

Div2 2019每周练习(五)标程和题解

A

快速幂模板题：

```
#include<iostream>
using namespace std;
#define ll long long
ll t, n;
ll qpow(ll a, ll b, ll m)
{
    ll ans = 1;
    while (b > 0)
    {
        if (b & 1)
        {
            ans = ans * a % m;
        }
        a = a * a % m;
        b >>= 1;
    }
    return ans;
}
int main()
{
    cin >> t;
    while(t--)
    {
        cin >> n;
        cout << qpow(n, n, 10) << endl;
    }
    return 0;
}
```

B

快速幂模板题

```
#include<iostream>
using namespace std;
#define ll long long
#define mod 1000000007
ll n;
ll qpow(ll a, ll b, ll m)
{
    ll ans = 1;
    while (b > 0)
    {
        if (b & 1)
```

```

        {
            ans = ans * a % m;
        }
        a = a * a % m;
        b >>= 1;
    }
    return ans;
}
int main()
{
    while(cin >> n)
    {
        cout << qpow(2, n * n - 1, mod) << endl;
    }
    return 0;
}

```

C

GCD模板题

```

#include<iostream>
using namespace std;
#define ll long long
ll a, b;
ll gcd(int a , int b)
{
    return b == 0 ? a : gcd(b, a % b);
}

int main()
{
    while(cin >> a >> b)
    {
        cout << gcd(a, b) << endl;
    }
    return 0;
}

```

D

只有最大公约数为1才能遍历所有位置

```

#include<iostream>
int gcd(int a , int b)
{
    return b == 0 ? a : gcd(b, a % b);
}
int main()
{
    int n,m;

```

```

while(scanf("%d%d",&n,&m),n!=-1||m!=-1)
{
    if(gcb(n,m)==1)
        printf("YES\n");
    else{
        printf("POOR Haha\n");
    }
}
return 0;
}

```

E

```

#include<iostream>
using namespace std;
#define ll long long
ll t, a, b, c, d;
ll gcd(int a , int b)
{
    return b == 0 ? a : gcd(b, a % b);
}

int main()
{
    cin >> t;
    while(t--)
    {
        cin >> a >> b >> c >> d;
        cout << (a * d + b * c) / gcd(a * d + b * c, b * d) << " " << b * d / gcd(a * d + b * c, b * d) << endl;
    }
    return 0;
}

```

F

两个最小公倍数公式为: $a/\gcd(a,b) * b$

```

#include<iostream>
using namespace std;
#define ll long long
ll a, b;
ll gcd(int a , int b)
{
    return b == 0 ? a : gcd(b, a % b);
}

ll lcm(ll a,ll b)
{
    return a/gcd(a,b)*b;
}

```

```

int main()
{
    while(cin >> a >> b)
    {
        gcd(a, b);
        cout << lcm(a,b) << endl;
    }
    return 0;
}

```

G

从第二数开始依次和第一个数取lcm(最小公倍数)

```

#include<iostream>
using namespace std;
#define ll long long
ll t, n;
ll w[1005];
ll sum, ans;
ll gcd(int a , int b)
{
    return b == 0 ? a : gcd(b, a % b);
}

int main()
{
    cin >> t;
    while(t--)
    {
        ans = 0;
        cin >> n;
        for(int i = 1; i <= n; i++) cin >> w[i];
        ans = w[1];
        for(int i = 2; i <= n; i++)
        {
            ans *= w[i] / gcd(ans, w[i]);
        }
        cout << ans << endl;
    }
    return 0;
}

```

H

数据较小，暴力判断素数即可

```

#include<cmath>
#include<stack>
#include<queue>

```

```

#include<cstring>
#include<algorithm>
using namespace std;
bool is_p(int x)
{
    if(x==2)return false;
    for(int i=2;i*i<=x;i++)
        if(x%i==0)
            return false;
    return true;
}
int n,sum;
int main()
{
    while (cin>>n)
    {
        sum=0;
        for(int i=2;i<=n/2;i++)
            if(is_p(i)&&is_p(n-i)) sum++;
        cout<<sum<<endl;
    }
}

```

I

可以先打出范围内的素数表，然后去枚举方程的正确性。

```

#include<cstdio>
int prime[3002];
bool judge(int x,int y)
{
    for(int i=x; i<=y; i++)
        if(prime[i*i+i+41])
            return false;
    return true;
}
int main()
{
    prime[0]=prime[1]=1;
    for(int i=2; i<3002; i++)
    {
        if(!prime[i])
            for(int j=i*i; j<3002; j+=i)
                prime[j]=1;
    }
    int a,b;
    while(scanf("%d %d",&a,&b),a||b)
    {
        printf("%s\n",judge(a,b)?"OK":"Sorry");
    }
    return 0;
}

