Common Palindrome

```cpp
//UVa - 12473 - Common Palindrome
string A,B;
int dp[60][60][60][60];


int call(int al,int ar,int bl,int br) {
    if( al>ar || bl>br ) return 0;
    if(dp[al][ar][bl][br]!=-1)
        return dp[al][ar][bl][br];

    if(al == ar || bl == br) {
        if(A[al]==B[bl] and A[ar]==B[br] and A[al]==A[ar] and B[bl]==B[br]) {
            return 1;
        } else if(al == ar and bl!=br) {
            return dp[al][ar][bl][br] =
max(call(al,ar,bl+1,br),call(al,ar,bl,br-1)) ;
        } else if(bl == br and al!=ar) {
            return dp[al][ar][bl][br] = max(call(al+1,ar,bl,br),call(al,ar-
1,bl,br)) ;
        } else {
            return 0;
        }
    }
    if(A[al]==B[bl] and A[ar]==B[br] and A[al]==A[ar] and B[bl]==B[br])
        return 2 + call(al+1,ar-1,bl+1,br-1);

    int p = max(call(al+1,ar,bl,br),max(call(al,ar-
1,bl,br),max(call(al,ar,bl+1,br),call(al,ar,bl,br-1))));
    return dp[al][ar][bl][br] = p;

}

int main() {
    READ("in.txt");
    int tc,cas=1;
    cin>>tc;
    while(tc--) {
        pf("Case %d: ",cas++);
        memdp(dp);
        cin>>A>>B;
        int ans = call(0,A.size()-1,0,B.size()-1);
        pf("%d\n",ans);
    }
    return 0;
}
```

Just Make A Wish

```cpp
//UVa - 12619 - Just Make a wish
struct Euclid {
    i64 x, y, d;
    Euclid() {}
    Euclid( i64 xx, i64 yy, i64 dd ) { x = xx, y = yy, d = dd; }
};
i64 gcd( i64 a, i64 b ) { return !b ? a : gcd ( b, a % b ); }
Euclid egcd( i64 a, i64 b ) { // Input a, b; Output x, y, d; ax + by = d, d =
gcd(a,b)
    if( !b ) return Euclid ( 1, 0, a );
    Euclid r = egcd ( b, a % b );
    return Euclid( r.y, r.x - a / b * r.y, r.d );
}
i64 modularInverse( i64 a, i64 n ) { // given a and n, returns x, ax mod n = 1
    Euclid t = egcd( a, n );
    if( t.d > 1 ) return 0;
    i64 r = t.x % n;
    return r < 0 ? r + n : r;
}

void prim()
{
  int i, j, sqrtN;

  sqrtN = int( sqrt( N ) );
  for( i = 2; i <= sqrtN; i += 1 )
    if( status[i] == 0 )
        for( j = 2*i; j <= N; j += i )
            status[j] = 1;

  prime[0]=2;
  for( i = 3,j=1; i <= N; i += 2 )
    if( status[i] == 0 )
        prime[j++]=i;
    //pf("%d\n",prime[j-1]);
}

void divisor(int n){
    bool neg = false;
    if(n<0) {
        n*=-1;
        neg = true;
    }
    int N = (int)sqrt(n);
    for(int i=0;prime[i]<=N;i++){
        int count=1;
        if(n%prime[i]==0){
            i64 inv = modularInverse(DivisorCounter[ prime[i] ]+1,mod);
            ans = ( ans%mod * inv%mod )%mod;
        }
        while(n%prime[i]==0){
            n/=prime[i];
```

```
            count++;
            if(neg) DivisorCounter[ prime[i] ]--;
            else DivisorCounter[ prime[i] ]++;
        }
        if(count>1){
            N = (int)sqrt(n);
            ans=(ans%mod * (DivisorCounter[ prime[i] ]+ 1)%mod)%mod;
        }
    }

    if(n!=1){
        i64 inv = modularInverse(DivisorCounter[ n ]+1,mod);
            ans = ( ans%mod * inv%mod )%mod;
        if(neg) DivisorCounter[ n ]--;
        else DivisorCounter[ n ]++;
        ans=(ans%mod * (DivisorCounter[ n ]+ 1)%mod)%mod;
    }
}


int main()
{
#ifndef ONLINE_JUDGE
    READ("in.txt");
#endif
    int tc,cas=1;
    cin>>tc;
    prim();
    while(tc--){
        pf("Case %d: ",cas++);
        memca(DivisorCounter);
        int d;
        SDi(d);
        int total = 0;
        rep(i,d){
            int u;
            SDi(u);
            divisor(u);
            total = ( total%mod + ans%mod )%mod;
        }
        pf("%d\n",total);
        ans = 1;
    }
    return 0;
}
```

Ahoy, Pirates!

```
//UVa - 11402
```

Subtraction Game 1

```
//http://www.codechef.com/COOK34/problems/AMSGAME1
template<class T> inline T gcd(T a,T b) {if(a<0)return gcd(-a,b);if(b<0)return
gcd(a,-b);return (b==0)?a:gcd(b,a%b);}
template<class T> inline T lcm(T a,T b) {if(a<0)return lcm(-a,b);if(b<0)return
lcm(a,-b);return a*(b/gcd(a,b));}

int main()
{
#ifndef ONLINE_JUDGE
    READ("in.txt");
#endif
    int tc;
    cin>>tc;
    while(tc--){
        int n;//,a[1010];
        SDi(n);
        int GcD,x;
        SDi(GcD);
        rep(i,n-1){
            SDi(x);
            GcD = gcd(GcD,x);
        } pf("%d\n",GcD);
    }
    return 0;
}
```

Subtraction Game 2

```
// http://www.codechef.com/COOK34/problems/AMSGAME2
```