**SOLUTION**

**Bài 1 CĂN BẬC HAI**

Mọi số tự nhiên *a* > 0 luôn luôn phân tích được dưới dạng .

Như vậy ta cần tìm thừa số x đầu tiên lớn nhất thỏa mãn biểu thức trên, x chính là phần nguyên cần tìm, phần dưới căn là y=a/x2, sau khi tìm được thì Break thoát kết thúc việc tìm kiếm.

**Bài 2**  **SỐ T – NGUYÊN TỐ**

Để giải quyết được dữ liệu của đề bài (2 ≤ *n* ≤ 109)

Bản chất của bài toán chính là tìm ước, với điều kiện ước số của số N bằng 3

Như vậy ta xây dựng thuật toán đếm ước của số tự nhiên n

long long uoc(long long n)

{

int dem=0;

for (long long i=1;i<=sqrt(n);i++)

if (n%i==0)

{

dem++;

if (n/i!=i) dem++;

if (dem>3) break;

}

return dem;

}

Sau đó ta tìm số n chỉ có 3 ước như sau:

int main()

{

cin>>m;int t=0;

for (long long j=1; j<=m;++j)

if (uoc(j)==3) t++;

cout<<t;

return 0;

}

Hoặc ta có thuật toán khác như sau:

#include<bits/stdc++.h>

using namespace std;

bool uoc(long long n)

{

if(n<2) return 0;

else

{

for(long long i=2;i<=sqrt(n);i++)

{

if(n%i==0) return 0;

}

return 1;

}

}

int main()

{

freopen("prime.inp","r",stdin);

freopen("prime.out","w",stdout);

ios::sync\_with\_stdio(0);

cin.tie(0);

cout.tie(0);

int dem=0;

long long n;cin>>n;

for(long long i=2;i<=sqrt(n);)

{

if(uoc(i)) dem++;

if(i%2==1) i+=2;

else if(i%2==0) i++;

}

cout<<dem;

}

**Bài 3**: **PHÒNG CHỐNG DỊCH COVID-19**

Gọi a[1], a[2], a[3], a[4] là lần lượt là số lượng của các nhóm 1 người, 2 người, 3 người, 4 người

Đầu tiên tính tổng dem = a[4] + a[3];

Sau đó xét các trường hợp:

if (a[1] > a[3])

a[1] = a[1] - a[3];

else

a[1] = 0;

Xét temp = a[1] +2\*a[2];

if (temp % 4 == 0)

{

dem += temp / 4;

}

else

{

dem += temp / 4 + 1;

}

Dem cuối cùng là giá trị cần tìm

**Bài 4:** **CẶP KHÁN GIẢ MAY MẮN**

- Xây dựng mảng T với T[i] là phần tử nhỏ nhất từ a1 đến ai. Khi đó mảng T sẽ là dãy giảm dần.

- Trên mảng T duyệt tất cả các chỉ số j từ 1 đến N, với mỗi j ta tìm kiếm nhị phân trên đoạn chỉ số [1, j-1] phần tử lớn nhất thỏa mãn ≤ a[j]-p.