Contents

1 C++

1.1 UVa 100: The $3n + 1$ Problem

```
#include <iostream>
#include <cmath>

using namespace std;

int gcd(int a, int b) {
	return (b == 0) ? a : gcd(b,a%b);
}

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

int collatz(int m) {
	int count = 1;
	while (m != 1) {
		if (m%2 == 1) m = 3*m+1;
		else m = m/2;
		count++;
	}
	return count;
}

int main() {
	int m,n,max,temp;
    int mOriginal,nOriginal;
    int i;
	while (cin >> m >> n) {
		mOriginal = m;
		nOriginal = n;
		if (m > n) {
			temp = m;
			m = n;
			n = temp;
		}
		max = collatz(m);
		for (i=m+1; i<=n; i++) {
			temp = collatz(i);
			if (temp > max)
			max = temp;
		}
		cout << mOriginal << " " << nOriginal << " " << max << endl;
	}
	return 0;
}
```

1.2 UVa 102: Ecological Bin Packing

```
#include <iostream>
#include <algorithm>
#include <map>

using namespace std;
```

```
int main() {
    char glass[3] = {'B','C','G'};
    map<int, map<char,int> > bins;

    while (cin >> bins[0]['B'] >> bins[0]['G'] >> bins[0]['C'] >> bins[1]['B'] >> bins[1]['G']↩
         >> bins[1]['C'] >> bins[2]['B'] >> bins[2]['G'] >> bins[2]['C']) {
        int mn = bins[1][glass[0]] + bins[2][glass[0]] + bins[0][glass[1]] + bins[2][glass[1]] ↩
            + bins[0][glass[2]] + bins[1][glass[2]];
        char gl[3] = {'B','C','G'};
        do {
            int b0 = bins[1][glass[0]] + bins[2][glass[0]];
            int b1 = bins[0][glass[1]] + bins[2][glass[1]];
            int b2 = bins[0][glass[2]] + bins[1][glass[2]];

            if (mn > b0+b1+b2) {
                mn = b0+b1+b2;
                for (int i = 0; i < 3; i++) gl[i] = glass[i];
            }
        } while (next_permutation(glass, glass+3));

        cout << gl[0] << gl[1] << gl[2] << "␣" << mn << endl;
    }
}
```

1.3 UVa 109: SCUD Busters

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

using namespace std;
const double EPS = 1e-7;

struct Point { double x,y; };

bool cmp(Point a, Point b) {
    if (fabs(a.x - b.x) < EPS) return a.y < b.y;
    else return a.x < b.x;
}

double cross(Point a, Point b) {
    return a.x * b.y - a.y * b.x;
} // CCW: (+) CW: (-)
double cross(Point a, Point b, Point c) {
    return cross(a, b) + cross(b, c) + cross(c, a);
}

vector<Point> CH(vector<Point> &p) {
    int n = p.size(), k = 0;
    if (n <= 1) return p;
    sort(p.begin(), p.end(), cmp);
    vector<Point> h(2*n);
    for (int i = 0; i < n; h[k++] = p[i++])
        while (k >= 2 && cross(h[k-2], h[k-1], p[i]) > -EPS)
            --k;
    for (int i = n-2, t = k; i >= 0; h[k++] = p[i--])
        while (k > t && cross(h[k-2], h[k-1], p[i]) > -EPS)
            --k;
    k -= 1 + ((h[0].x == h[1].x && h[0].y == h[1].y) ? 1 : 0);
    h.resize(k);
    return h;
```

```
}

bool in_poly(vector<Point> &p, Point &q) {
    bool in = false; int n = p.size();
    for (int i = 0, j = n-1; i < n; j = i++)
        in ^= (((p[i].y>q.y) != (p[j].y>q.y)) && q.x < (p[j].x-p[i].x)*(q.y-p[i].y)/(p[j].y-p[i↩
            ].y)+p[i].x);
    return in;
}

double area(vector<Point> &p) {
    int n = p.size(); double a = 0;
    for (int i = 0, j = n-1; i < n; j = i++)
        a += cross(p[i], p[j]);
    return abs(a)/2;
}

int main() {
    vector< vector<Point> > kingdoms, chs;
    int n;
    while (cin >> n && n != -1) {
        vector<Point> pts(n);
        for (int i = 0; i < n; i++) cin >> pts[i].x >> pts[i].y;
        kingdoms.push_back(pts);
        chs.push_back(CH(pts));
    }
    Point q; double tot = 0;
    bool out[chs.size()];
    for (int i = 0; i < chs.size(); i++) out[i] = false;
    while (cin >> q.x >> q.y) {
        for (int i = 0; i < chs.size(); i++) {
            if (in_poly(chs[i], q)) out[i] = true;
        }
    }
    for (int i = 0; i < chs.size(); i++) {
        if (out[i]) tot += area(chs[i]);
    }
    printf("%.2f\n", tot);
}
```

1.4 UVa 108: Maximum Sum

```
#include <iostream>

using namespace std;

int a[110][110];

int kadane2D(int sz) {
    int n = sz, m = sz;
    int s[n][m];
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            s[i][j] = a[i][j] + (i==0 ? 0 : s[i-1][j]);
        }
    }
    int mx = -100000;
    for (int k = 0; k < n; ++k) {
        for (int i = 0; i + k < n; ++i) {
```

```
            int sum = 0;
            for (int j = 0; j < m; ++j) {
                sum += s[i+k][j] - (i==0 ? 0 : s[i-1][j]);
                if (mx < sum) mx = sum;
                if (sum < 0) sum = 0;
            }
        }
    }
    return mx;
}

int main() {
    int sz;
    while (cin >> sz) {
        int grid[sz][sz];
        for (int i = 0; i < sz; i++) {
            for (int j = 0; j < sz; j++) {
                cin >> a[i][j];
            }
        }
        int sum = kadane2D(sz);
        cout << sum << endl;
    }
}
```

1.5 UVa 113: Power of Cryptography

```
#include <iostream>
#include <cstdio>
#include <cmath>

using namespace std;

int main() {
        int n; double p;
        while (cin >> n >> p) {
                printf("%.0f\n",exp(log(p)/n));
        }
}
```

1.6 UVa 119: Greedy Gift Givers

```
#include <iostream>
#include <map>

using namespace std;

int main() {
    int n, c = 0;
    while (cin >> n) {
        map<string, int> net;
        string names[n];
        for (int i = 0; i < n; i++) {
            string p; cin >> p;
            net[p] = 0;
            names[i] = p;
        }
        for (int i = 0; i < n; i++) {
            string p; int a, k; cin >> p >> a >> k;
```

```
            if (k > 0) {
                int give = a/k, extra = a - k*give;
                net[p] -= a;
                for (int j = 0; j < k; j++) {
                    string q; cin >> q;
                    net[q] += give;
                }
                net[p] += extra;
            }
        }
        if (++c > 1) cout << endl;
        for (int i = 0; i < n; i++) {
            cout << names[i] << " " << net[names[i]] << endl;
        }
    }
}
```

1.7 UVa 136: Ugly Numbers

```
#include <iostream>
#include <cmath>
#include <vector>

using namespace std;

vector<int> ugly;

void build() {
    int i2 = 0, i3 = 0, i5 = 0;
        ugly.push_back(1);
        for (int i = 1; i < 1501; i++) {
                int ug = min(min(ugly[i2]*2,ugly[i3]*3),ugly[i5]*5);
                ugly.push_back(ug);
                if (ugly[i]%2 == 0) i2++;
                if (ugly[i]%3 == 0) i3++;
                if (ugly[i]%5 == 0) i5++;
    }
}

int main() {
    build();
    cout << "The 1500'th ugly number is " << ugly[1499] << "." << endl;
}
```

1.8 UVa 146: ID Codes

```
#include <iostream>
#include <algorithm>
#include <string>

using namespace std;

int main() {
        string inp;
        while (getline(cin, inp)) {
                if (inp.compare("#") == 0) break;
                if (next_permutation(inp.begin(), inp.end())) {
                        cout << inp << endl;
                } else cout << "No Successor" << endl;
```

```
        }
}
```

1.9 UVa 151: Power Crisis

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

using namespace std;

int main() {
    int dp[101][101];
    for (int i = 1; i < 101; i++) {
        for (int j = 1; j < 101; j++) {
            if (i == 1) dp[i][j] = 1;
            else dp[i][j] = (dp[i-1][j]+j-1)%i+1;
        }
    }
    int n;
    while (cin >> n && n) {
        for (int m = 1; m <= n; m++) {
            cerr << "m␣=␣" << m <<":␣" << dp[n][m] << endl;
            if (dp[n][m] == 13) {
                cout << m << endl;
                break;
            }
        }
    }
}
```

1.10 UVa 160: Factors and Factorials

```
#include <bits/stdc++.h>
#define pb push_back

using namespace std;
typedef vector<int> vi;
typedef long long ll;
const int N = 1e6;

bool isprime[N];
vi primes;
void sieve() {
    isprime[2] = true; primes.pb(2);
    for (int i = 3; i < N; i += 2) isprime[i] = true;
    for (int i = 3; i < N; i += 2) {
        if (isprime[i]) {
            primes.pb(i);
            if ((ll)i*i >= N) continue;
            for (int j = i*i; j < N; j += i) isprime[j] = false;
        }
    }
}

void factorize(int n, map<int, int> &factors) {
    factors.clear();
    for (int i = 0; i < primes.size(); i++) {
        int p = primes[i];
        if (n/p == 0) break;
        while (n/p > 0) {
```

```
            factors[primes[i]] += n/p;
            p *= primes[i];
        }
    }
}

int main() {
    int n;
    while (cin >> n && n) {
        map<int, int> factors;
        sieve();
        factorize(n, factors);
        cout << setw(3) << n << "!␣=";
        int c = 1;
        for (map<int, int>::iterator it = factors.begin(); it != factors.end(); it++, c++) {
            if (c%15 == 1 && c > 15) cout << "␣␣␣␣␣␣";
            cout << setw(3) << it->second;
            if (c%15 == 0 && factors.size() > 15) cout << endl;
        }
        cout << endl;
    }
}
```

1.11 UVa 190: Circle Through Three Points

```
#include <iostream>
#include <cstdio>
#include <cmath>

using namespace std;
const double eps = 1e-6;

struct pt {
    double x;
    double y;
};

double dist(pt a, pt b) {
    return sqrt(pow(a.x-b.x, 2)+pow(a.y-b.y, 2));
}

pt circumcenter(pt l, pt m, pt n) {
    pt k;
    double xx = ((m.y-n.y)*(l.x*l.x+(l.y-m.y)*(l.y-n.y))+m.x*m.x*(n.y-l.y)+n.x*n.x*(l.y-m.y))↩
        /(2*(l.x*(m.y-n.y) + m.x*(n.y-l.y) + n.x*(l.y-m.y)));
    double yy = (2*(m.x-n.x)*(-l.x*l.x+n.x*n.x-l.y*l.y+n.y*n.y)+2*(l.x-n.x)*(m.x*m.x-n.x*n.x+m↩
        .y*m.y-n.y*n.y))/(4*(l.x*(m.y-n.y) + m.x*(n.y-l.y) + n.x*(l.y-m.y)));
    k.x = xx; k.y = yy;
    return k;
}

int main() {
    pt a, b, c;
    while (cin >> a.x >> a.y >> b.x >> b.y >> c.x >> c.y) {
        pt R = circumcenter(a, b, c);
        double r = dist(R, a);
        if (fabs(R.x) < eps) printf("x^2␣");
        else {
            printf("(x␣");
```

```
            if (R.x < 0) printf("+ %.3f)^2 ", fabs(R.x));
            else printf("- %.3f)^2 ", R.x);
        }
        if (fabs(R.y) < eps) printf("+ y^2 ");
        else {
            printf("+ (y ");
            if (R.y < 0) printf("+ %.3f)^2 ", fabs(R.y));
            else printf("- %.3f)^2 ", R.y);
        }
        printf("= %.3f^2\n", r);
        printf("x^2 + y^2 ");
        if (!(fabs(R.x) < eps)) {
            if (R.x < 0) printf("+ %.3fx ", fabs(2*R.x));
            else printf("- %.3fx ", 2*R.x);
        }
        if (!(fabs(R.y) < eps)) {
            if (R.y < 0) printf("+ %.3fy ", fabs(2*R.y));
            else printf("- %.3fy ", 2*R.y);
        }
        double e = r*r-R.x*R.x-R.y*R.y;
        if (!(fabs(e) < eps)) {
            if (e < 0) printf("+ %.3f ", fabs(e));
            else printf("- %.3f ", e);
        }
        printf("= 0\n\n");
    }
}
```

1.12 UVa 195: Anagram

```
#include <iostream>
#include <algorithm>
#include <string>
#include <cctype>

using namespace std;

bool cmp(char a, char b) {
    if (tolower(a) == tolower(b)) return a<b;
    else return tolower(a) < tolower(b);
}

int main() {
    int t; cin >> t;
    while (t--) {
        string s; cin >> s;
        sort(s.begin(), s.end(), cmp);
        do {
            cout << s << endl;
        } while (next_permutation(s.begin(), s.end(), cmp));
    }
}
```

1.13 UVa 270: Lining Up

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

using namespace std;
```

```
struct Point{
    double x, y;
    Point(double _x, double _y): x(_x), y(_y) {};
};

int main() {
    int t; cin >> t; cin.ignore();
    string s; getline(cin, s);
    while (t--) {
        vector<Point> pts;
        while(getline(cin, s) && s != "") {
            istringstream is(s);
            double x, y; is >> x >> y;
            pts.push_back(Point(x, y));
        }
        int mx = 0;
        for (int i = 0; i < pts.size(); i++) {
            int q = 0, v = 0, h = 0, r = 0;
            map<pair<int,int>, int> m;
            for (int j = i+1; j < pts.size(); j++) {
                int x = pts[i].x-pts[j].x, y = pts[i].y-pts[j].y;
                if (x == 0 && y == 0) q++;
                else if (x == 0) h++;
                else if (y == 0) v++;
                else {
                    int g = __gcd(x, y); x /= g; y /= g;
                    if (x < 0) {
                        x *= -1; y *= -1;
                    }
                    pair<int, int> ff = make_pair(x, y);
                    if (m.find(ff) == m.end()) m[ff] = 1;
                    else m[ff]++;
                    r = max(r, m[ff]);
                }
                mx = max(mx, max(v+q+1, max(h+q+1, r+q+1)));
            }
        }
        cout << mx << endl;
        if (t) cout << endl;
    }
}
```

1.14 UVa 272: TEXQuotes

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string s;
    bool op = true;
    while (getline(cin, s)) {
        for (int i = 0; i < s.length(); i++) {
            if (s[i] == '"') {
                cout << ((op) ? "``" : "''");
                op = !op;
            } else cout << s[i];
        }
```

```
        cout << endl;
    }
}
```

1.15 UVa 280: Vertex

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

using namespace std;
typedef map< int, map<int,int> > graph;

void dfs(graph &G, int v, vector<int> &visited) {
    for (map<int,int>::iterator w = G[v].begin(); w != G[v].end(); w++) {
        if (visited[w->first] == 0) {
            visited[w->first] = 1;
            dfs(G, w->first, visited);
        }
    }
}

int main() {
    int V;
    while (cin >> V && V) {
        graph G;
        int start;
        while (cin >> start && start) {
            int end;
            while (cin >> end && end) {
                G[start][end] = 1;
            }
        }
        int sv; cin >> sv;
        while (sv--) {
            int u; cin >> u;
            vector<int> visited(V+1, 0);
            dfs(G, u, visited);
            vector<int> no;
            for (int i = 1; i <= V; i++) {
                if (!visited[i]) no.push_back(i);
            }
            cout << no.size();
            for (int k = 0; k < no.size(); k++)
                cout << "␣" << no[k];

            cout << endl;
        }
    }
}
```

1.16 UVa 291: The House of Santa Claus

```
#include <iostream>

using namespace std;

int main() {
        int ways[44] = {123153452,123154352,123451352,123453152,123513452,
                                    123543152,125134532,125135432,125315432,125345132,
                                    125431532,125435132,132153452,132154352,132534512,
```

```
                                    132543512,134512352,134512532,134521532,134523512,
                                    134532152,134532512,135123452,135125432,135215432,
                                    135234512,135432152,135432512,152134532,152135432,
                                    152345312,152354312,153123452,153125432,153213452,
                                    153254312,153452132,153452312,154312352,154312532,
                                    154321352,154325312,154352132,154352312};
        for (int i = 0; i < 44; i++) {
                cout << ways[i] << endl;
        }
}
```

1.17 UVa 299: Train Swapping

```
#include <iostream>
#include <algorithm>

using namespace std;
int l[60];

int bubble(int n) {
    int j = 0, k = 0;
    bool swapped = true;
    while (swapped) {
        swapped = false;
        j++;
        for (int i = 0; i < n-j; i++) {
            if (l[i] > l[i+1]) {
                swap(l[i], l[i+1]);
                k++;
                swapped = true;
            }
        }
    }
    return k;
}

int main() {
    int t; cin >> t;
    while (t--) {
        int n; cin >> n;
        for (int i = 0; i < n; i++) cin >> l[i];
        cout << "Optimal␣train␣swapping␣takes␣" << bubble(n) << "␣swaps." << endl;
    }
}
```

1.18 UVa 357: Let Me Count the Ways

```
#include <iostream>
#include <cstring>

using namespace std;
typedef long long ll;

ll count(int S[], int m, int n) {
        ll memo[n+1];
        memset(memo, 0LL, sizeof memo);
        memo[0] = 1;
        for (ll i = 0; i < m; i++)
                for (ll j = S[i]; j <= n; j++)
```

```
                        memo[j] += memo[j-S[i]];

        return memo[n];
}

int main() {
        int k, coins[5] = {1,5,10,25,50};
        while (cin >> k) {
                ll n = count(coins, 5, k);
                if (n == 1) cout << "There␣is␣only␣1␣way␣";
                else cout << "There␣are␣" << n << "␣ways␣";
                cout << "to␣produce␣" << k << "␣cents␣change." << endl;
        }
}
```

1.19 UVa 369: Combinations