```

素数删模板题，主要在于题意有点坑点。

题意：

给你两个数 n 和 c ，如果 $1 \sim n$ （包括 1 和 n ）之间的素数个数为偶数个，则输出中间 $c * 2$ 个素数否则输出中间 $c * 2 - 1$ 个素数。（此题中 1 为质数）

```
#include <stdio.h>
#include <math.h>
bool nprim[1005];
int count[1005];
int main()
{
    for(int i=2;i * i<=1005;i++)
    {
        if(!nprim[i])
            for(int j=2*i;j<=1005;j+=i)
            {
                nprim[j]=true;
            }
    }
    nprim[0]=true;
    int cnt=0;
    for(int i=1;i<1005;i++)
    {
        if(!nprim[i])
            cnt++;
        count[i]=cnt;
    }
    int n,c;
    while(~scanf("%d %d",&n,&c))
    {
        int all=c*2;
        if(count[n]%2==1) all--;
        int cut=(count[n]-all)/2;
        bool first=true;
        printf("%d %d: ",n,c);
        for(int i=1;i<=n;i++)
        {
            if(cut>0&&!nprim[i])
            {
                cut--;
                continue;
            }
            if(cut<=0&&!nprim[i])
            {
                if(!all) break;
                all--;
                if(first)
                {
                    first=false;
                    printf("%d",i);
                }
            }
            else
            {
                printf(" %d",i);
            }
        }
    }
}
```

```

    }
}
}
printf("\n\n");
}
return 0;
}

```

K

求素数的问题，但数据范围是 $0 < n < 1000000$ 这么大，很明显平时我们穷举数字，再一个个去判断是否是素数，肯定超时，这就要用到效率非常高的求素数的算法了，竞赛中一般用素数筛选法来处理此类问题，关于素数筛选法生成素数表，这个网上资料也很多，可以自己搜索学习，简单的说是用了一个思想：（素数的倍数一定是个合数）。建立一个全1的数组(下标2~N)，先将下标是2的倍数的全置0，再将下标是3的倍数全置0,.....，以此类推，最后剩下的，仍是1的那些下标，就肯定是素数了。这样做远远比过去穷举数字快是不是？你们可以体会下。

```

#include <stdio>
const int MAX=1000000;
int prime[MAX]={0};
int mark[MAX]={0};
void init()
{
    for(int i=2,n=1;i<MAX;i++)    //枚举
    {
        if(prime[i]==0)    //0标志位代表是素数
        {
            mark[i]=n++;    //记录这个素数对应的序号
            for(int j=i;j<MAX;j=j+i)    //素数的倍数，肯定不是素数
            {
                prime[j]=i;    //去掉标志位，记录最大质因数
            }
        }
    }
}
int main()
{
    int n;
    init();
    while(~scanf("%d",&n))
    {
        printf("%d\n",mark[prime[n]]);    //prime中存放这个数的最大质因数
    }
    return 0;
}

```

L

一道比较毒瘤的，需要预算空间的打表题。。

```
#include <stdio.h>
using namespace std;
bool a[9989900];
int prime[1005];
void set()
{
    int i,j;
    for(i = 2; i<=9989899; i+=2) a[i] = true;
    for(i = 3; i<=3161; i++)
    {
        if(a[i]) continue;
        for(j = i+i; j<=9989899; j+=i) a[j] = true;
    }
}

int huiwen(int n)
{
    int a = 0, b = n;
    while(b)
    {
        int r = b%10;
        a = a*10+r;
        b/=10;
    }
    if(a == n) return 1;
    return 0;
}

int main()
{
    int n,m,i,k = 2;
    set();
    prime[0] = 5;
    prime[1] = 7;
    for(i = 11; i<=9989899; i+=2)
    {
        if(!a[i] && huiwen(i)) prime[k++] = i;
    }
    while(~scanf("%d%d",&n,&m))
    {
        for(i = 0; i<k; i++)
        {
            if(prime[i]<n) continue;
            else if(prime[i]>m) break;
            else printf("%d\n",prime[i]);
        }
        printf("\n");
    }
    return 0;
}
```


M

模拟二分搜索，二分思想模板题

```
#include<iostream>
#include<algorithm>
using namespace std;
int n,k;
int ans;
int binary_search(int l,int r,int k)
{
    int sum=0;
    while(l<=r)
    {
        sum++;
        int mid=(l+r)/2;
        if(mid==k) return sum;
        if(mid<k) l=mid+1;
        if(mid>k) r=mid-1;
    }
}
int main()
{
    while(cin>>n>>k)
    {
        ans=binary_search(1,n,k);
        cout<<ans<<endl;
    }
}
```