

# AtCoder Beginner Contest 177

## A - Don't be late

题意：每秒走s格，t秒能否走完d格？

分类：简单数学

一个匀速直线运动问题，速度\*时间=路程

```
#include <bits/stdc++.h>
using namespace std;
typedef long long LL;
int main(){
    int d, t, s;
    cin>>d>>t>>s;
    s * t >= d ? cout<<"Yes" : cout<<"No";
}
```

## B - Substring

题意：将t变为s的子串需要改动的最少字符。

分类：搜索

由于长度较小，可以直接暴力搜索子串的可能位置

```
#include <bits/stdc++.h>
using namespace std;
typedef long long LL;
int main(){
    string s, t;
    cin>>s;
    cin>>t;
    int ls = s.length();
    int lt = t.length();
    int ans = INT32_MAX;
    for (int i = 0; i + lt <= ls; ++i) {
        int tot = 0;
        for (int j = 0; j < lt; ++j) {
            if(t[j] != s[i + j]){
                tot++;
            }
        }
        ans = min(tot, ans);
    }
    cout<<ans;
}
```

```
}
```

## C - Sum of product of pairs

题意：数列中任意两数乘积之和。

分类：前缀和

每个数乘当前位置之前的前缀和，再求和即可

```
#include <bits/stdc++.h>
using namespace std;
typedef long long LL;
const int mod = 1e9+7;
int a[300000];
LL s[300000];
int main(){
    int n;
    cin>>n;
    LL ss = 0;
    for (int k = 0; k < n; ++k) {
        scanf("%d", a + k);
        ss += a[k];
        ss %= mod;
        s[k] = ss;
    }
    LL ans = 0;
    for (int i = 1; i < n; ++i) {
        ans += s[i-1] * a[i];
        ans %= mod;
    }
    cout<<ans;
}
```

## D - Friends

题意：给定一些朋友关系信息。如果a-b□a-c是朋友那么a-c也是朋友。求至少要分成多少组使每组中不存在一对朋友。

分类：并查集

相当于求图的最大连通块包含的点数，用并查集处理即可。

发现函数名命名为find居然会RE

```
#include <bits/stdc++.h>
using namespace std;
```

```
typedef long long LL;
int f[300000] = {};
int cnt[300000] = {};
int ff(int x){
    if(f[x] == x) return x;
    return f[x] = ff(f[x]);
}
void link(int x, int y){
    f[ff(y)] = ff(x);
}
int main(){
    int n;
    cin>>n;
    for (int j = 0; j <= n; ++j) {
        f[j] = j;
    }
    int m;
    cin>>m;
    for (int i = 0; i < m; ++i) {
        int p, q;
        scanf("%d%d", &p, &q);
        link(p, q);
    }
    for (int i = 1; i <= n; ++i) {
        cnt[ff(i)]++;
    }

    int ans = 0;
    for (int i = 1; i <= n; ++i) {
        ans = max(ans, cnt[i]);
    }

    cout<<ans;
}
```

## E - Coprime

判断一组数是否两两最大公约数为1、整体最大公约数为1。

整体最大公约数逐个求gcd即可。两两最大公约数为1可以在筛素数的同时统计出是否有两个数包含同一个素因子来判断，不过要注意两个数相等的情况。

```
#include <bits/stdc++.h>

using namespace std;
typedef long long LL;

int gcd(int a, int b) {
    if (b == 0) return a;
```

```
        return gcd(b, a % b);
    }

    const int MAXN = 1000005;
    int a[MAXN];
    int p[MAXN] = {};
    set<int> fac;

    bool notprime[MAXN];
    bool gg = false;
    void init() {
        memset(notprime, false, sizeof(notprime));
        notprime[0] = notprime[1] = true;
        for (int i = 2; i < MAXN; i++) {
            if (!notprime[i]) {
                int cnt = 0;
                if(p[i])cnt++;
                //if (i > MAXN / i)continue;
                for (int j = i * 2; j < MAXN; j += i) {
                    notprime[j] = true;
                    if(p[j]) cnt++;
                    if(cnt >= 2) {
                        gg = true;
                        return;
                    }
                }
            }
        }
    }

    int main() {
        int n;
        cin >> n;
        int gcda = 0;
        init();
        for (int i = 0; i < n; ++i) {
            scanf("%d", a + i);
            if (gcda == 0)gcda = a[i];
            else gcda = gcd(gcda, a[i]);
            if(p[a[i]] && a[i] != 1){
                gg = true;
            }
            p[a[i]] = true;
        }
        if(!gg)init();

        if(!gg){
```

```
    cout<<"pairwise coprime";  
  }  
  else if(gcda == 1){  
    cout<<"setwise coprime";  
  }else cout<<"not coprime";  
}
```

From:

<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:

[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:too\\_low:abc177hj&rev=1599211643](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:too_low:abc177hj&rev=1599211643) 

Last update: **2020/09/04 17:27**