```
#include <iostream>
#include <algorithm>

using namespace std;

int binom(int n, int k) {
    int c[n+1][k+1], i, j;
    for (i = 0; i <= n; i++) {
        for (j = 0; j <= min(i,k); j++) {
            if (j == 0 || j == i) c[i][j] = 1;
            else c[i][j] = c[i-1][j-1] + c[i-1][j];
        }
    }
    return c[n][k];
}

int main() {
    int n, m;
    while (cin >> n >> m) {
        if (n == 0 && m == 0) break;
        cout << n << "␣things␣taken␣" << m << "␣at␣a␣time␣is␣" << binom(n,m) << "␣exactly." << ↩
            endl;
    }
}
```

1.20 UVa 374: Big Mod

```
#include <iostream>
#include <cmath>

using namespace std;
typedef long long ll;

int modpow(ll base, ll pw, int mod) {
        ll res = 1;
        base = base%mod;
        while (pw > 0) {
                if (pw%2 == 1) {
                        res = (res*base)%mod;
                }
                pw = pw >> 1;
                base = (base*base)%mod;
```

```
        }
        return res;
}

int main() {
        ll b; ll p; int m;
        while (cin >> b >> p >> m) {
                cout << modpow(b,p,m) << endl;
        }
}
```

1.21 UVa 378: Intersecting Lines

```
#include <iostream>
#include <cmath>
#include <iomanip>
#include <vector>

using namespace std;

int main() {
        int t; cin >> t;
        cout << "INTERSECTING␣LINES␣OUTPUT" << endl;
        while (t--) {
                int ax1, ay1, ax2, ay2; cin >> ax1 >> ay1 >> ax2 >> ay2;
                int bx1, by1, bx2, by2; cin >> bx1 >> by1 >> bx2 >> by2;
                int s[2][3];
                s[0][0] = ay1-ay2; s[0][1] = ax2-ax1; s[0][2] = ax2*ay1-ax1*ay2;
                s[1][0] = by1-by2; s[1][1] = bx2-bx1; s[1][2] = bx2*by1-bx1*by2;

                int det = s[0][0]*s[1][1]-s[0][1]*s[1][0];
                int x = (s[0][2]*s[1][1]-s[0][1]*s[1][2]);
                int y = (s[0][0]*s[1][2]-s[0][2]*s[1][0]);

                if (det == 0) {
                        if (x == 0 && y == 0) cout << "LINE" << endl;
                        else cout << "NONE" << endl;
                } else {
                        cout << fixed << setprecision(2);
                        cout << "POINT␣" << (double)x/det << "␣" << (double)y/det << endl;
                }
        }
        cout << "END␣OF␣OUTPUT" << endl;
}
```

1.22 UVa 382: Perfection

```
#include <iostream>
#include <iomanip>

using namespace std;

int divsum(int n) {
    int sum = 0;
    for (int i = 1; i < n/2+1; i++) {
        if (n%i == 0) sum += i;
    }
    return sum;
}
```

```
int main() {
    int n;
    cout << "PERFECTION␣OUTPUT" << endl;
    while (cin >> n) {
        if (n == 0) {
            cout << "END␣OF␣OUTPUT" << endl;
            break;
        }
        cout << setw(5) << n << "␣␣";
        if (n < divsum(n)) cout << "ABUNDANT" << endl;
        else if (n > divsum(n)) cout << "DEFICIENT" << endl;
        else cout << "PERFECT" << endl;
    }
}
```

1.23 UVa 386: Perfect Cubes

```
#include <cstdio>
#include <cmath>

using namespace std;
typedef long long ll;

int main() {
    ll a, b, c, d, ia, ib, ic, id;
    for (ll ia = 6; ia <= 200; ia++) {
        a = ia*ia*ia;
        for (ll id = 2; id < ia; id++) {
            d = id*id*id;
            for (ll ic = id+1; ic < ia; ic++) {
                c = ic*ic*ic;
                for (ll ib = ic+1; ib < ia; ib++) {
                    b = ib*ib*ib;
                    if (a == b+c+d) printf("Cube␣=␣%lld,␣Triple␣=␣(%lld,%lld,%lld)\n",ia,id,ic,↩
                        ib);
                }
            }
        }
    }
}
```

1.24 UVa 429: Word Transformation

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

using namespace std;
typedef map< int,vector<int> > graph;

template<class T> T poll(queue<T> &q) {T a=q.front();q.pop();return a;}
inline int min(int a, int b, int c) {return min(a, min(b, c));}

vector<string> split(const string &s) {
    vector<string> vs; string ss;
    istringstream iss(s);
    while (iss >> ss) vs.push_back(ss);
    return vs;
}
```

```
int bfs(graph &G, int src, int dst, int V) {
    int color[V], dist[V];
    for (int i = 0; i < V; i++) color[i] = 0;
    color[src] = 1; dist[src] = 0;
    queue<int> q;
    q.push(src);
    while (!q.empty()) {
        int u = poll(q);
        for (vector<int>::iterator v = G[u].begin(); v != G[u].end(); v++) {
            if (color[*v] == 0) {
                dist[*v] = dist[u] + 1;
                color[*v] = 1;
                q.push(*v);
            }
        }
    }
    return dist[dst];
}

int edit(string s1, string s2) {
    int m = s1.length(), n = s2.length();
    int memo[m+1][n+1];
    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0) memo[i][j] = j;
            else if (j == 0) memo[i][j] = i;
            else if (s1[i-1] == s2[j-1]) memo[i][j] = memo[i-1][j-1];
            else memo[i][j] = 1 + min(memo[i][j-1], memo[i-1][j], memo[i-1][j-1]);
        }
    }
    return memo[m][n];
}

int main() {
    int t; cin >> t;
    while (t--) {
        vector<string> dict; map<string,int> idx;
        string w; int id = 0;
        while (getline(cin, w) && w.compare("*")) {
            dict.push_back(w);
            idx[w] = id++;
        }
        graph G;
        for (int i = 0; i < dict.size(); i++) {
            for (int j = 0; j < dict.size(); j++) {
                if (edit(dict[i], dict[j]) == 1) G[i].push_back(j);
            }
        }
        string wp;
        while (getline(cin, wp) && wp != "") {
            vector<string> ft = split(wp);
            string from = ft[0], to = ft[1];
            int dis = bfs(G, idx[from], idx[to], dict.size());
            cout << from << "␣" << to << "␣" << dis << endl;
        }
        if (t > 0) cout << endl;
    }
}
```

1.25 UVa 438: The Circumference of the Circle

```
#include <iostream>
#include <cstdio>
#include <cmath>

using namespace std;
const double pi = 3.14159265358973;

struct pt {
    double x;
    double y;
};

double dist(pt a, pt b) {
    return sqrt(pow(a.x-b.x, 2)+pow(a.y-b.y, 2));
}

pt circumcenter(pt l, pt m, pt n) {
    pt k;
    double xx = ((m.y-n.y)*(l.x*l.x+(l.y-m.y)*(l.y-n.y))+m.x*m.x*(n.y-l.y)+n.x*n.x*(l.y-m.y))↩
        /(2*(l.x*(m.y-n.y) + m.x*(n.y-l.y) + n.x*(l.y-m.y)));
    double yy = (2*(m.x-n.x)*(-l.x*l.x+n.x*n.x-l.y*l.y+n.y*n.y)+2*(l.x-n.x)*(m.x*m.x-n.x*n.x+m↩
        .y*m.y-n.y*n.y))/(4*(l.x*(m.y-n.y) + m.x*(n.y-l.y) + n.x*(l.y-m.y)));
    k.x = xx; k.y = yy;
    return k;
}

int main() {
    pt a, b, c;
    while (cin >> a.x >> a.y >> b.x >> b.y >> c.x >> c.y) {
        pt R = circumcenter(a, b, c);
        double r = dist(R, a);
        double p = 2*pi*r;
        printf("%.2f\n", p);
    }
}
```

1.26 UVa 441: Lotto

Note! Solution not optimal (2.279s runtime with 3s time limit)

```
#include <iostream>
#include <algorithm>

using namespace std;

bool sorted(int a[]) {
        bool isit = true;
        for (int i = 0; i < 5; i++) {
                if (a[i] > a[i+1]) {
                        isit = false;
                        break;
                }
        }
        return isit;
}

bool arreq(int a[], int b[]) {
```

```
        bool isit = true;
        for (int i = 0; i < 6; i++) {
                if (a[i] != b[i]) {
                        isit = false;
                        break;
                }
        }
        return isit;
}

int main() {
        int n; cin >> n;
        while (n) {
                int k[n], kk[n];
                for (int i = 0; i < n; i++) {
                        cin >> k[i];
                }
                do {
                        if (sorted(k) && !arreq(k, kk)) {
                                for (int i = 0; i < 6; i++) {
                                        cout << k[i];
                                        if (i < 5) cout << "␣";
                                        kk[i] = k[i];
                                }
                                cout << endl;
                        }
                } while (next_permutation(k, k+n));
                cin >> n;
                if (n != 0) cout << endl;
        }
}
```

1.27 UVa 443: Humble Numbers

```
#include <iostream>
#include <cmath>
#include <vector>

using namespace std;

vector<int> hum;

void build() {
    int i2 = 0, i3 = 0, i5 = 0, i7 = 0;
        hum.push_back(1);
        for (int i = 1; i < 5842; i++) {
                int hm = min(min(hum[i2]*2,hum[i3]*3),min(hum[i5]*5,hum[i7]*7));
                hum.push_back(hm);
                if (hum[i]%2 == 0) i2++;
                if (hum[i]%3 == 0) i3++;
                if (hum[i]%5 == 0) i5++;
                if (hum[i]%7 == 0) i7++;
    }
}

int main() {
    build();
```

```
    int n; cin >> n;
    while (n) {
        cout << "The␣" << n;
        switch (n%10) {
            case 1:
            if (n%100 != 11) cout << "st␣";
            else cout << "th␣";
            break;
            case 2:
            if (n%100 != 12) cout << "nd␣";
            else cout << "th␣";
            break;
            case 3:
            if (n%100 != 13) cout << "rd␣";
            else cout << "th␣";
            break;
            default:
            cout << "th␣";
            break;
        }
        cout << "humble␣number␣is␣" << hum[n-1] << "." << endl;
        cin >> n;
    }
}
```

1.28 UVa 446: Kibbles "n" Bits "n" Bits "n" Bits

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

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int a, b; string op;
        cin >> hex >> a >> op >> b;
        bitset<13> b1(a), b2(b);
        cout << b1.to_string() << "␣" << op << "␣" << b2.to_string() << "␣=␣";
        if (op.compare("+") == 0) cout << a+b << endl;
        else cout << a-b << endl;
    }
}
```

1.29 UVa 457: Linear Cellular Automata

```
#include <iostream>

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int dish[50][40], dna[10];
        for (int i = 0; i < 10; i++) cin >> dna[i];
        dish[0][19] = 1;
        for (int j = 1; j < 50; j++) {
            for (int k = 0; k < 40; k++) {
                if (k == 0) dish[j][k] = dna[dish[j-1][k]+dish[j-1][k+1]];
                else if (k == 39) dish[j][k] = dna[dish[j-1][k]+dish[j-1][k-1]];
```

```
                else dish[j][k] = dna[dish[j-1][k-1]+dish[j-1][k]+dish[j-1][k+1]];
            }
        }
        for (int j = 0; j < 50; j++) {
            for (int k = 0; k < 40; k++) {
                switch (dish[j][k]) {
                    case 0:
                        cout << "␣";
                        break;
                    case 1:
                        cout << ".";
                        break;
                    case 2:
                        cout << "x";
                        break;
                    case 3:
                        cout << "W";
                        break;
                }
            }
            cout << endl;
        }
        if (t > 0) cout << endl;
    }
}
```

1.30 UVa 458: The Decoder

```
#include <iostream>
#include <string>

using namespace std;

int main() {
        string inp;
        while (getline(cin,inp)) {
                for (int i = 0; i < inp.size(); i++)
                        if (inp[i] != 0) inp[i] -= 7;
                cout << inp << endl;
        }
        return 0;
}
```

1.31 UVa 460: Overlapping Rectangles

```
#include <iostream>
#include <cstdio>
#include <algorithm>

using namespace std;

int main() {
        int t; cin >> t;
        for (int i = 1; i <= t; i++) {
                int x1, y1, x2, y2;
                int xx1, yy1, xx2, yy2;
                cin >> x1 >> y1 >> x2 >> y2;
                cin >> xx1 >> yy1 >> xx2 >> yy2;
                int ix1 = max(x1,xx1);
```

```
                int iy1 = max(y1,yy1);
                int ix2 = min(x2,xx2);
                int iy2 = min(y2,yy2);
                if (ix2-ix1<=0 || iy2-iy1<=0) {
                        printf("No␣Overlap\n");
                        if (i != t) cout << endl;
                }
                else {
                        printf("%d␣%d␣%d␣%d\n", ix1, iy1, ix2, iy2);
                        if (i != t) cout << endl;
                }
        }
}
```

1.32 UVa 471: Magic Numbers

```
#include <iostream>
using namespace std;

bool is_digits_repeated(long long n)
{
  unsigned int digits = 0;
  do {
    int d = 1 << static_cast<int>(n % 10);
    if (digits & d)
      return true;
    n /= 10;
    digits |= d;
  } while (n);
  return false;
}

int main()
{
  const long long s1_max = 9876543210LL;
  int t;
  cin >> t;
  while (t--) {
    long long N;
    cin >> N;
    for (long long s2 = 1, s2_max = s1_max / N; s2 <= s2_max; s2++)
      if (!is_digits_repeated(s2)) {
        long long s1 = s2 * N;
        if (!is_digits_repeated(s1))
          cout << s1 << "␣/␣" << s2 << "␣=␣" << N << endl;
      }
    if (t)
      cout << endl;
  }
  return 0;
}
```

1.33 UVa 476: Points in Figures: Rectangles

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

using namespace std;

struct Point { double x,y; };
```

```
bool contains(Point c1, Point c2, Point a) {
    return (c1.x < a.x && a.x < c2.x) && (c2.y < a.y && a.y < c1.y);
}

int main() {
    string dims;
    vector< pair<Point,Point> > recs;
    while (getline(cin, dims) && dims != "*") {
        istringstream is(dims);
        char t; Point c1, c2;
        is >> t >> c1.x >> c1.y >> c2.x >> c2.y;
        recs.push_back(make_pair(c1, c2));
    }
    Point p; int c = 0;
    while (cin >> p.x >> p.y && p.x != 9999.9 && p.y != 9999.9) {
        bool cont = false; c++;
        for (int i = 0; i < recs.size(); i++) {
            if (contains(recs[i].first, recs[i].second, p)) {
                cont = true;
                cout << "Point␣" << c << "␣is␣contained␣in␣figure␣" << i+1 << endl;
            }
        }
        if (!cont) cout << "Point␣" << c << "␣is␣not␣contained␣in␣any␣figure" << endl;
    }
}
```

1.34 UVa 477: Points in Figures: Rectangles and Circles

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

using namespace std;

struct Point {
    double x,y;
    Point(double _x, double _y): x(_x), y(_y) {};
};

bool containsR(Point c1, Point c2, Point a) {
    return (c1.x < a.x && a.x < c2.x) && (c2.y < a.y && a.y < c1.y);
}

bool containsC(Point c, double r, Point a) {
    return (hypot(c.x-a.x, c.y-a.y) < r);
}

vector<string> parse(string &s) {
    istringstream is(s);
    string w; vector<string> v;
    while (is >> w) v.push_back(w);
    return v;
}

double pd(string &s) {
    istringstream is(s);
    double d; is >> d;
    return d;
}
```

```
int main() {
    string dim;
    vector<string> dims;
    while (getline(cin, dim) && dim != "*") {
        dims.push_back(dim);
    }
    Point p(0,0); int idx = 0;
    while (cin >> p.x >> p.y && p.x != 9999.9 && p.y != 9999.9) {
        bool cont = false; idx++;
        for (int i = 0; i < dims.size(); i++) {
            vector<string> nums = parse(dims[i]);
            if (nums[0] == "r") {
                Point c1(pd(nums[1]),pd(nums[2])), c2(pd(nums[3]),pd(nums[4]));
                if (containsR(c1, c2, p)) {
                    cont = true;
                    cout << "Point␣" << idx << "␣is␣contained␣in␣figure␣" << i+1 << endl;
                }
            } else if (nums[0] == "c") {
                Point c(pd(nums[1]),pd(nums[2])); double r = pd(nums[3]);
                if (containsC(c, r, p)) {
                    cont = true;
                    cout << "Point␣" << idx << "␣is␣contained␣in␣figure␣" << i+1 << endl;
                }
            }
        }
        if (!cont) cout << "Point␣" << idx << "␣is␣not␣contained␣in␣any␣figure" << endl;
    }

