if(i != n/i) {
                b.push_back(n/i);
            }
        }
    }

    sort(b.begin(), b.end()); // 2, 3, 4, 6

    cout << ((a==b) ? x : -1) << '\n';
```

Your a given pi, xi
pi = the prime
xi = the times it divides

number of div = (1 + x1)*(1 + x2).. (1 + xn)

sum of div = (1 + p1 + p1^2... p1^x1)*(1 + p2 + p2^x2).. (1 + pn +.. pn^xn)

prod of div = n ^ (d(n)/2) : d(n) = count of div

https://www.spoj.com/problems/CDRSANJ/

```
int b = 60;

    while(hi >= lo) {
        mid = lo + (hi-lo)/2;
        if(n % (1ll << mid) == 0) {
            ans = mid;
            lo = mid + 1;
        }
        else {
            hi = mid - 1;
        }
    }
```