}
```

1.35 UVa 478: Points in Figures: Rectangles, Circles, and Triangles

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

using namespace std;

struct Point {
    double x,y;
    Point(double _x, double _y): x(_x), y(_y) {};
};

double cross(Point a, Point b) {
    return a.x * b.y - a.y * b.x;
} // CCW: (+) CW: (-)
double cross(Point a, Point b, Point c) {
    return cross(a, b) + cross(b, c) + cross(c, a);
}

bool containsR(Point c1, Point c2, Point a) {
    return (c1.x < a.x && a.x < c2.x) && (c2.y < a.y && a.y < c1.y);
}

bool containsC(Point c, double r, Point a) {
    return (hypot(c.x-a.x, c.y-a.y) < r);
}

bool containsT(Point p1, Point p2, Point p3, Point a) {
```

```
    return (cross(p1, p2, a) > 0 && cross(p2, p3, a) > 0 && cross(p3, p1, a) > 0) || (cross(p1↩
        , p2, a) < 0 && cross(p2, p3, a) < 0 && cross(p3, p1, a) < 0);
}

vector<string> parse(string &s) {
    istringstream is(s);
    string w; vector<string> v;
    while (is >> w) v.push_back(w);
    return v;
}

double pd(string &s) {
    istringstream is(s);
    double d; is >> d;
    return d;
}

int main() {
    string dim;
    vector<string> dims;
    while (getline(cin, dim) && dim != "*") {
        dims.push_back(dim);
    }
    Point p(0,0); int idx = 0;
    while (cin >> p.x >> p.y && p.x != 9999.9 && p.y != 9999.9) {
        bool cont = false; idx++;
        for (int i = 0; i < dims.size(); i++) {
            vector<string> nums = parse(dims[i]);
            if (nums[0] == "r") {
                Point c1(pd(nums[1]),pd(nums[2])), c2(pd(nums[3]),pd(nums[4]));
                if (containsR(c1, c2, p)) {
                    cont = true;
                    cout << "Point␣" << idx << "␣is␣contained␣in␣figure␣" << i+1 << endl;
                }
            } else if (nums[0] == "c") {
                Point c(pd(nums[1]),pd(nums[2])); double r = pd(nums[3]);
                if (containsC(c, r, p)) {
                    cont = true;
                    cout << "Point␣" << idx << "␣is␣contained␣in␣figure␣" << i+1 << endl;
                }
            } else if (nums[0] == "t") {
                Point p1(pd(nums[1]),pd(nums[2])), p2(pd(nums[3]),pd(nums[4])), p3(pd(nums[5]),↩
                    pd(nums[6]));
                if (containsT(p1, p2, p3, p)) {
                    cont = true;
                    cout << "Point␣" << idx << "␣is␣contained␣in␣figure␣" << i+1 << endl;
                }
            }
        }
        if (!cont) cout << "Point␣" << idx << "␣is␣not␣contained␣in␣any␣figure" << endl;
    }

}
```

1.36 UVa 488: Triangle Wave

```
#include <iostream>

using namespace std;
```

```
int main() {
	int t; cin >> t;
	while (t--) {
		int a, f; cin >> a >> f;
		while (f--) {
			for (int i = 1; i <= a; i++) {
				for (int j = 1; j <= i; j++) cout << i;
				cout << endl;
			}
			for (int i = a-1; i >= 1; i--) {
				for (int j = 1; j <= i; j++) cout << i;
				cout << endl;
			}
			if (t > 0 || f > 0) cout << endl;
		}
	}
}
```

1.37 UVa 494: Kindergarten Counting Game

Note! First problem solved with C++11

```
#include <iostream>
#include <string>
#include <regex>

using namespace std;

int main() {
    regex re("[A-Za-z]+");
    string words;
    while (getline(cin, words)) {
        auto cnt(distance(sregex_iterator(words.begin(), words.end(), re), sregex_iterator()));
        cout << cnt << endl;
    }
}
```

1.38 UVa 496: Simply Subsets

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

using namespace std;

vector<int> split(string &s) {
	vector<int> v; int n;
	istringstream is(s);
	while(is >> n) v.push_back(n);
	return v;
}

int intersections(vector<int> &a, vector<int> &b) {
	int n = 0;
	for (int i = 0; i < b.size(); i++) {
	bool eq = false;
		for (int j = 0; j < a.size(); j++)
			if (a[j] == b[i]) eq = true;
	if (eq) n++;
```

```
    }
        return n;
}

int main() {
        string s1;
        while (getline(cin, s1)) {
                string s2; getline(cin, s2);
                vector<int> set1 = split(s1);
                vector<int> set2 = split(s2);
                int a = set1.size();
                int b = set2.size();
                int x = intersections(set1, set2);
                if (a == 0 && b == 0) cout << "A␣equals␣B";
        else if (b == 0) cout << "B␣is␣a␣proper␣subset␣of␣A";
                else if (a == 0) cout << "A␣is␣a␣proper␣subset␣of␣B";
        else if (x == a && x == b) cout << "A␣equals␣B";
        else if (x == 0) cout << "A␣and␣B␣are␣disjoint";
                else if (x == b) cout << "B␣is␣a␣proper␣subset␣of␣A";
                else if (x == a) cout << "A␣is␣a␣proper␣subset␣of␣B";
                else cout << "I'm␣confused!";
                cout << endl;
        }
}
```

1.39 UVa 498: Polly the Polynomial

```
#include <iostream>
#include <sstream>
#include <cmath>
#include <vector>

using namespace std;

int polyeval(vector<int> coeff, int x) {
    int n = coeff.size();
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += coeff[i]*pow(x, n-i-1);
    }
    return sum;
}

int main() {
    string pl, val;
    while (getline(cin, pl) && getline(cin, val)) {
        if (pl.empty() || val.empty()) break;
        vector<int> poly;
        stringstream sp(pl);
        stringstream sv(val);

        int cc, vv;
        while (sp >> cc) poly.push_back(cc);

        while (sv >> vv) {
            int thing = sv.peek();
            int res = polyeval(poly, vv);
            cout << res;
            if (thing != char_traits<char>::eof()) cout << "␣";
```

```
        }
        cout << endl;
    }
}
```

1.40 UVa 499: What's The Frequency, Kenneth?

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

using namespace std;

int main() {
    string line; int t = 0;
    while (getline(cin, line)) {
        int asc[256], mx = 0;
        for (int i = 0; i < 256; i++) asc[i] = 0;
        for (int i = 0; i < line.length(); i++) {
            char c = line[i];
            if (isalpha(c)) mx = max(mx, ++asc[(int)c]);
        }
        for (int i = 0; i < 256; i++) {
            if (asc[i] == mx) cout << (char)i;
        }
        cout << "␣" << mx << endl;
    }
}
```

1.41 UVa 541: Error Correction

```
#include <iostream>

using namespace std;

int suml(int a[], int sz) {
        int sum = 0;
        for (int i = 0; i < sz; i++) {
                sum += a[i];
        }
        return sum;
}
int main() {
        int sz;
        cin >> sz;
        while (sz != 0) {
                int mat[sz][sz];
                int srow[sz];
                int scol[sz];
                int a,b;
                for (int i = 0; i < sz; i++) {
                        for (int j = 0; j < sz; j++) {
                                int el;
                                cin >> el;
                                mat[i][j] = el;
                        }
                        srow[i] = 0;
                        scol[i] = 0;
                }

                for (int i = 0; i < sz; i++) {
```

```
                        for (int j = 0; j < sz; j++) {
                                srow[i] += mat[i][j];
                                scol[i] += mat[j][i];
                        }
                }
                for (int i = 0; i < sz; i++) {
                        srow[i] = srow[i] % 2;
                        scol[i] = scol[i] % 2;
                }
                int sumr = suml(srow, sz);
                int sumc = suml(scol, sz);
                for (int i = 0; i < sz; i++) {
                        if (srow[i] == 1) a = i+1;
                        if (scol[i] == 1) b = i+1;
                }
                if (sumr == 0 && sumc == 0) cout << "OK" << endl;
                else if (sumr == 1 && sumc == 1) cout << "Change␣bit␣(" << a << "," << b << ")" ↩
                    << endl;
                else cout << "Corrupt" << endl;
                cin >> sz;
        }
}
```

1.42 UVa 558: Wormholes

```
#include <iostream>
#include <algorithm>
#include <map>
#include <limits>

using namespace std;
typedef map< int, map<int,int> > graph;
const int INF = numeric_limits<int>::max();

bool bf(graph G, int V, int s) {
        map<int,int> d, pred;
        for (graph::iterator v = G.begin(); v != G.end(); v++) {
                d[v->first] = 0;
                pred[v->first] = NULL;
        }
        d[s] = 0;
        for (int i = 1; i < V; i++) {
                for (graph::iterator u = G.begin(); u != G.end(); u++) {
                        for (map<int,int>::iterator v = G[u->first].begin(); v != G[u->first].↩
                            end(); v++) {
                                if (d[u->first] + G[u->first][v->first] < d[v->first]) {
                                        d[v->first] = d[u->first] + G[u->first][v->first];
                                        pred[v->first] = u->first;
                                }
                        }
                }
        }
        for (graph::iterator u = G.begin(); u != G.end(); u++) {
                for (map<int,int>::iterator v = G[u->first].begin(); v != G[u->first].end(); v↩
                    ++) {
                        if (d[u->first] + G[u->first][v->first] < d[v->first]) return true;
                }
        }
        return false;
```

```
}

int main() {
	int t; cin >> t;
	while (t--) {
		int n, m; cin >> n >> m;
		graph G;
		while (m--) {
			int a, b, w;
			cin >> a >> b >> w;
			G[a][b] = w;
		}
		bool hasnegc = bf(G, n, 0);
		cout << ((hasnegc) ? "possible" : "not possible") << endl;
	}
}
```

1.43 UVa 573: The Snail

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    int h, u, d, f;
    while (cin >> h >> u >> d >> f) {
        if (h == 0) break;
        int day = 0;
        bool done = false;
        double hh = h, uu = u, dd = d, hi = 0, ff = uu*f/100.0;
        while (!done) {
            day++;
            hi += uu;
            if (hh < hi) {
                cout << "success on day " << day << endl;
                done = true;
            }
            hi -= dd;
            if (hi < 0) {
                cout << "failure on day " << day << endl;
                done = true;
            }
            uu -= ff;
            if (uu < 0) uu = 0;
        }
    }
}
```

1.44 UVa 575: Skew Binary

```
#include <iostream>
#include <string>

using namespace std;

int skew(string s) {
    int n = 0;
    for (int i = 0; i < s.length(); i++) {
```

```
        n += (s[i]-'0') * ((1 << s.length()-i)-1);
    }
    return n;
}

int main() {
    string s;
    while (cin >> s) {
        if (s.compare("0") == 0) break;
        cout << skew(s) << endl;
    }
}
```

1.45 UVa 579: Clock Hands

```
#include <iostream>
#include <cstdio>
#include <cmath>

using namespace std;

int main() {
        int h,m;
        char dummy;
        cin >> h >> dummy >> m;
        while (h != 0 || m != 0) {
                h = h%12;
                double diff = (double)abs(0.5*(60*h-11*m));
                printf("%.3f\n",(diff > 180)?360-diff:diff);
                cin >> h >> dummy >> m;
        }
        return 0;
}
```

1.46 UVa 583: Prime Factors

```
#include <iostream>
#include <cmath>
#include <vector>

using namespace std;
typedef vector<int> vi;

vi primefacs(int n) {
    vi pfs;
    if (n < 0) {
        pfs.push_back(-1);
        n *= -1;
    }
    while (n%2 == 0) {
        pfs.push_back(2);
        n /= 2;
    }
    for (int i = 3; i <= sqrt(n); i += 2) {
        while (n%i == 0) {
            pfs.push_back(i);
            n /= i;
        }
    }
```

```
    if (n > 2) pfs.push_back(n);
    return pfs;
}

void printvector(vi v) {
    for (int i = 0; i < v.size(); i++) {
        cout << v[i];
        if (i != v.size()-1) cout << " x ";
    }
}

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        vi facs = primefacs(n);
        cout << n << " = ";
        printvector(facs);
        cout << endl;
    }
}
```

1.47 UVa 591: Box of Bricks

```
#include <iostream>

using namespace std;

int main() {
    int t;
    int setn = 0;
    while (cin >> t) {
        if (t == 0) break;
        setn++;
        int arr[t];
        int sum = 0, moves = 0;
        for (int i = 0; i < t; i++) {
            cin >> arr[i];
            sum += arr[i];
        }
        int l = sum/t;
        for (int i = 0; i < t; i++) {
            if (arr[i] > l) moves += arr[i]-l;
        }
        cout << "Set #" << setn << endl;
        cout << "The minimum number of moves is " << moves << "." << endl << endl;
    }
}
```

1.48 UVa 594: One Little, Two Little, Three Little Endians

```
#include <iostream>

using namespace std;

typedef union {
        char c[4];
        int n;
} bits;
```

```
int main() {
	bits a, b;
	while (cin >> a.n) {
		for (int i = 0; i < 4; i++) {
			b.c[i] = a.c[3-i];
		}
		cout << a.n << " converts to " << b.n << endl;
	}
}
```

1.49 UVa 621: Secret Research

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        string seq;
        char res;
        cin >> seq;
        if (seq.compare("1") == 0 || seq.compare("4") == 0 || seq.compare("78") == 0 ) res = '+';
        else if (seq[seq.length()-2] == '3' && seq[seq.length()-1] == '5') res = '-';
        else if (seq[0] == '9' && seq[seq.length()-1] == '4') res = '*';
        else if (seq[0] == '1' && seq[1] == '9' && seq[2] == '0') res = '?';
        cout << res << endl;
    }
}
```

1.50 UVa 637: Booklet Printing

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        int make4 = (int)ceil(n/4.0)*4;
        cout << "Printing order for " << n << " pages:" << endl;
        if (n == 1) cout << "Sheet 1, front: Blank, 1" << endl;
        else {
            for (int i = 1; i <= make4/2; i++) {
                cout << "Sheet " << ceil(i/2.0) << ", ";
                if (i%2) {
                    cout << "front: ";
                    if (make4-i+1 > n) cout << "Blank";
                    else cout << make4-i+1;
                    cout << ", " << i;
                } else {
                    cout << "back : " << i;
                    cout << ", ";
                    if (make4-i+1 > n) cout << "Blank";
```

```
                else cout << make4-i+1;
            }
            cout << endl;
        }
    }
}
```

1.51 UVa 661: Blowing Fuses

```
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    int n, m, c, tc = 0;
    while (cin >> n >> m >> c && n+m+c) {
        int A[n], O[m]; bool tog[n], blown = false;
        for (int i = 0; i < n; i++) {
            cin >> A[i];
            tog[i] = false;
        }
        for (int i = 0; i < m; i++) cin >> O[i];
        int C = 0, temp = 0;
        for (int i = 0; i < m; i++) {
            tog[O[i]-1] = !tog[O[i]-1];
            if (tog[O[i]-1]) temp += A[O[i]-1];
            else temp -= A[O[i]-1];
            if (temp > c) {
                blown = true;
                break;
            } else C = max(C, temp);
        }
        cout << "Sequence␣" << ++tc << endl;
        if (blown) {
            cout << "Fuse␣was␣blown." << endl;
        } else {
            cout << "Fuse␣was␣not␣blown." << endl;
            cout << "Maximal␣power␣consumption␣was␣" << C << "␣amperes." << endl;
        }
        if (tc > 0) cout << endl;
    }
}
```

1.52 UVa 673: Parentheses Balance

```
#include <iostream>
#include <cstdio>
#include <string>
#include <stack>

using namespace std;

bool paired(char op, char cl) {
    if (op == '[' && cl == ']') return true;
    else if (op == '(' && cl == ')') return true;
    return false;
}
```

```
bool check(string delims) {
    stack<char> parens;
    for (string::size_type i = 0; i < delims.length(); i++) {
        char delim = delims[i];
        if (delim == '[' || delim == '(') parens.push(delim);
        else if (delim == ']' || delim == ')') {
            if (parens.empty() || !(paired(parens.top(), delim))) return false;
            else parens.pop();
        }
    }
    if (parens.empty()) return true;
    else return false;
}

int main() {
    int t;
    scanf("%d\n",&t);
    for (int i = 1; i <= t; i++) {
        string delims;
        getline(cin, delims);
        if (check(delims)) cout << "Yes" << endl;
        else cout << "No" << endl;
    }
}
```

1.53 UVa 674: Coin Change

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

using namespace std;

int count(int S[], int m, int n) {
        int memo[n+1];
        memset(memo, 0, sizeof memo);
        memo[0] = 1;
        for (int i = 0; i < m; i++)
                for (int j = S[i]; j <= n; j++)
                        memo[j] += memo[j-S[i]];

        return memo[n];
}

int main() {
        int k, coins[5] = {1,5,10,25,50};
        while (cin >> k) {
                cout << count(coins, 5, k) << endl;
        }
}
```

1.54 UVa 681: Convex Hull Finding

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

using namespace std;
const double EPS = 1e-7;

struct Point { double x,y; };
```

```
bool cmp(Point a, Point b) {
    if (fabs(a.x - b.x) < EPS) return a.y < b.y;
    else return a.x < b.x;
}

double cross(Point a, Point b) {
    return a.x * b.y - a.y * b.x;
} // CCW: (+) CW: (-)
double cross(Point a, Point b, Point c) {
    return cross(a, b) + cross(b, c) + cross(c, a);
}

vector<Point> CH(vector<Point> &p) {
    int n = p.size(), k = 0;
    if (n <= 1) return p;
    sort(p.begin(), p.end(), cmp);
    vector<Point> h(2*n);
    for (int i = 0; i < n; h[k++] = p[i++])
        while (k >= 2 && cross(h[k-2], h[k-1], p[i]) > -EPS)
            --k;
    for (int i = n-2, t = k; i >= 0; h[k++] = p[i--])
        while (k > t && cross(h[k-2], h[k-1], p[i]) > -EPS)
            --k;
    k -= 1 + ((h[0].x == h[1].x && h[0].y == h[1].y) ? 1 : 0);
    h.resize(k);
    return h;
}

int main() {
    int k; cin >> k;
    cout << k << endl;
    while (k--) {
        int n; cin >> n;
        vector<Point> p(n);
        for (int i = 0; i < n; i++) {
            cin >> p[i].x >> p[i].y;
        }
        if (k) {
            int minus1; cin >> minus1;
        }
        vector<Point> h = CH(p);
        reverse(h.begin(), h.end());
        cout << h.size()+1 << endl;
        int mn_x = h[0].x, mn_y = h[0].y, idx = 0;
        for (int i = 0; i < h.size(); i++) {
            if (h[i].y < mn_y) {
                idx = i;
                mn_y = h[i].y;
            } else if (h[i].y == mn_y && h[i].x < mn_x) {
                idx = i;
                mn_x = h[i].x;
            }
        }
        for (int i = 0; i <= h.size(); i++) {
            cout << h[(i+idx)%h.size()].x << "␣" << h[(i+idx)%h.size()].y << endl;
        }
        if (k) cout << -1 << endl;
    }
```

```
}
```

1.55 UVa 729: The Hamming Distance Problem

```
#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int n, h; cin >> n >> h;
        string s = "";
        for (int i = 0; i < n-h; i++) s += "0";
        for (int i = n-h; i < n; i++) s += "1";
        do {
            cout << s << endl;
        } while (next_permutation(s.begin(), s.end()));
        if (t > 0) cout << endl;
    }
}
```

1.56 UVa 756: Biorhythms

```
#include <iostream>
#include <cmath>

using namespace std;

int x, y;

int iegcd(int a, int b) {
    if (b == 0) {
        x = 1; y = 0;
        return a;
    }
    int g = iegcd(b, a%b);
    int z = x - y*(a/b);
    x = y; y = z;
    return g;
}

int invmod(int a, int m) {
    int g = iegcd(a, m);
    return (x%m + m) % m;
}

int ichinrem(int a[], int m[]) {
    int n = 3, mm = 1, sum = 0;
    for (int i = 0; i < n; i++) {
        mm *= m[i];
    }
    for (int i = 0; i < n; i++) {
        int mmi = mm/m[i];
        int x = invmod(mmi, m[i]);
        int nxt = (x%mm * a[i]%mm * mmi%mm);
        sum = (sum+nxt) % mm;
```

```
    }
    return (sum+mm) % mm;
}

int main() {
    int p, e, i, d, n = 0;
    while (cin >> p >> e >> i >> d) {
        if (p == -1 && e == -1 && i == -1) break;
        n++;
        int peaks[3] = {p,e,i};
        int cycle[3] = {23,28,33};
        int days = ichinrem(peaks, cycle);
        cout << "Case␣" << n << ":␣the␣next␣triple␣peak␣occurs␣in␣";
        int nrm = ((days-d)%21252 + 21252)%21252;
        if (nrm == 0) cout << 21252;
        else cout << nrm;
        cout << "␣days." << endl;
    }
}
```

1.57 UVa 759: The Return of the Roman Empire

```
#include <iostream>
#include <string>
#include <map>

using namespace std;

string i2r(int n) {
    string r[] = {"M","CM","D","CD","C","XC","L","XL","X","IX","V","IV","I"};
    int h[] = {1000,900,500,400,100,90,50,40,10,9,5,4,1};
    string rom = "";
    int i = 0;
    while (n) {
        if (n < h[i]) i++;
        else {
            n -= h[i];
            rom += r[i];
        }
    }
    return rom;
}

int r2i(string s) {
    if (s.compare("") == 0) return 0;
    map<char,int> rd;
    rd['M'] = 1000; rd['D'] = 500; rd['C'] = 100; rd['L'] = 50;
    rd['X'] = 10; rd['V'] = 5; rd['I'] = 1;
    int n = 0;
    for (int i = 0; i < s.length()-1; i++) {
        if (rd[s[i]] < rd[s[i+1]]) n -= rd[s[i]];
        else n += rd[s[i]];
    }
    n += rd[s[s.length()-1]];
    return n;
}

int main() {
    string rom;
```

```
    while (getline(cin, rom)) {
        if (rom.compare(i2r(r2i(rom))) != 0 || !(0 <= r2i(rom) && r2i(rom) <= 3999)) cout << "This is not a valid number" << endl;
        else cout << r2i(rom) << endl;
    }
}
```

1.58 UVa 821: Page Hopping

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <limits>
#include <map>

using namespace std;
int d[101][101], adj[101][101];
const int INF = numeric_limits<int>::max();
const int inf = INF/2;

void fw(int V) {
	for(int i = 0; i < V; i++)
		for(int j = 0; j < V; j++)
			d[i][j] = adj[i][j];
	for(int k = 0; k < V; k++)
		for(int i = 0; i < V; i++)
			for(int j = 0; j < V; j++)
				d[i][j] = min(d[i][j], d[i][k] + d[k][j]);

}

void reset() {
	for (int i = 0; i < 101; i++) {
		for (int j = 0; j < 101; j++) {
			d[i][j] =  INF;
			adj[i][j] = inf;
		}
	}
}


int main() {
	reset();
	int a, b, aa, bb, c = 0, V = 0, v = 0, dist = 0;
	while (cin >> aa >> bb) {
		map<int,int> vv;
		if (!aa & !bb) break;
		else {
			adj[aa-1][bb-1] = 1;
			V = max(V, max(aa,bb));

			while (cin >> a >> b) {
				if (!a && !b) {
					fw(V);
					v = vv.size();
					for (int i = 0; i < V; i++)
						for (int j = 0; j < V; j++)
							if (i != j && d[i][j] != inf) dist += d[i][j];
```

```
                                printf("Case␣%d:␣average␣length␣between␣pages␣=␣%.3f␣↩
                                    clicks\n", ++c, dist/(v*(v-1)*1.0));
                                V = 0; dist = 0;
                                reset();
                                break;
                            }
                            adj[a-1][b-1] = 1;
                            if (vv.find(a) == vv.end()) vv[a]=1;
                            if (vv.find(b) == vv.end()) vv[b]=1;
                            V = max(V, max(a,b));
                        }
                    }
                }
}
```

1.59 UVa 834: Continued Fractions

```
#include <iostream>
#include <vector>

using namespace std;
typedef long long ll;

ll x, y;
vector<ll> coef;

ll iegcd(ll a, ll b) {
    if (b == 0) {
        x = 1; y = 0;
        return a;
    }
    coef.push_back(a/b);
    ll g = iegcd(b, a%b);
    ll z = x - y*(a/b);
    x = y; y = z;
    return g;
}

int main() {
    ll a, b;
    while (cin >> a >> b) {
        if (a%b == 0) cout << "[" << a/b << "]" << endl;
        else {
            iegcd(a, b);
            cout << "[" << coef[0] << ";";
            for (int i = 1; i < coef.size()-1; i++) {
                cout << coef[i] << ",";
            }
            cout << coef[coef.size()-1] << "]" << endl;
            coef.clear();
        }
    }
}
```

1.60 UVa 1124: Celebrity Jeopardy

```
#include <iostream>
#include <string>
```

```
using namespace std;

int main() {
    string line;
    while (getline(cin, line)) {
        cout << line << endl;
    }
}
```

1.61 UVa 1230: MODEX

```
#include <iostream>
#include <cmath>

using namespace std;
typedef long long ll;

ll modpow(ll base, ll pw, int mod) {
    ll res = 1;
    base = base%mod;
    while (pw > 0) {
        if (pw%2 == 1) {
            res = (res*base)%mod;
        }
        pw >>= 1;
        base = (base*base)%mod;
    }
    return res;
}

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        while (n--) {
            ll x, y; int n;
            cin >> x >> y >> n;
            cout << modpow(x, y, n) << endl;
        }
    }
}
```

1.62 UVa 1237: Expert Enough?

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

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int n; cin >> n;
        string maker[n]; int mn[n], mx[n];
        for (int i = 0; i < n; i++) {
            cin >> maker[i] >> mn[i] >> mx[i];
        }
        int q; cin >> q;
        for (int i = 0; i < q; i++) {
```

```
            int k, idx = -1, cnt = 0; cin >> k;
            for (int j = 0; j < n; j++) {
                if (mn[j] <= k && k <= mx[j]) {
                    idx = j; cnt++;
                }
            }
            if (cnt == 1) cout << maker[idx] << endl;
            else cout << "UNDETERMINED" << endl;
        }
        if (t > 0) cout << endl;
    }
}
```

1.63 UVa 10004: Bicoloring

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

using namespace std;
typedef map< int, vector<int> > graph;

template <class T>
T poll(queue<T> &q) {
    T a = q.front(); q.pop();
    return a;
}

bool bfs(graph &G, int src, int V) {
    int color[V];
    for (int i = 0; i < V; i++) color[i] = -1;
    color[src] = 1;

    queue<int> q;
    q.push(src);
    while (!q.empty()) {
        int u = poll(q);
        for (vector<int>::iterator v = G[u].begin(); v != G[u].end(); v++) {
            if (find(G[u].begin(), G[u].end(), *v) != G[u].end() && color[*v] == -1) {
                color[*v] = 1-color[u];
                q.push(*v);
            } else if (find(G[u].begin(), G[u].end(), *v) != G[u].end() && color[*v] == color[u↩
                ]) return false;
        }
    }
    return true;
}

int main() {
    int V;
    while (cin >> V && V) {
        int E; cin >> E;
        graph G;
        while (E--) {
            int a, b; cin >> a >> b;
            G[a].push_back(b);
        }
        if (!bfs(G, 0, V)) cout << "NOT␣";
        cout << "BICOLORABLE." << endl;
    }
}
```

1.64 UVa 10006: Carmichael Numbers

```
#include <iostream>

using namespace std;

bool car(int n) {
    int nums[15] = {561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, 41041, ↩
        46657, 52633, 62745, 63973};
    bool isit = false;
    for (int i = 0; i < 15; i++) {
        if (n == nums[i]) {
            isit = true;
            break;
        }
    }
    return isit;
}

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        if (car(n)) cout << "The␣number␣";
        cout << n;
        if (car(n)) cout << "␣is␣a␣Carmichael␣number.";
        else cout << "␣is␣normal.";
        cout << endl;
    }
}
```

1.65 UVa 10008: What's Cryptanalysis?

```
#include <iostream>
#include <cstdio>
#include <string>
#include <map>
#include <algorithm>
#include <utility>
#include <vector>
#include <cctype>

using namespace std;

bool cmp(pair<char,int> a, pair<char,int> b) {
    if (a.second == b.second) return a.first < b.first;
    else return a.second > b.second;
}

int main() {
    int t; scanf("%d\n", &t);
    map<char,int> tally;
    vector<pair<char,int> > cnt;
    while (t--) {
        string words;
        getline(cin, words);
        for (int i = 0; i < words.size(); i++) {
            char l = words[i];
```

```
            if (isalpha(l)) tally[toupper(l)]++;
        }
    }
    for (map<char,int>::iterator it = tally.begin(); it != tally.end(); it++) {
        cnt.push_back(make_pair(it->first,it->second));
    }
    sort(cnt.begin(), cnt.end(), cmp);
    for (int i = 0; i < cnt.size(); i++) {
        cout << cnt[i].first << "␣" << cnt[i].second << endl;
    }
}
```

1.66 UVa 10019: Funny Encryption Method

```
#include <iostream>
#include <sstream>
#include <bitset>

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int n; cin >> n;
        int h; stringstream hh;
        hh << n;
        hh >> hex >> h;
        bitset<16> bd(n);
        bitset<16> bh(h);
        cout << bd.count() << "␣" << bh.count() << endl;
    }
}
```

1.67 UVa 10038: Jolly Jumpers

```
#include <iostream>
#include <algorithm>
#include <cmath>

using namespace std;

int main() {
    int n;
    while (cin >> n) {
        int s[n], d[n-1];
        for (int i = 0; i < n; i++) cin >> s[i];
        for (int i = 0; i < n-1; i++) {
            d[i] = abs(s[i+1]-s[i]);
        }
        bool jolly = true;
        sort(d, d+n-1);
        for (int i = 0; i < n-1; i++) {
            if (i+1 != d[i]) {
                jolly = false;
                break;
            }
        }
        if (jolly) cout << "Jolly" << endl;
        else cout << "Not␣jolly" << endl;
```

```
    }
}
```

1.68 UVa 10050: Hartals

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

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int d, n; cin >> d >> n;
        int h[n], cnt = 0;
        for (int i = 0; i < n; i++) cin >> h[i];
        for (int i = 1; i <= d; i++) {
            bool strike = false;
            if (!(i%7 == 0 || i%7 == 6)) {
                for (int j = 0; j < n; j++) strike |= i%h[j] == 0;
            }
            if (strike) cnt++;
        }
        cout << cnt << endl;
    }
}
```

1.69 UVa 10055: Hashmat the Brave Warrior

```
#include <iostream>
using namespace std;

int main() {
        long long a,b;
        while (cin >> a >> b) {
                long long res;
                if (a>b) res = a-b;
                else if (a<b) res = b-a;
                else res = 0;
                cout << res << endl;
        }
        return 0;
}
```

1.70 UVa 10062: Tell Me the Frequencies!

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

using namespace std;

bool cmp(pair<int,int> a, pair<int,int> b) {
    if (a.second == b.second) return a.first > b.first;
    return a.second < b.second;
}

int main() {
    string line; int t = 0;
    while (getline(cin, line)) {
        if (t++ > 0) cout << endl;
        map<int,int> fr;
```

```
        vector< pair<int,int> > rf;
        for (int i = 0; i < line.length(); i++) {
            char c = line[i];
            fr[(int)c]++;
        }
        for (map<int,int>::iterator it = fr.begin(); it != fr.end(); it++) {
            rf.push_back(make_pair(it->first, it->second));
        }
        sort(rf.begin(), rf.end(), cmp);
        for (int i = 0; i < rf.size(); i++) cout << rf[i].first << "␣" << rf[i].second << endl;
    }
}
```

1.71 UVa 10070: Leap Year or not Leap Year and...

```
#include <iostream>

using namespace std;

int bigmod(string num, int m) {
    int res = 0;
    for (int i = 0; i < num.length(); i++) {
        res = (10*res + (num[i]-'0'))%m;
    }
    return res;
}

int main() {
    string y;
    bool done = false;
    while (cin >> y) {
        if (done) cout << endl;
        done = true;
        bool leap = (bigmod(y,400) == 0) || (bigmod(y,4) == 0 && bigmod(y,100) != 0);
        bool hul = bigmod(y,15) == 0;
        bool bul = leap && (bigmod(y,55) == 0);
        if (leap) cout << "This␣is␣leap␣year." << endl;
        if (hul) cout << "This␣is␣huluculu␣festival␣year." << endl;
        if (bul) cout << "This␣is␣bulukulu␣festival␣year." << endl;
        if (!(leap || hul || bul)) cout << "This␣is␣an␣ordinary␣year." << endl;
    }
}
```

1.72 UVa 10078: The Art Gallery

```
#include <iostream>
#include <cmath>

using namespace std;

struct Point {
    double x, y;
};

double cross(Point a, Point b) {
    return a.x * b.y - a.y * b.x;
} // CCW: (+) CW: (-)

double cross(Point a, Point b, Point c) {
```

```
    return cross(a, b) + cross(b, c) + cross(c, a);
}

int sgn(int n) {
    if (n == 0) return 0;
    else return ((n>0)?1:-1);
}

int main() {
    int n;
    while (cin >> n && n) {
        Point p[n]; bool convex = true;
        for (int i = 0; i < n; i++) cin >> p[i].x >> p[i].y;
        int sign = sgn(cross(p[0], p[1], p[2]));
        for (int i = 0; i < n; i++) {
            convex &= sgn(cross(p[i%n], p[(i+1)%n], p[(i+2)%n])) == sign;
        }
        cout << (convex ? "No" : "Yes") << endl;
    }
}
```

1.73 UVa 10079: Pizza Cutting

```
#include <iostream>

using namespace std;

int main() {
        while (true) {
                long long n; cin >> n;
                if (n < 0) break;
                cout << (n*n + n + 2)/2 << endl;
        }
}
```

1.74 UVa 10082: WERTYU

```
#include <iostream>
#include <cstdio>
#include <map>

using namespace std;

int main() {
        map<char,char> sublist;
        char input;
        sublist['='] ='-';
        sublist['1'] = '`';
        sublist['2'] = '1';
        sublist['3'] = '2';
        sublist['4'] = '3';
        sublist['5'] = '4';
        sublist['6'] = '5';
        sublist['7'] = '6';
        sublist['8'] = '7';
        sublist['9'] = '8';
        sublist['0'] = '9';
        sublist['-'] = '0';
        sublist['\\'] = ']';
```

```
        sublist['W'] = 'Q';
        sublist['E'] = 'W';
        sublist['R'] = 'E';
        sublist['T'] = 'R';
        sublist['Y'] = 'T';
        sublist['U'] = 'Y';
        sublist['I'] = 'U';
        sublist['O'] = 'I';
        sublist['P'] = 'O';
        sublist['['] = 'P';
        sublist[']'] = '[';
        sublist['S'] = 'A';
        sublist['D'] = 'S';
        sublist['F'] = 'D';
        sublist['G'] = 'F';
        sublist['H'] = 'G';
        sublist['J'] = 'H';
        sublist['K'] = 'J';
        sublist['L'] = 'K';
        sublist[';'] = 'L';
        sublist['\''] = ';';
        sublist['X'] = 'Z';
        sublist['C'] = 'X';
        sublist['V'] = 'C';
        sublist['B'] = 'V';
        sublist['N'] = 'B';
        sublist['M'] = 'N';
        sublist[','] = 'M';
        sublist['.'] = ',';
        sublist['/'] = '.';
        while (scanf("%c",&input) == 1) {
                if (input == ' ') cout << " ";
                else if (input == '\n') cout << "\n";
                else cout << sublist[input];
        }
}
```

1.75 UVa 10104: Euclid Problem

```
#include <iostream>

using namespace std;
typedef long long ll;

ll x, y;
ll gcdext(ll a, ll b) {
        if (b == 0) {
                x = 1;
                y = 0;
                return a;
        }
        ll g = gcdext(b, a%b);
        ll z = x - (a/b)*y;
        x = y; y = z;
        return g;
}

int main() {
        ll a, b;
```

```
        while (cin >> a >> b) {
                ll g = gcdext(a, b);
                cout << x << "␣" << y << "␣" << g << endl;
        }
}
```

1.76 UVa 10107: What is the Median?

```
#include <iostream>
#include <algorithm>
#include <vector>

using namespace std;

int main() {
    int n;
    vector<int> l;
    while (cin >> n) {
        l.push_back(n);
        nth_element(l.begin(), l.begin()+l.size()/2, l.end());
        cout << l[l.size()/2] << endl;
    }
}
```

1.77 UVa 10110: Light, More Light

```
#include <iostream>
#include <cmath>

using namespace std;
typedef long long ll;

int main() {
    ll n;
    while (cin >> n) {
        if (n == 0) break;
        cout << (((ll)sqrt(n)*(ll)sqrt(n) == n)?"yes":"no") << endl;
    }
}
```

1.78 UVa 10130: SuperSale

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

using namespace std;
int memo[1010][31], v[1010], w[1010];

int main() {
    int t; cin >> t;
    while (t--) {
        int n; cin >> n;
        for (int i = 0; i < n; i++) cin >> v[i] >> w[i];
        for (int c = 0; c < 31; c++) memo[0][c] = 0;
        for (int i = 1; i <= n; i++) {
            for (int c = 0; c < 31; c++) {
                memo[i][c] = memo[i-1][c];
                if (c >= w[i-1]) {
                    if(v[i-1] + memo[i-1][c - w[i-1]] > memo[i][c])
                        memo[i][c] = v[i-1] + memo[i-1][c - w[i-1]];
```

```
                }
            }
        }
        int f, sum = 0; cin >> f;
        while (f--) {
            int mw; cin >> mw;
            sum += memo[n][mw];
        }
        cout << sum << endl;
    }
}
```

1.79 UVa 10179: Irreducible Basic Fractions

```
#include <iostream>
#include <cmath>

using namespace std;
typedef long long ll;

ll phi(ll n) {
        ll ret = 1, i, pw;
        for (i = 2; n != 1; i++) {
                pw = 1;
                if (i > sqrt(n)) break;
                while (!(n % i)) {
                        n /= i;
                        pw *= i;
                }
                ret *= (pw - (pw/i));
        }
        if (n != 1) ret *= (n-1);
        return ret;
}

int main() {
        ll n; cin >> n;
        while (n != 0) {
                cout << phi(n) << endl;
                cin >> n;
        }
}
```

1.80 UVa 10189: Minesweeper

```
#include <iostream>

using namespace std;

int main() {
    int n, m, c = 0;
    while (cin >> n >> m) {
        if (n == 0 && m == 0) break;
        else if (c != 0) cout << endl;

        char g[101][101];
        int p[101][101];
        for (int i = 0; i < 101; i++) {
            for (int j = 0; j < 101; j++) {
```

```
            g[i][j] = '.';
            p[i][j] = 0;
        }
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> g[i][j];
        }
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (g[i][j] == '*') {
                if (i == 0 && j == 0) {
                    p[0][1]++; p[1][0]++; p[1][1]++;
                } else if (i == 0 && 0 < j && j < m-1) {
                    p[0][j-1]++; p[1][j-1]++; p[0][j]++; p[1][j]++; p[0][j+1]++; p[1][j↩
                        +1]++;
                } else if (j == 0 && 0 < i && i < n-1) {
                    p[i-1][0]++; p[i-1][1]++; p[i][0]++; p[i][1]++; p[i+1][0]++; p[i↩
                        +1][1]++;
                } else if (i == n-1 && j == m-1) {
                    p[n-2][m-2]++; p[n-2][m-1]++; p[n-1][m-2]++;
                } else {
                    p[i-1][j-1]++; p[i-1][j]++; p[i-1][j+1]++;
                    p[i][j-1]++; p[i][j+1]++;
                    p[i+1][j-1]++; p[i+1][j]++; p[i+1][j+1]++;
                }
            }
        }

        cout << "Field␣#" << ++c << ":" << endl;

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                if (g[i][j] == '*') cout << "*";
                else cout << p[i][j];
            }
            cout << endl;
        }

    }
}
```

1.81 UVa 10195: The Knights of the Round Table

```
#include <iostream>
#include <cmath>
#include <cstdio>

using namespace std;

int main() {
    double a, b, c;
    while (cin >> a >> b >> c) {
        double s = (a+b+c)/2, area = sqrt(s*(s-a)*(s-b)*(s-c));
        double r = (s>0 ? area/s : 0);
```

```
                printf("The␣radius␣of␣the␣round␣table␣is:␣%.3f\n", r);
        }
}
```

1.82 UVa 10209: Is This Integration?

```
#include <iostream>
#include <cmath>
#include <cstdio>

using namespace std;
const double pi = 2*acos(0);

int main() {
        double a;
        while (cin >> a) {
                double grid = 8*(pow(a,2)/2 - pow(a/2,2)*sin(pi/3) - (pi*pow(a,2))/12);
                double dot = 4*(pow(a,2) - grid/2 - (pi*pow(a,2))/4);
                double strip = pow(a,2) - grid - dot;
                printf("%.3f␣%.3f␣%.3f\n", strip, dot, grid);
        }
}
```

1.83 UVa 10226: Hardwood Species

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

using namespace std;

int main() {
        int t; scanf("%d\n\n", &t);
        while (t--) {
                map<string,int> trees; int cnt = 0;
                string tree;
                while (getline(cin, tree)) {
                        if (tree.length() == 0) break;
                        else trees[tree]++; cnt++;
                }
                for (map<string,int>::iterator it = trees.begin(); it != trees.end(); it++) {
                        cout << it->first;
                        printf("␣%.4f\n", 100.0*(it->second)/cnt);
                }
                if (t > 0) cout << endl;
        }

}
```

1.84 UVa 10235: Simply Emirp

```
#include <iostream>
#include <sstream>
#include <cmath>
#include <algorithm>

using namespace std;

int parse_int(string s) {
    stringstream ss(s);
    int i; ss >> i;
```

```
    return i;
}

bool isprime(int n) {
    if (n == 2) return true;
    else if (n%2 == 0) return false;
    else {
        for (int i = 3; i <= (int)sqrt(n); i += 2) {
            if (n%i == 0) return false;
        }
        return true;
    }
}

int main() {
    string s;
    while (cin >> s) {
        int n = parse_int(s);
        reverse(s.begin(), s.end());
        int nr = parse_int(s);
        cout << n << " is ";
        if (isprime(n)) {
            if (isprime(nr) && n != nr) cout << "emirp.";
            else cout << "prime.";
        } else cout << "not prime.";
        cout << endl;
    }
}
```

1.85 UVa 10264: The Most Potent Corner

```
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
        int n;
        while (cin >> n) {
                int mx = 1 << n;
                int edges[mx];
                for (int i = 0; i < mx; i++) cin >> edges[i];
                int sum = 0, pot[mx];
                for (int i = 0; i < mx; i++) {
                        int esum = 0;
                        for (int j = 0; j < n; j++) {
                                esum += edges[i ^ (1 << j)];
                        }
                        pot[i] = esum;
                }
                for (int i = 0; i < mx; i++) {
                        for (int j = 0; j < n; j++) {
                                sum = max(sum, pot[i] + pot[i ^ (1 << j)]);
                        }
                }
                cout << sum << endl;
        }
}
```

1.86 UVa 10299: Relatives

```
#include <iostream>
#include <cmath>

using namespace std;
typedef long long ll;

ll phi(ll n) {
	ll ret = 1, i, pw;
	for (i = 2; n != 1; i++) {
		pw = 1;
		if (i > sqrt(n)) break;
		while (!(n % i)) {
			n /= i;
			pw *= i;
		}
		ret *= (pw - (pw/i));
	}
	if (n != 1) ret *= (n-1);
	return ret;
}

int main() {
	ll n;
	while (cin >> n) {
	if (n == 0) break;
		cout << ((n == 1) ? 0 : phi(n)) << endl;
	}
}
```

1.87 UVa 10300: Ecological Premium

```
#include <iostream>

using namespace std;

int main() {
	int t; cin >> t;
	while (t--) {
		int n; cin >> n;
		int s, a, f, pr = 0;
		for (int i = 0; i < n; i++) {
			cin >> s >> a >> f;
			pr += s*f;
		}
		cout << pr << endl;
	}
}
```

1.88 UVa 10302: Summation of Polynomials

```
#include <iostream>
#include <cmath>

typedef long long ll;
using namespace std;

int main() {
```

```
        ll n;
        while (cin >> n) {
                cout << ((n*(n+1))/2)*((n*(n+1))/2) << endl;
        }
}
```

1.89 UVa 10341: Solve It

```
#include <iostream>
#include <cmath>
#include <cstdio>

using namespace std;

const double eps = 1e-8;
int p, q, r, s, t, u;

double f(double x) {
        return p*exp(-x) + q*sin(x) + r*cos(x) + s*tan(x) + t*x*x + u;
}

double bisection() {
        double a = 0, b = 1;
        while (a + eps < b) {
                double c = (a+b)/2;
                if (f(c) * f(a) <= 0) b = c;
                else a = c;
        }
        return (a+b)/2;
}

int main() {
        while (scanf("%d␣%d␣%d␣%d␣%d",&p,&q,&r,&s,&t,&u) != EOF) {
                if (f(0) * f(1) > 0) {
                        cout << "No␣solution" << endl;
                } else {
                        printf("%.4f\n", bisection());
                }
        }

}
```

1.90 UVa 10346: Peter's Smokes

```
#include <iostream>

using namespace std;
typedef long long ll;

int main() {
    ll n, k;
    while (cin >> n >> k) {
        cout << n + (n-1)/(k-1) << endl;
    }
}
```

1.91 UVa 10347: Medians

```
#include <iostream>
```

```
#include <cmath>
#include <cstdio>

using namespace std;

int main() {
	double a, b, c;
	while (cin >> a >> b >> c) {
		double s = (a+b+c)/2, area = (4.0/3)*sqrt(s*(s-a)*(s-b)*(s-c));
		printf("%.3f\n", (area > 0) ? area : -1);
	}
}
```

1.92 UVa 10370: Above Average

```
#include <iostream>
#include <cstdio>

using namespace std;

int main() {
	int tc;
	cin >> tc;
	for (int i = 1; i <= tc; i++) {
		int nc;
		cin >> nc;
		int sum = 0;
		int aa = 0;
		int grs[nc];
		for (int c = 0; c < nc; c++) {
			int gr;
			cin >> gr;
			sum += gr;
			grs[c] = gr;
		}
		for (int j = 0; j < nc; j++) {
			if (grs[j] > sum/nc) aa++;
		}
		double per = ((double)aa/nc)*100;
		printf("%.3f", per);
		cout << "%" << endl;
	}
	return 0;
}
```

1.93 UVa 10378: Complex Numbers

```
#include <iostream>
#include <cstdio>
#include <complex>
#include <vector>
#include <algorithm>

using namespace std;
const double pi = 2*acos(0);

bool cmp(complex<double> a, complex<double> b) {
    if (fabs(a.real()-b.real()) > 1e-6) return a.real() > b.real();
    else if (fabs(a.imag()-b.imag()) < 1e-6) return false;
```

```
    else return a.imag() > b.imag();
}

int main() {
    int re, im, n, c = 0; char sign, i;
    while (cin >> re >> sign >> im >> i >> n) {
        complex<double> z(re, (sign == '-')?-im:im);
        vector<complex<double> > roots;
        double r = abs(z);
        double theta = arg(z);
        cout << "Case␣" << ++c << ":" << endl;
        for (int k = 0; k < n; k++) {
            double nr = pow(r, 1.0/n);
            complex<double> root(nr*cos((theta+2*pi*k)/n), nr*sin((theta+2*pi*k)/n));
            roots.push_back(root);
        }
        sort(roots.begin(), roots.end(), cmp);
        for (int k = 0; k < n; k++) {
            complex<double> rt = roots[k];
            printf("%.3f%+.3fi\n", (fabs(rt.real()) < 0.0005)?0:rt.real(), (fabs(rt.imag()) < ↩
                0.0005)?0:rt.imag());
        }
        cout << endl;
    }
}
```

1.94 UVa 10405: Longest Common Subsequence

```
#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

int lcs(string x, string y) {
    int m = x.length(), n = y.length();
    int l[m+1][n+1];
    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0 || j == 0) l[i][j] = 0;
            else if (x[i-1] == y[j-1])
                l[i][j] = l[i-1][j-1] +1;
            else
                l[i][j] = max(l[i-1][j], l[i][j-1]);
        }
    }
    return l[m][n];
}

int main() {
    string s1, s2;
    while (getline(cin, s1) && getline(cin, s2)) {
        cout << lcs(s1,s2) << endl;
    }
}
```

1.95 UVa 10408: Farey Sequences

```
#include <iostream>
```

```
#include <cmath>
#include <vector>
#include <utility>

using namespace std;
typedef pair<int,int> frac;

void farey(int n, vector<frac> &ff) {
    double x1 = 0, y1 = 1, x2 = 1, y2 = n;
    ff.push_back(make_pair(x1, y1));
    ff.push_back(make_pair(x2, y2));
    double x, y = 0;
    while (y != 1.0) {
        x = floor((y1+n)/y2) * x2 - x1;
        y = floor((y1+n)/y2) * y2 - y1;
        ff.push_back(make_pair(x, y));
        x1 = x2; x2 = x;
        y1 = y2; y2 = y;
    }
}

void printfrac(frac f) {
    cout << f.first << "/" << f.second << endl;
}

int main() {
    int n, k;
    while (cin >> n >> k) {
        vector<frac> fr;
        farey(n, fr);
        printfrac(fr[k]);
    }
}
```

1.96 UVa 10420: List of Conquests

```
#include <iostream>
#include <sstream>
#include <string>
#include <map>

using namespace std;

int main() {
    string tt; int t;
    getline(cin, tt);
    stringstream st(tt);
    st >> t;
    map<string,int> key;
    for (int i = 1; i <= t; i++) {
        string line, ct;
        getline(cin, line);
        stringstream ss(line);
        ss >> ct;
        key[ct]++;
    }
    for (map<string,int>::iterator it = key.begin(); it != key.end(); it++) {
        string ctr = it -> first;
        int cnt = it -> second;
```

```
        cout << ctr << "␣" << cnt << endl;
    }
}
```

1.97 UVa 10424: Love Calculator

```
#include <iostream>
#include <cstdio>
#include <cmath>
#include <cctype>
#include <string>

using namespace std;

int dr(int n) {
    return 1 + (n-1)%9;
}

int value(string s) {
    int sum = 0;
    for (int i = 0; i < s.length(); i++) {
        if (isalpha(s[i])) {
            if (isupper(s[i])) sum += (s[i]-'A'+1);
            else if (islower(s[i])) sum += (s[i]-'a'+1);
        }
    }
    return dr(sum);
}

int main() {
    string n1, n2;
    while (getline(cin, n1) && getline(cin, n2)) {
        int v1 = value(n1), v2 = value(n2);
        double pc = (min(v1,v2)*100.0)/max(v1,v2);
        if (v1 == 0 && v2 == 0) cout << endl;
        else printf("%.2f␣%%\n", pc);
    }
}
```

1.98 UVa 10432: Polygon Inside a Circle

```
#include <iostream>
#include <cstdio>
#include <cmath>

using namespace std;
const double pi = 2*acos(0);

int main() {
        double r; int n;
        while (cin >> r >> n) {
                double area = (n/2.0)*pow(r,2)*sin(2*pi/n);
                printf("%.3f\n", area);
        }
}
```

1.99 UVa 10451: Ancient Village Sports

```
#include <iostream>
#include <cmath>
#include <cstdio>

using namespace std;
const double pi = 2*acos(0);

int main() {
    int n, c = 0; double A;
    while (cin >> n >> A) {
        if (n < 3) break;
        double R = sqrt(2*A/(n*sin(2*pi/n)));
        double r = sqrt(A/(n*tan(pi/n)));
        double off = A-(pi*r*r);
        double spec = (pi*R*R)-A;
        printf("Case␣%d:␣%.5f␣%.5f\n", ++c, spec, off);
    }
}
```

1.100 UVa 10469: To Carry or Not to Carry

```
#include <iostream>

using namespace std;

int main() {
    unsigned long a, b, c;
    while (cin >> a >> b) {
        c = a^b;
        cout << c << endl;
    }
}
```

1.101 UVa 10550: Combination Lock

```
#include <iostream>

using namespace std;

int main() {
    int a, b, c, d;
    while (cin >> a >> b >> c >> d) {
        if (a==0 && b==0 && c==0 && d==0) break;
        int degs = 1080;
        if (a < b) degs += (40+a-b)*9;
        else degs += (a-b)*9;
        if (b > c) degs += (40+c-b)*9;
        else degs += (c-b)*9;
        if (c < d) degs += (40+c-d)*9;
        else degs += (c-d)*9;
        cout << degs << endl;
    }
}
```

1.102 UVa 10583: Ubiquitous Religions

```
#include <iostream>

using namespace std;
```

```
const int MAX = 50000;
int parent[MAX], rank[MAX];
int cnt;

void init(int n) {
    for (int i = 0; i < n; i++) {
        parent[i] = i;
        rank[i] = 0;
    }
}

int find(int obj) {
    if (parent[obj] != obj) parent[obj] = find(parent[obj]);
    return parent[obj];
}

void unite(int a, int b) {
    int a_root = find(a);
    int b_root = find(b);
    if (a_root != b_root) {
        if (rank[a_root] < rank[b_root]) parent[a_root] = b_root;
        else if (rank[a_root] > rank[b_root]) parent[b_root] = a_root;
        else {
            parent[b_root] = a_root;
            rank[a_root]++;
        }
    }
}

int main() {
    int n, m, c = 0;
    while (cin >> n >> m) {
        if (n == 0 && m == 0) break;
        cnt = 0;
        init(n);
        c++;
        for (int i = 0; i < m; i++) {
            int a, b;
            cin >> a >> b;
            unite(a, b);
        }
        for (int i = 0; i < n; i++) {
            if (parent[i] == i) cnt++;
        }
        cout << "Case␣" << c << ":␣" << cnt << endl;
    }
}
```

1.103 UVa 10684: The Jackpot

```
#include <iostream>

using namespace std;

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
```

```
        int gains[n];
        for (int i = 0; i < n; i++) cin >> gains[i];
        int sum = 0, mx = 0;
        for (int i = 0; i < n; i++) {
            sum += gains[i];
            if (sum < 0) sum = 0;
            if (mx < sum) mx = sum;
        }
        if (sum == 0) cout << "Losing␣streak." << endl;
        else cout << "The␣maximum␣winning␣streak␣is␣" << mx << "." << endl;
    }
}
```

1.104 UVa 10696: $f91$

```
#include <iostream>

using namespace std;

int f91(int n) {
        return (n >= 101)?n-10:91;
}

int main() {
        int m;
        cin >> m;
        while (m != 0) {
                cout << "f91(" << m << ")␣=␣" << f91(m) << endl;
                cin >> m;
        }
}
```

1.105 UVa 10699: Count the Factors

```
#include <iostream>
#include <cmath>
#include <vector>
#include <algorithm>

using namespace std;
typedef vector<int> vi;

vi primefacs(int n) {
    vi pfs;
    if (n < 0) {
        pfs.push_back(-1);
        n *= -1;
    }
    while (n%2 == 0) {
        pfs.push_back(2);
        n /= 2;
    }
    for (int i = 3; i <= sqrt(n); i += 2) {
        while (n%i == 0) {
            pfs.push_back(i);
            n /= i;
        }
    }
    if (n > 2) pfs.push_back(n);
```

```
    return pfs;
}

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        vi facs = primefacs(n);
        facs.resize(unique(facs.begin(), facs.end())-facs.begin());
        cout << n << " : " << facs.size() << endl;
    }
}
```

1.106 UVa 10773: Back to Intermediate Math

```
#include <iostream>
#include <cmath>
#include <cstdio>

using namespace std;

int main() {
    int t, c = 0;; cin >> t;
    while (t--) {
        double d, v, u; cin >> d >> v >> u;
        cout << "Case " << ++c << ": ";
        if (u == 0 || v == 0 || u <= v) {
            cout << "can't determine";
        } else {
            double fast = d/u;
            double shrt = d/sqrt(u*u-v*v);
            double dif = fabs(fast-shrt);
            printf("%.3f", dif);
        }
        cout << endl;
    }
}
```

1.107 UVa 10783: Odd Sum

```
#include <iostream>

using namespace std;

int main() {
        int tc;
        cin >> tc;
        for (int i = 1; i <= tc; i++) {
                int start, end;
                cin >> start >> end;
                int sum = 0;
                for (int n = start; n <= end; n++) {
                        if (n%2 != 0) sum += n;
                }
                cout << "Case " << i << ": " << sum << endl;
        }
        return 0;
}
```

1.108 UVa 10789: Prime Frequency

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

using namespace std;

bool isprime(int n) {
    if (n == 2) return true;
    else if (n%2 == 0 || n < 2) return false;
    else {
        for (int i = 3; i <= (int)sqrt(n); i += 2)
            if (n%i == 0) return false;
        return true;
    }
}

int main() {
    int t, c = 0; cin >> t;
    while (t--) {
        string s, fr = ""; cin >> s;
        map<char,int> v;
        for (int i = 0; i < s.length(); i++) v[s[i]]++;
        for (map<char,int>::iterator it = v.begin(); it != v.end(); it++)
            if (isprime(it->second)) fr += it->first;
        cout << "Case␣" << ++c << ":␣" << ((fr.length()>0)?fr:"empty") << endl;
    }
}
```

1.109 UVa 10812: Beat the Spread!

```
#include <iostream>
using namespace std;

int main() {
        int tc;
        cin >> tc;
        for (int i = 1; i <= tc; i++) {
                long long s, d;
                long long x, y;
                cin >> s >> d;
                x = (s+d)/2;
                y = s-x;
                if (2*x == s+d && y >= 0) cout << x << "␣" << y << endl;
                else cout << "impossible" << endl;
        }
        return 0;
}
```

1.110 UVa 10851: 2D Hieroglyphics Decoder

```
#include <iostream>
#include <cstdio>
#include <string>

using namespace std;

int main() {
    int t; scanf("%d\n", &t);
    while (t--) {
```

```
        string st; getline(cin, st);
        string txt(st.length()-2, '\0');
        for (int i = 1; i < 10; i++) {
            string ln; getline(cin, ln);
            if (i == 9) continue;
            for (int j = 1; j < ln.length()-1; j++) {
                if (ln[j] == '\\') txt[j-1] += 1<<(i-1);
            }
        }
        if (t > 0) getline(cin, st);
        cout << txt << endl;
    }
}
```

1.111 UVa 10878: Decode the Tape

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string s;
    while (getline(cin, s)) {
        if (s.compare("___________")) {
            char ch = 0;
            int t = 1 << 6;
            for (int i = 2; i <= 5; i++) {
                if (s[i] == 'o') ch += t;
                t >>= 1;
            }
            for (int i = 7; i <= 9; i++) {
                if (s[i] == 'o') ch += t;
                t >>= 1;
            }
            cout << ch;
        }
    }
}
```

1.112 UVa 10879: Code Refactoring

```
#include <iostream>
#include <cmath>
#include <vector>

using namespace std;

int main() {
        int t;
        cin >> t;
        for (int n = 1; n <= t; n++) {
                int num;
                cin >> num;
                int c = 0;
                vector<int> facs;
                for (int i = 2; i <= (int)sqrt(num); i++) {

                        if (num%i == 0) {
```

```
                        facs.push_back(i);
                        facs.push_back(num/i);
                        c++;
                        if (c == 2) break;
                    }
                }
                cout << "Case␣#" << n << ":␣" << num << "␣=␣" << facs[0] << "␣*␣" << facs[1] << ↩
                    "␣=␣" << facs[2] << "␣*␣" << facs[3] << endl;
        }
}
```

1.113 UVa 10905: Children's Game

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

using namespace std;

bool cmp(string a, string b) {
        return b+a < a+b;
}

int main() {
        int n;
        while (cin >> n && n) {
                string nums[n];
                for (int i = 0; i < n; i++) cin >> nums[i];
                sort(nums, nums+n, cmp);
                for (int i = 0; i < n; i++) cout << nums[i];
                cout << endl;
        }
}
```

1.114 UVa 10919: Prerequisites?

```
#include <iostream>
#include <string>
#include <map>

using namespace std;

int main() {
    int k;
    while (cin >> k && k) {
        int m; cin >> m;
        map<string, int> c;
        for (int i = 0; i < k; i++) {
            string s; cin >> s;
            c[s] = 0;
        }
        bool isit = true;
        for (int i = 0; i < m; i++) {
            int r, t; cin >> r >> t;
            for (int j = 0; j < r; j++) {
                string s; cin >> s;
                if (c.find(s) != c.end()) t--;
            }
            if (t > 0) isit = false;
        }
        cout << (isit ? "yes" : "no") << endl;
```

```
    }
}
```

1.115 UVa 10924: Prime Words

```
#include <iostream>
#include <cmath>
#include <string>
#include <cctype>

using namespace std;

bool isprime(int n) {
    if (n == 2) return true;
    else if (n%2 == 0) return false;
    else {
        for (int i = 3; i <= (int)sqrt(n); i++) {
            if (n%i == 0) return false;
        }
        return true;
    }
}

int letter(char l) {
    if (islower(l)) return l-'a'+1;
    else return l-'A'+27;
}

int main() {
    string s;
    while (cin >> s) {
        int sum = 0;
        for (int i = 0; i < s.length(); i++) {
            sum += letter(s[i]);
        }
        cout << "It␣is␣";
        if (!isprime(sum)) cout << "not␣";
        cout << "a␣prime␣word." << endl;
    }
}
```

1.116 UVa 10921: Find the Telephone

```
#include <iostream>
#include <string>
#include <map>

using namespace std;

int main() {
    map<char,int> keys;
    keys['A'] = keys['B'] = keys['C'] = 2;
    keys['D'] = keys['E'] = keys['F'] = 3;
    keys['G'] = keys['H'] = keys['I'] = 4;
    keys['J'] = keys['K'] = keys['L'] = 5;
    keys['M'] = keys['N'] = keys['O'] = 6;
    keys['P'] = keys['Q'] = keys['R'] = keys['S'] = 7;
    keys['T'] = keys['U'] = keys['V'] = 8;
    keys['W'] = keys['X'] = keys['Y'] = keys['Z'] = 9;
```

```
    string s;
    while (getline(cin, s)) {
        for (int i = 0; i < s.length(); i++) {
            if (keys.find(s[i]) != keys.end()) cout << keys[s[i]];
            else cout << s[i];
        }
        cout << endl;
    }
}
```

1.117 UVa 10929: You can say 11

```
#include <iostream>
#include <string>
#include <sstream>

using namespace std;

int bigmod(string num, int m) {
    stringstream nn(num);
    int res = 0;
    char d;
    while (nn >> d) {
        if (d == '\n') break;
        stringstream ds;
        ds << d;
        int dd; ds >> dd;
        res = (10*res + dd)%m;
    }
    return res;
}

int main() {
    string num;
    while (cin >> num) {
        if (num == "0") break;
        cout << num << " is ";
        if (bigmod(num, 11) != 0) cout << "not ";
        cout << "a multiple of 11." << endl;
    }

}
```

1.118 UVa 10931: Parity

```
#include <iostream>
#include <bitset>

using namespace std;

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        bitset<32> b(n);
        string bin = b.to_string();
        cout << "The parity of " << bin.substr(bin.find('1'));
        cout << " is " << b.count() << " (mod 2)." << endl;
```

```
    }
}
```

1.119 UVa 10935: Throwing Cards Away I

```
#include <iostream>
#include <queue>
#include <vector>

using namespace std;

void printvector(vector<int> v) {
    cout << "␣";
    for (int i = 0; i < v.size(); i++) {
        cout << v[i];
        if (i != v.size()-1) cout << ",␣";
    }
}

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        queue<int> cards;
        vector<int> discards;
        for (int i = 1; i <= n; i++) cards.push(i);
        while (cards.size() > 1) {
            discards.push_back(cards.front());
            cards.pop();
            cards.push(cards.front());
            cards.pop();
        }
        cout << "Discarded␣cards:";
        if (discards.size() > 0) printvector(discards);
        cout << endl;
        cout << "Remaining␣card:␣" << cards.front() << endl;
    }
}
```

1.120 UVa 10940: Throwing Cards Away II

```
#include <iostream>

using namespace std;
int a[500010];

int main() {
    int n;
    a[1] = 1; a[2] = 2;
    for (int i = 3; i <= 500000; i++) {
        if (i < a[i-1]+2) a[i] = 2;
        else a[i] = a[i-1]+2;
    }
    while (cin >> n) {
        if (n == 0) break;
        cout << a[n] << endl;
    }
}
```

1.121 UVa 10945: Mother Bear

```
#include <iostream>
#include <string>
#include <cctype>
#include <algorithm>

using namespace std;

bool check(string s) {
    string ss = "";
    for (int i = 0; i < s.length(); i++) {
        if (isalpha(s[i])) ss += tolower(s[i]);
    }
    string rev(ss);
    reverse(rev.begin(), rev.end());
    return ss.compare(rev) == 0;
}

int main() {
    string s;
    while (getline(cin, s)) {
        if (s.compare("DONE") == 0) break;
        if (check(s)) cout << "You␣won't␣be␣eaten!";
        else cout << "Uh␣oh..";
        cout << endl;
    }
}
```

1.122 UVa 10954: Add All

```
#include <iostream>
#include <queue>

using namespace std;

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        int cost = 0;
        priority_queue<int> adds;
        while (n--) {
            int p; cin >> p;
            adds.push(-p);
        }
        while (!adds.empty()) {
            int a = -adds.top();
            adds.pop();
            int b = -adds.top();
            adds.pop();
            cost += a+b;
            if (!adds.empty()) adds.push(-a-b);
        }
        cout << cost << endl;
    }
}
```

1.123 UVa 10970: Big Chocolate

```
#include <iostream>

using namespace std;

int main() {
    int m, n;
    while (cin >> m >> n) {
        cout << (m*n-1) << endl;
    }
}
```

1.124 UVa 10976: Fractions Again?!

```
#include <iostream>
#include <vector>
#include <utility>

using namespace std;

int main() {
    int k;
    while (cin >> k) {
        vector<pair<int,int> > den;
        for (int x = k+1; x <= 2*k; x++) {
            int y = (k*x)/(x-k);
            if (x*y == k*(x+y)) den.push_back(make_pair(y, x));
        }
        cout << den.size() << endl;
        for (int i = 0; i < den.size(); i++) {
            cout << "1/" << k << " = 1/" << den[i].first << " + 1/" << den[i].second << endl;
        }
    }
}
```

1.125 UVa 11044: Searching for Nessy

```
#include <iostream>

using namespace std;

int main() {
        int t; cin >> t;
        for (int i = 1; i <= t; i++) {
                int n; int m; cin >> n >> m;
                cout << (n/3)*(m/3) << endl;
        }
}
```

1.126 UVa 11150: Cola

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    int n;
```

```
    while (cin >> n) cout << 3*n/2 << endl;
}
```

1.127 UVa 11173: Gray Codes

```
#include <iostream>

using namespace std;

unsigned int i2g(int num) {
    return num ^ (num >> 1);
}

int main() {
  int t; cin >> t;
  while (t--) {
    int n, k; cin >> n >> k;
    cout << i2g(k) << endl;
  }
}
```

1.128 UVa 11192: Group Reverse

```
#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        string s; cin >> s;
        int l = s.length()/n;
        for (int i = 0; i < n; i++) reverse(s.begin()+l*i, s.begin()+l*(i+1));
        cout << s << endl;
    }
```

1.129 UVa 11264: Coin Collector

```
#include <iostream>

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int n; cin >> n;
        int d[n];
        for (int i = 0; i < n; i++) cin >> d[i];
        int sum = d[0], s = 1;
        for (int i = 1; i < n-1; i++) {
            if (d[i]+sum < d[i+1]) {
                sum += d[i];
                s++;
            }
        }
        cout << s+1 << endl;
```

```
    }
}
```

1.130 UVa 11292: The Dragon of Loowater

```
#include <iostream>
#include <queue>

using namespace std;

int main() {
    int n, m;
    while (cin >> n >> m) {
        if (n == 0 && m == 0) break;
        priority_queue<int> d, k;
        for (int i = 0; i < n; i++) {
            int dd; cin >> dd;
            d.push(-dd);
        }
        for (int i = 0; i < m; i++) {
            int kk; cin >> kk;
            k.push(-kk);
        }
        int paid = 0;
        while (!k.empty()) {
            if (d.top() >= k.top()) {
                d.pop();
                paid -= k.top();
                k.pop();
            } else k.pop();
            if (d.empty()) break;
        }
        if (!d.empty()) cout << "Loowater␣is␣doomed!";
        else cout << paid;
        cout << endl;
    }
}
```

1.131 UVa 11321: Sort! Sort!! and Sort!!!

```
#include <iostream>
#include <algorithm>
#include <vector>

using namespace std;

int m;
bool cmp(int a, int b) {
    if (a%m == b%m) {
        if (abs(a%2) != abs(b%2)) return abs(a%2) > abs(b%2);
        else if (abs(a%2) == abs(b%2) && abs(a%2) == 1) return a > b;
        else if (abs(a%2) == abs(b%2) && a%2 == 0) return a < b;
    } else return a%m < b%m;
}

int main() {
    int n;
    while (cin >> n >> m) {
        cout << n << "␣" << m << endl;
```

```
        if (n == 0 && m == 0) break;
        int a[n];
        for (int i = 0; i < n; i++) cin >> a[i];
        sort(a, a+n, cmp);
        for (int i = 0; i < n; i++) cout << a[i] << endl;
    }
}
```

1.132 UVa 11332: Summing Digits

```
#include <iostream>

using namespace std;
typedef long long ll;

int main() {
    ll n;
    while (cin >> n) {
        if (n == 0) break;
        ll dr = ((n-1)%9) + 1;
        cout << dr << endl;
    }
}
```

1.133 UVa 11349: Symmetric Matrix

```
#include <iostream>

using namespace std;

long long M[101][101];

int main() {
    int t, n;
    char dum1, dum2;
    cin >> t;
    for (int s = 1; s <= t; s++) {
        bool sym = true;
        cin >> dum1 >> dum2 >> n;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                cin >> M[i][j];
            }
        }
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (M[i][j] != M[n-i-1][n-j-1]) sym = false;
                if (M[i][j] < 0) sym = false;
            }
        }
        cout << "Test␣#" << s << ":␣";
        if (sym) cout << "Symmetric." << endl;
        else cout << "Non-symmetric." << endl;
    }
}
```

1.134 UVa 11364: Parking

```
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int n, mn = 100, mx = 0; cin >> n;
        while (n--) {
            int i; cin >> i;
            mn = min(mn, i);
            mx = max(mx, i);
        }
        cout << 2*(mx-mn) << endl;
    }
}
```

1.135 UVa 11371: Number Theory for Newbies

```
#include <iostream>
#include <algorithm>
#include <string>
#include <sstream>

using namespace std;
typedef long long ll;

ll parse_ll(string s) {
        istringstream ss(s);
        ll n; ss >> n;
        return n;
}

int main() {
        string n;
        while (cin >> n) {
                sort(n.begin(), n.end());
                while (n[0] == '0') next_permutation(n.begin(), n.end());
                ll a = parse_ll(n);
                sort(n.begin(), n.end());
                reverse(n.begin(), n.end());
                ll b = parse_ll(n);
                ll n = b - a;
                cout << b << " - " << a << " = " << n << " = 9 * " << n/9 << endl;
        }
}
```

1.136 UVa 11388: GCD LCM

```
#include <iostream>
using namespace std;

int main() {
        int t; cin >> t;
        for (int i = 1; i <= t; i++) {
                long long a, b; cin >> a >> b;
                if (b%a == 0) {
                        cout << a << " " << b << endl;
```

```
            } else cout << -1 << endl;
        }
}
```

1.137 UVa 11389: The Bus Driver Problem

```
#include <iostream>
#include <algorithm>

using namespace std;

int main() {
    int n, d, r;
    while (cin >> n >> d >> r) {
        if (n == 0 && d == 0 && r == 0) break;
        int m[n], a[n], ot = 0;
        for (int i = 0; i < n; i++) cin >> m[i];
        for (int i = 0; i < n; i++) cin >> a[i];
        sort(m, m+n);
        sort(a, a+n, greater<int>());
        for (int i = 0; i < n; i++) {
            if (m[i]+a[i] > d) ot += m[i]+a[i]-d;
        }
        cout << r*ot << endl;
    }
}
```

1.138 UVa 11417: GCD

```
#include <iostream>

using namespace std;

int gcd(int a, int b) {
    return (b==0) ? a : gcd(b, a%b);
}

int G(int n) {
    int g = 0;
    for (int i = 1; i < n; i++)
        for (int j = i+1; j <= n; j++)
            g += gcd(i,j);

    return g;
}

int main() {
    int n;
    while (cin >> n) {
        if (n == 0) break;
        cout << G(n) << endl;
    }
}
```

1.139 UVa 11461: Square Numbers

```
#include <iostream>
#include <cmath>
```

```
using namespace std;

int main() {
    int a, b;
    while (cin >> a >> b) {
        if (a == 0 && b == 0) break;
        int c = 0;
        for (int i = a; i <= b; i++) {
            if ((int)sqrt(i)*(int)sqrt(i) == i) c++;
        }
        cout << c << endl;
    }
}
```

1.140 UVa 11462: Age Sort

```
#include <iostream>
#include <map>

using namespace std;

int main() {
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        int t;
        while (cin >> t && t) {
                map<int, int> a;
                while (t--) {
                        int n; cin >> n;
                        a[n]++;
                }
                for(map<int, int>::iterator it = a.begin(); it != a.end(); it++) {
                        for (int i = 0; i < it->second; i++) {
                                if (i > 0 || it != a.begin()) cout << "␣";
                                cout << it->first;
                        }
                        //if (it != a.begin()) cout << " ";
                }
                cout << "\n";
        }
}
```

1.141 UVa 11479: Is this the easiest problem?

```
#include <iostream>
#include <string>

using namespace std;

int main() {
        int t;
        cin >> t;
        for (int i = 1; i <= t; i++) {
                long long a,b,c;
                string ttype;
                cin >> a >> b >> c;
                if ((a>0 && b>0 && c>0) && (a+b>c && b+c>a && a+c>b)) {
                        if (a==b && b==c && a==c) ttype = "Equilateral";
                        else if (a!=b && b!=c && a!=c) ttype = "Scalene";
```

```
                else if (a==b || b==c || a==c) ttype = "Isosceles";
            } else ttype = "Invalid";
            cout << "Case␣" << i << ":␣" << ttype << endl;
        }
        return 0;
}
```

1.142 UVa 11496: Musical Loop

```
#include <iostream>

using namespace std;

int main() {
        int t;
        while (cin >> t) {
            if (t == 0) break;
            int peaks = 0;
            int amps[10000];
            for (int i = 0; i < t; i++) {
                int a; cin >> a;
                amps[i] = a;
            }
            for (int n = 1; n < t-1; n++) {
                if (amps[n-1] < amps[n] && amps[n] > amps[n+1]) peaks++;
                else if (amps[n-1] > amps[n] && amps[n] < amps[n+1]) peaks++;
            }
            if (amps[0] > amps[t-1] && amps[0] > amps[1]) peaks++;
        else if (amps[0] < amps[t-1] && amps[0] < amps[1]) peaks++;
        if (amps[t-1] > amps[t-2] && amps[t-1] > amps[0]) peaks++;
        else if (amps[t-1] < amps[t-2] && amps[t-1] < amps[0]) peaks++;
            cout << peaks << endl;
        }
}
```

1.143 UVa 11498: Division of Nlogonia

```
#include <iostream>
using namespace std;

int main() {
        int cases;
        cin >> cases;
        while (cases != 0) {
            int divx, divy;
            cin >> divx >> divy;
            for (int i = 1; i <= cases; i++) {
                int x,y;
                cin >> x >> y;
                if (x == divx || y == divy) cout << "divisa" << endl;
                else if (x-divx > 0 && y - divy > 0) cout << "NE" << endl;
                else if (x-divx > 0 && y - divy < 0) cout << "SE" << endl;
                else if (x-divx < 0 && y - divy > 0) cout << "NO" << endl;
                else if (x-divx < 0 && y - divy < 0) cout << "SO" << endl;
            }
            cin >> cases;
        }
        return 0;
}
```

1.144 UVa 11541: Decoding

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

using namespace std;

int main() {
    int t, c = 0; scanf("%d\n", &t);
    while (t--) {
        char l; int n; string s = "";
        while (scanf("%c%d", &l, &n) == 2) {
            for(int i = 0; i < n; i++) s += l;
        }
        cout << "Case␣" << ++c << ":␣" << s << endl;
    }
}
```

1.145 UVa 11614: Etruscan Warriors Never Play Chess

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
        int tc;
        cin >> tc;
        for (int i = 1; i <= tc; i++) {
                long long n;
                cin >> n;
                long long num = floor((sqrt(8*n+1)-1)/2);
                cout << num << endl;
        }
        return 0;
}
```

1.146 UVa 11616: Roman Numerals

```
#include <iostream>
#include <sstream>
#include <cctype>
#include <map>

using namespace std;

int parse_int(string s) {
    stringstream ss(s);
    int i; ss >> i;
    return i;
}

string i2r(int n) {
    string r[] = {"M","CM","D","CD","C","XC","L","XL","X","IX","V","IV","I"};
    int h[] = {1000,900,500,400,100,90,50,40,10,9,5,4,1};
    string rom = "";
    int i = 0;
    while (n) {
        if (n < h[i]) i++;
        else {
```

```
            n -= h[i];
            rom += r[i];
        }
    }
    return rom;
}

int r2i(string s) {
    //map<char,int> rd = {{'M',1000},{'D',500},{'C',100},{'L',50},{'X',10},{'V',5},{'I',1}};
    map<char,int> rd;
    rd['M'] = 1000; rd['D'] = 500; rd['C'] = 100; rd['L'] = 50; rd['X'] = 10; rd['V'] = 5; rd[↩
        'I'] = 1;
    int n = 0;
    for (int i = 0; i < s.length()-1; i++) {
        if (rd[s[i]] < rd[s[i+1]]) n -= rd[s[i]];
        else n += rd[s[i]];
    }
    n += rd[s[s.length()-1]];
    return n;
}

int main() {
    string s;
    while (getline(cin, s)) {
        if (isalpha(s[0])) cout << r2i(s) << endl;
        else cout << i2r(parse_int(s)) << endl;
    }
}
```

1.147 UVa 11716: Digital Fortress

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

using namespace std;

bool isperf(int n) {
    return (int)sqrt(n) * (int)sqrt(n) == n;
}

int main() {
    int t; scanf("%d\n", &t);
    while (t--) {
        string enc; getline(cin, enc);
        if (!isperf(enc.length())) cout << "INVALID" << endl;
        else {
            int k = (int)sqrt(enc.length());
            for (int i = 0; i < k; i++)
                for (int j = 0; j < k; j++)
                    cout << enc[k*j+i];
            cout << endl;
        }
    }
}
```

1.148 UVa 11723: Numbering Roads

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

using namespace std;
```

```
typedef long long ll;

int main() {
    ll n, r, c = 0;
    while (cin >> n >> r && n+r) {
        ll cnt = (n-1)/r;
        cout << "Case␣" << ++c << ":␣";
        if (cnt > 26) cout << "impossible";
        else cout << cnt;
        cout << endl;
    }
}
```

1.149 UVa 11727: Cost Cutting

```
#include <iostream>
using namespace std;

int main() {
        int cases;
        cin >> cases;
        for (int i = 1; i <= cases; i++) {
                int a,b,c;
                cin >> a >> b >> c;
                if ((a<b && b<c) || (c<b && b<a)) cout << "Case␣" << i << ":␣" << b << endl;
                else if ((a<b && b>c) && a<c) cout << "Case␣" << i << ":␣" << c << endl;
                else if ((a<b && b>c) && a>c) cout << "Case␣" << i << ":␣" << a << endl;
                else if ((a>b && b<c) && a<c) cout << "Case␣" << i << ":␣" << a << endl;
                else if ((a>b && b<c) && a>c) cout << "Case␣" << i << ":␣" << c << endl;
        }
        return 0;
}
```

1.150 UVa 11728: Alternate Task

```
#include <iostream>
#include <cmath>

using namespace std;
int divs[1001];

int divsum(int n) {
    int sum = 1;
    for (int i = 2; i <= (int)sqrt(n); i++) {
        if (n%i == 0) {
            if (n == i*i) sum += i;
            else sum += i + n/i;
        }
    }
    if (n == 1) return 1;
    return sum+n;
}

int main() {
    int n, c = 0;
    for (int i = 0; i <= 1000; i++) divs[i] = -1;
    for (int i = 1; i <= 1000; i++) {
        int d = divsum(i);
        if (d <= 1000) {
```

```
            divs[d] = i;
        }
    }
    while (cin >> n) {
        if (n == 0) break;
        cout << "Case␣" << ++c << ":␣";
        cout << divs[n] << endl;
    }
}
```

1.151 UVa 11764: Jumping Mario

```
#include <iostream>

using namespace std;

int main() {
    int t, c = 0; cin >> t;
    while (t--) {
        int n, u = 0, d = 0; cin >> n;
        int w[n];
        for (int i = 0; i < n; i++) cin >> w[i];
        for (int i = 0; i < n-1; i++) {
            if (w[i] < w[i+1]) u++;
            else if (w[i] > w[i+1]) d++;
        }
        cout << "Case␣" << ++c << ":␣" << u << "␣" << d << endl;
    }
}
```

1.152 UVa 11799: Horror Dash

```
#include <iostream>
using namespace std;

int main() {
        int cases;
        cin >> cases;
        for (int i = 1; i <= cases; i++) {
                int ic;
                cin >> ic;
                int temp = 0;
                for (int j = 1; j <= ic; j++) {
                        int inp;
                        cin >> inp;
                        if (inp > temp) temp = inp;
                }
                cout << "Case␣" << i << ":␣" << temp << endl;
        }
        return 0;
}
```

1.153 UVa 11805: Bafana Bafana

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

using namespace std;

int main() {
```

```
        int t, c = 0; cin >> t;
        while (t--) {
                int n, k, p; cin >> n >> k >> p;
                cout << "Case␣" << ++c << ":␣" << ((k+p)%n==0?n:(k+p)%n) << endl;
        }
}
```

1.154 UVa 11854: Egypt

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
        int a, b, c;
        while (cin >> a >> b >> c && a+b+c) {
                cout << ((c == hypot(a,b) || b == hypot(a,c) || a == hypot(c,b)) ? "right" : "↩
                    wrong") << endl;
        }
}
```

1.155 UVa 11875: Brick Game

```
#include <iostream>

using namespace std;

int main() {
    int t, c = 0; cin >> t;
    while (t--) {
        int n; cin >> n;
        int a[n];
        for (int i = 0; i < n; i++) cin >> a[i];
        cout << "Case␣" << ++c << ":␣" << a[n/2] << endl;
    }
}
```

1.156 UVa 11877: The Coco-Cola Store

```
#include <iostream>

using namespace std;

int main() {
    int n;
    while (cin >> n) {
        if (!n) break;
        int c = 0;
        while (n > 1) {
            if (n == 2) {
                c++;
                break;
            }
            n -= 3;
            c++;
            n++;
        }
        cout << c << endl;
```

```
    }
}
```

1.157 UVa 11933: Splitting Numbers

```
#include <iostream>

using namespace std;

int main() {
	int n;
	while (cin >> n) {
		if (n == 0) break;
		int a = 0, b = 0, i = 0, j = 0;
		while (n != 0) {
			if (1 & n) {
				if (j%2) b |= (1 << i);
				else a |= (1 << i);
				j++;
			}
			i++;
			n >>= 1;
		}
		cout << a << "␣" << b << endl;
	}
}
```

1.158 UVa 11936: The Lazy Lumberjacks

```
#include <iostream>

using namespace std;

int main() {
	int t; cin >> t;
	while (t--) {
		int a, b, c; cin >> a >> b >> c;
		cout << ((a+b>c && b+c>a && a+c>b) ? "OK" : "Wrong!!") << endl;
	}
}
```

1.159 UVa 11942: Lumberjack Sequencing

```
#include <iostream>

using namespace std;

int main() {
    int t; cin >> t;
    cout << "Lumberjacks:" << endl;
    while (t--) {
        int h[10];
        bool ord = true;
        for (int i = 0; i < 10; i++) cin >> h[i];
        for (int i = 1; i < 9; i++) {
            bool ineq1 = h[i-1] > h[i] && h[i] < h[i+1];
            bool ineq2 = h[i-1] < h[i] && h[i] > h[i+1];
            if (ineq1 || ineq2) {
                ord = false;
```

```
                break;
            }
        }
        if (ord) cout << "O";
        else cout << "Uno";
        cout << "rdered" << endl;
    }

}
```

1.160 UVa 11995: I Can Guess the Data Structure!

```
#include <iostream>
#include <stack>
#include <queue>

using namespace std;

int main() {
        int t;
        while (cin >> t) {
                stack<int> st;
                queue<int> qu;
                priority_queue<int> pq;
                bool isS = true;
                bool isQ = true;
                bool isP = true;
                for (int i = 1; i <= t; i++) {
                        int q; int n;
                        cin >> q >> n;
                        if (q == 1) {
                                st.push(n);
                                qu.push(n);
                                pq.push(n);
                        } else if (q == 2) {
                                if (st.empty() || qu.empty() || pq.empty()) {
                                        isS = false;
                                        isQ = false;
                                        isP = false;
                                } else {
                                        if (st.top() == n) {
                                                st.pop();
                                        } else {
                                                isS = false;
                                        }
                                        if (qu.front() == n) {
                                                qu.pop();
                                        } else {
                                                isQ = false;
                                        }
                                        if (pq.top() == n) {
                                                pq.pop();
                                        } else {
                                                isP = false;
                                        }
                                }
                        }
                }
                if (isS && !(isQ || isP)) cout << "stack" << endl;
```

```
                else if (isQ && !(isS || isP)) cout << "queue" << endl;
                else if (isP && !(isQ || isS)) cout << "priority queue" << endl;
                else if (!(isS || isQ || isP)) cout << "impossible" << endl;
                else cout << "not sure" << endl;
        }
}
```

1.161 UVa 12004: Bubble Sort

```
#include <iostream>

using namespace std;
typedef long long ll;

int main() {
    int t, c = 0; cin >> t;
    while (t--) {
        ll n; cin >> n;
        ll T = (n*n-n)/2;
        cout << "Case " << ++c << ": ";
        if (T%2 == 0) cout << T/2 << endl;
        else cout << T << "/2" << endl;
    }
}
```

1.162 UVa 12015: Google is Feeling Lucky

```
#include <iostream>
#include <string>
#include <algorithm>
#include <utility>

using namespace std;

int main() {
        int t, c = 0; cin >> t;
        while (t--) {
                pair<string, int> rel[10];
                int mx = 0;
                for (int i = 0; i < 10; i++) {
                        string w; int r;
                        cin >> w >> r;
                        rel[i] = make_pair(w, r);
                        mx = max(mx, r);
                }
                cout << "Case #" << ++c << ":" << endl;
                for (int i = 0; i < 10; i++) {
                        if (rel[i].second == mx) cout << rel[i].first << endl;
                }
        }
}
```

1.163 UVa 12019: Doom's Day Algorithm

```
#include <iostream>
#include <string>

using namespace std;
string names[] = {"Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"};
```

```
struct date {
    int y; int m; int d;
    date(int yy, int mm, int dd): y(yy), m(mm), d(dd) {};
    friend bool operator<(date d1, date d2) {
        if (d1.y == d2.y) {
            if (d1.m == d2.m) {
                return d1.d < d2.d;
            } else return d1.m < d2.m;
        } else return d1.y < d2.y;
    }
    friend bool operator==(date d1, date d2) {
        return (d1.y == d2.y && d1.m == d2.m && d1.d == d2.d);
    }
    friend bool operator>(date d1, date d2) {
        return !(d1 < d2);
    }
};

int dow(date dd) {
    date gg = date(1582, 10, 5);
    int year = dd.y, month = dd.m, day = dd.d;
    int a = (14-month)/12;
    int y = year-a;
    int m = month + 12*a - 2;
    if (gg > dd) return (5 + day + y + y/4 + (31*m)/12) % 7;
    else return (day + y + y/4 - y/100 + y/400 + (31*m)/12) % 7;
}

int main() {
    int t; cin >> t;
    while (t--) {
        int M, D; cin >> M >> D;
        date dd = date(2011, M, D);
        cout << names[dow(dd)] << endl;
    }
}
```

1.164 UVa 12060: All Integer Average

```
#include <iostream>
#include <cmath>

using namespace std;

int gcd(int a, int b) {
        return b == 0 ? a : gcd(b, a%b);
}

void formatFrac(int q, int r, int d, double avg) {
        int nnum = (int)floor(log10(abs(r)))+1;
        int nden = (int)floor(log10(abs(d)))+1;
        int bars = (int)max(nnum,nden);
        int barm = (int)min(nnum,nden);
        int qlen = (int)log10(abs(q))+1;
        if (r == 0) {
                if (q >= 0) cout << q << endl;
                else cout << "- " << -q << endl;
        } else if (fabs(avg) > 1) {
```

```
        if (avg > 0) {
                if (nnum < nden) {
                        for (int x = 1; x <= abs(nden-nnum); x++) cout << "␣";
                }
                for (int i = 1; i <= qlen; i++) {
                        cout << "␣";
                }
                cout << abs(r) << endl;
                cout << q;
                for (int k = 1; k <= bars; k++) {
                        cout << "-";
                }
                cout << endl;
                if (nnum > nden) {
                        for (int x = 1; x <= abs(nden-nnum); x++) cout << "␣";
                }
                for (int l = 1; l <= qlen; l++) {
                        cout << "␣";
                }
                cout << abs(d) << endl;
        } else if (avg < 0) {
                cout << "␣␣";
                if (nnum < nden) {
                        for (int x = 1; x <= abs(nden-nnum); x++) cout << "␣";
                }
                for (int i = 1; i <= qlen; i++) {
                        cout << "␣";
                }
                cout << abs(r) << endl;
                cout << "-␣" << abs(q);
                for (int k = 1; k <= bars; k++) {
                        cout << "-";
                }
                cout << endl;
                cout << "␣␣";
                if (nnum > nden) {
                        for (int x = 1; x <= abs(nden-nnum); x++) cout << "␣";
                }
                for (int l = 1; l <= qlen; l++) {
                        cout << "␣";
                }
                cout << abs(d) << endl;
        }
} else if (fabs(avg) < 1) {
        if (avg > 0) {
                if (nnum < nden) {
                        for (int x = 1; x <= abs(nden-nnum); x++) cout << "␣";
                }
                cout << abs(r) << endl;
                for (int k = 1; k <= bars; k++) {
                        cout << "-";
                }
                cout << endl;
                if (nnum > nden) {
                        for (int x = 1; x <= abs(nden-nnum); x++) cout << "␣";
                }
                cout << abs(d) << endl;
        } else if (avg < 0) {
```

```
                        cout << "␣␣";
                        if (nnum < nden) {
                                for (int x = 1; x <= abs(nden-nnum); x++) cout << "␣";
                        }
                        cout << abs(r) << endl;
                        cout << "-␣";
                        for (int k = 1; k <= bars; k++) {
                                cout << "-";
                        }
                        cout << endl;
                        cout << "␣␣";
                        if (nnum > nden) {
                                for (int x = 1; x <= abs(nden-nnum); x++) cout << "␣";
                        }
                        cout << abs(d) << endl;
                }
        }
}
int main() {
        int t;
        int c = 0;
        while (cin >> t) {
                if (t == 0) break;
                int sum = 0;
                for (int i = 1; i <= t; i++) {
                        int n; cin >> n;
                        sum += n;
                }
                int indiv = sum/t;
                double avg = (double)sum/t;
                int num = (sum%t)/gcd(sum,t);
                int den = t/gcd(sum,t);
                cout << "Case␣" << ++c << ":" << endl;
                formatFrac(indiv, num, den, avg);
        }
}
```

1.165 UVa 12149: Feynman

```
#include <iostream>

using namespace std;

int main() {
        int n;
        cin >> n;
        while (n != 0) {
                int num = n*(n+1)*(2*n+1)/6;
                cout << num << endl;
                cin >> n;
        }
        return 0;
}
```

1.166 UVa 12157: Tariff Plan

```
#include <iostream>

using namespace std;
```

```
int main() {
	int t, c = 0; cin >> t;
	while (t--) {
		int n, m = 0, j = 0; cin >> n;
		while (n--) {
			int s; cin >> s;
			m += 10 + 10*(s/30);
			j += 15 + 15*(s/60);
		}
		cout << "Case␣" << ++c << ":␣";
		if (m < j) cout << "Mile␣" << m << endl;
		else if (m > j) cout << "Juice␣" << j << endl;
		else cout << "Mile␣Juice␣" << m << endl;
	}
}
```

1.167 UVa 12195: Jingle Composing

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string s;
    while (getline(cin, s)) {
        if (s[0] == '*') break;
        int cnt = 0, t = 0;
        for (int i = 1; i < s.length(); i++) {
            if (s[i] == '/') {
                if (t == 64) cnt++;
                t = 0;
            } else {
                switch (s[i]) {
                    case 'W': t += 64; break;
                    case 'H': t += 32; break;
                    case 'Q': t += 16; break;
                    case 'E': t += 8; break;
                    case 'S': t += 4; break;
                    case 'T': t += 2; break;
                    case 'X': t += 1; break;
                }
            }
        }
        cout << cnt << endl;
    }
}
```

1.168 UVa 12279: Emoogle Balance

```
#include <iostream>

using namespace std;

int main() {
	int tc, count;
	cin >> tc;
	count = 0;
```

```
        while (tc != 0) {
                count++;
                int evt, trt, notrt;
                trt = 0; notrt = 0;
                for (int i = 1; i <= tc; i++) {
                        cin >> evt;
                        if (evt == 0) trt++;
                        else notrt++;
                }
                cout << "Case␣" << count << ":␣" << notrt-trt << endl;
                cin >> tc;
        }
        return 0;
}
```

1.169 UVa 12289: One-Two-Three

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        string s; cin >> s;
        if (s.length() == 5) cout << 3 << endl;
        else {
            if ((s[0]=='o' && s[1]=='n')||(s[2]=='e' && s[1]=='n')||(s[0]=='o' && s[2]=='e')) ↩
                cout << 1 << endl;
            else cout << 2 << endl;
        }
    }
}
```

1.170 UVa 12345: Dynamic `len(set(a[L:R]))`

```
#include <iostream>
#include <cstdio>

using namespace std;

int a[50000];
int ls[1000000];

void query(int x, int y) {
        int res = 0;
        for (int i = x; i < y; i++) {
                if (!ls[a[i]]) res++;
                ls[a[i]]++;
        }
        printf("%d\n",res);
        for (int i = x; i < y; i++) {
                ls[a[i]]--;
        }
}

int main() {
        int n, m;
```

```
        cin >> n >> m;
        for (int i = 0; i < n; i++) {
                cin >> a[i];
        }
        for (int j = 1; j <= m; j++) {
                char fn; int l; int r;
                scanf("\n%c",&fn);
                if (fn == 'M') {
                        int x;
                        scanf("%d",&x);
                        scanf("%d",&a[x]);
                }
                if (fn == 'Q') {
                        scanf("%d%d",&l,&r);
                        query(l,r);
                }
        }
}
```

1.171 UVa 12372: Packing for Holiday

```
#include <iostream>

using namespace std;

int main() {
    int t, c = 0; cin >> t;
    while (t--) {
        int l, w, h; cin >> l >> w >> h;
        cout << "Case␣" << ++c << ":␣";
        if (l>20 || w>20 || h>20) cout << "bad";
        else cout << "good";
        cout << endl;
    }
}
```

1.172 UVa 12397: Roman Numerals

```
#include <iostream>
#include <string>
#include <map>

using namespace std;

string i2r(int n) {
    string r[] = {"M","CM","D","CD","C","XC","L","XL","X","IX","V","IV","I"};
    int h[] = {1000,900,500,400,100,90,50,40,10,9,5,4,1};
    string rom = "";
    int i = 0;
    while (n) {
        if (n < h[i]) i++;
        else {
            n -= h[i];
            rom += r[i];
        }
    }
    return rom;
}
```

```
int matchsticks(string rom) {
    map<char,int> rd;
    rd['M'] = 4; rd['D'] = 3; rd['C'] = 2; rd['L'] = 2;
    rd['X'] = 2; rd['V'] = 2; rd['I'] = 1;
    int n = 0;
    for (int i = 0; i < rom.length(); i++) n += rd[rom[i]];
    return n;
}

int main() {
    int n;
    while (cin >> n) {
        cout << matchsticks(i2r(n)) << endl;
    }
}
```

1.173 UVa 12403: Save Setu

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    int t, sum = 0; cin >> t;
    while (t--) {
        string cmd; int k;
        cin >> cmd;
        if (cmd.compare("donate") == 0) {
            cin >> k;
            sum += k;
        }
        else if (cmd.compare("report") == 0) cout << sum << endl;
    }
}
```

1.174 UVa 12405: Scarecrow

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    int t, c = 0; cin >> t;
    while (t--) {
        int n, sc = 0; cin >> n;
        char spots[n];

        for (int i = 0; i < n; i++) {
            cin >> spots[i];
        }
        for (int i = 0; i < n;) {
            if (spots[i] == '#') i++;
            else {
                sc++;
                i += 3;
            }
        }
```

```
        cout << "Case␣" << ++c << ":␣" << sc << endl;
    }
}
```

1.175 UVa 12468: Zapping

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    int a, b;
    while (cin >> a >> b) {
        if (a == -1 && b == -1) break;
        int d = abs(a-b);
        cout << ((d>50)?100-d:d) << endl;
    }
}
```

1.176 UVa 12478: Hardest Problem Ever (Easy)

```
#include <iostream>

using namespace std;

int main() {
    cout << "KABIR" << endl;
}
```

1.177 UVa 12502: Three Families

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

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int x, y, z;
        cin >> x >> y >> z;
        cout << z*(2*x-y)/(x+y) << endl;
    }
}
```

1.178 UVa 12503: Robot Instructions

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        int n, fin = 0; cin >> n;
        int steps[n];
        for (int i = 0; i < n; i++) {
            string cmd; cin >> cmd;
```

```
            if (cmd.compare("LEFT") == 0) {
                steps[i] = -1;
                fin--;
            } else if (cmd.compare("RIGHT") == 0) {
                steps[i] = 1;
                fin++;
            } if (cmd.compare("SAME") == 0) {
                string dum; int pos; cin >> dum >> pos;
                steps[i] = steps[pos-1];
                fin += steps[pos-1];
            }
        }
        cout << fin << endl;
    }
}
```

1.179 UVa 12542: Prime Substring

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

using namespace std;

int parseint(string s) {
    int i; istringstream(s) >> i;
    return i;
}

string tostring(int i) {
    ostringstream s; s << i;
    return s.str();
}

bool isprime(int n) {
    if (n == 2) return true;
    else if (n%2 == 0) return false;
    else {
        for (int i = 3; i <= (int)sqrt(n); i += 2) {
            if (n%i == 0) return false;
        }
        return true;
    }
}

int main() {
    string n;
    while (cin >> n && n.compare("0") != 0) {
        int maxprime = 0;
        for (int k = 0; k < 5; k++) {
            for (int i = 0; i < n.length()-k; i++) {
                int subn = parseint(n.substr(i, k+1));
                if (isprime(subn)) maxprime = max(maxprime, subn);
            }
        }
        cout << maxprime << endl;
    }
}
```

1.180 UVa 12554: A Special "Happy Birthday" Song!!!

```
#include <iostream>
#include <cmath>
#include <string>

using namespace std;
string hbd[16] = {"Happy","birthday","to","you","Happy","birthday","to","you","Happy","↩
    birthday","to","Rujia","Happy","birthday","to","you"};

int main() {
    int n; cin >> n;
    int reps = (int)ceil(n/16.0);
    string names[n];
    for (int i = 0; i < n; i++) cin >> names[i];
    for (int i = 0; i < 16*reps; i++) {
        cout << names[i%n] << ": " << hbd[i%16] << endl;
    }
}
```

1.181 UVa 12575: Sin Cos Problem

```
#include <iostream>
#include <cmath>
#include <cstdio>

using namespace std;
const double pi = acos(-1);
const double eps = 1e-6;

int main() {
        int t; cin >> t;
        while (t--) {
                int a, b; cin >> a >> b;
                double c = hypot(a, b);
                double phi = atan2(b, a);
                double theta = (pi/2)-phi;
                while (theta > eps) theta -= 2*pi;
                while (theta < -eps) theta += 2*pi;
        if (a == 0 && b == 0) theta = 0;
                double mx = c*sin(theta+phi);
                printf("%.2f %.2f\n", theta, mx);
        }
}
```

1.182 UVa 12577: Hajj-e-Akbar

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string line; int c = 0;
    while (getline(cin, line)) {
        if (line.compare("*") == 0) break;
        cout << "Case " << ++c << ": Hajj-e-";
        if (line.compare("Hajj") == 0) cout << "Akbar";
        else if (line.compare("Umrah") == 0) cout << "Asghar";
        cout << endl;
    }
```

```
}
```

1.183 UVa 12578: 10 : 6 : 2

```
#include <iostream>
#include <cmath>
#include <cstdio>

using namespace std;
const double pi = acos(-1);

int main() {
	int t; cin >> t;
	while (t--) {
		double l; cin >> l;
		double w = 3*l/5.0;
		double r = l/5.0;

		double red = pi*r*r;
		double green = (l*w)-red;
		printf("%.2f␣%.2f\n", red, green);
	}
}
```

1.184 UVa 12602: Nice Licence Plates

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
	int tc;
	cin >> tc;
	for (int i = 1; i <= tc; i++){
		char a,b,c,dummy;
		int nums;
		cin >> a >> b >> c >> dummy >> nums;
		int na = (int)a-65;
		int nb = (int)b-65;
		int nc = (int)c-65;
		int ltr = na*26*26 + nb*26 + nc;
		int diff = abs(ltr-nums);
		(diff <= 100) ? cout << "nice" << endl: cout <<"not␣nice" << endl;
	}
}
```

1.185 UVa 12820: Cool Word

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

using namespace std;

int main() {
    int n, c = 0;
    while (cin >> n) {
        int cool = 0;
        while (n--) {
            string s; cin >> s;
```

```
            map<char,int> f;
            map<int,int> chk;
            for (int i = 0; i < s.length(); i++) f[s[i]]++;
            for (map<char,int>::iterator it = f.begin(); it != f.end(); it++) {
                chk[it->second]++;
            }
            if (f.size() == chk.size() && f.size() >= 2) cool++;
        }
        cout << "Case␣" << ++c << ":␣" << cool << endl;
    }
}
```

1.186 UVa 12854: Automated Checking Machine

```
#include <iostream>

using namespace std;

int main() {
        int a[5], b[5];
        while (cin >> a[0] >> a[1] >> a[2] >> a[3] >> a[4]) {
                cin >> b[0] >> b[1] >> b[2] >> b[3] >> b[4];
                bool comp = true;
                for (int i = 0; i < 5; i++) comp &= a[i]^b[i];
                cout << (comp?"Y":"N") << endl;
        }
}
```

1.187 UVa 12893: Count It!

Note! Bitset constructor for `long long` only supported starting C++11

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

using namespace std;

int main() {
    int t; cin >> t;
    while (t--) {
        long long n; cin >> n;
        bitset<65> b(n);
        cout << b.count() << endl;
    }
}
```

1.188 UVa 12895: Armstrong Number

```
#include <iostream>
#include <cstdio>
#include <string>
#include <cmath>
#include <sstream>

using namespace std;

bool armstrong(string num) {
    int n = num.length();
    long long sum = 0;
```

```
    for (int i = 0; i < n; i++) {
        char d = num[i];
        stringstream c2i; c2i << d;
        int dd; c2i >> dd;
        sum += (long long)floor(pow(dd, n));
    }
    stringstream ss; ss << sum;
    return (num == ss.str());
}

int main() {
    int t; scanf("%d\n", &t);
    for (int i = 1; i <= t; i++) {
        string n;
        getline(cin, n);
        if (!armstrong(n)) cout << "Not␣";
        cout << "Armstrong" << endl;
    }
}
```

1.189 UVa 12896: Mobile SMS

```
#include <bits/stdc++.h>
#define FOR(i,n) for (int i = 0; i < n; i++)

using namespace std;

int main() {
    string keypad[10] = {"␣", ".,?\"", "abc", "def", "ghi", "jkl", "mno", "pqrs", "tuv", "wxyz↩
        "};
    int t; cin >> t;
    while (t--) {
        int n; cin >> n;
        int a[n], b[n];
        FOR(i, n) cin >> a[i];
        FOR(i, n) cin >> b[i];
        FOR(i, n) cout << keypad[a[i]][b[i]-1];
        cout << endl;
    }
}
```

1.190 UVa 12946: Peanoland Contacting Gaussland

```
// TODO: Find (or reconstruct) UVa 12946
```

1.191 UVa 12952: Tri-du

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

using namespace std;

int main() {
    int a, b;
    while (cin >> a >> b) cout << max(a, b) << endl;
}
```

1.192 UVa 13012: Identifying Tea

```
#include <iostream>

using namespace std;

int main() {
    int t;
    while (cin >> t) {
        int c = 0;
        for (int i = 0; i < 5; i++) {
            int n; cin >> n;
            if (t == n) c++;
        }
        cout << c << endl;
    }
}
```

1.193 UVa 13025: Back to the Past

```
#include <iostream>

using namespace std;

int main() {
    cout << "May␣29,␣2013␣Wednesday" << endl;
}
```

1.194 UVa 13026: Search the Khoj

```
#include <bits/stdc++.h>
#define GETINT(x) int x; cin >> x;
#define FOR(i, n) for (int i = 0; i < n; i++)

using namespace std;

int min(int a, int b, int c) {
    return min(a, min(b, c));
}

int edit(string s1, string s2) {
    int m = s1.length(), n = s2.length();
    int memo[m+1][n+1];
    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0) memo[i][j] = j;
            else if (j == 0) memo[i][j] = i;
            else if (s1[i-1] == s2[j-1]) memo[i][j] = memo[i-1][j-1];
            else memo[i][j] = 1 + min(memo[i][j-1], memo[i-1][j], memo[i-1][j-1]);
        }
    }
    return memo[m][n];
}

int main() {
    GETINT(t); int c = 0;
    while (t--) {
        GETINT(n);
        string nums[n];
        FOR(i, n) cin >> nums[i];
        string mom; cin >> mom;
```

```
        cout << "Case␣" << ++c << ":" << endl;
        FOR(i, n) {
            int x = nums[i].length(), y = mom.length();
            //cerr << "Edit distance of " << edit(nums[i], mom) << " between " << nums[i] << " ↩
                and " << mom << endl;
            if (edit(nums[i], mom) <= 1 && x == y) cout << nums[i] << endl;
        }
    }
}
```

1.195 UVa 13031: Geek Power Inc.

```
#include <iostream>
#include <algorithm>
#include <utility>

using namespace std;
typedef long long ll;

bool cmp(pair<int,int> a, pair<int,int> b) {
        return a.second > b.second;
}

int main() {
        int t, c = 0; cin >> t;
        while (t--) {
                int n; cin >> n;
                pair<int,int> src[n];
                for (int i = 0; i < n; i++) {
                        int k, p; cin >> k >> p;
                        src[i] = make_pair(k, p);
                }
                sort(src, src+n, cmp);
                ll maxout = 0, minpw = 1010, cnt = 0;
                for (int i = 0; i < n; i++) {
                        cnt += src[i].first;
                        minpw = min(minpw, (ll)src[i].second);
                        maxout = max(maxout, (ll)cnt*minpw);
                }
                cout << "Case␣" << ++c << ":␣" << maxout << endl;
        }
}
```

1.196 UVa 13034: Solve Everything :-)

```
#include <iostream>

using namespace std;

int main() {
    int t, c = 0; cin >> t;
    while (t--) {
        bool solvable = true;
        for (int i = 0; i < 13; i++) {
            int n; cin >> n;
            //cerr << " debug: " << n << endl;
            solvable &= (n != 0);
        }
        cout << "Set␣#" << ++c << ":␣";
```

```
        cout << ((solvable) ? "Yes" : "No") << endl;
    }
}
```

1.197 UVa 13059: Tennis Championship

```
#include <iostream>

using namespace std;
typedef long long ll;

int main() {
        ll n;
        while (cin >> n) {
                cout << n-1 << endl;
        }
}
```

1.198 UVa 13093: Acronyms

```
#include <iostream>
#include <sstream>
#include <string>
#include <vector>

using namespace std;

vector<string> split(string s) {
        vector<string> vs; string ss;
        istringstream iss(s);
        while (iss >> ss) vs.push_back(ss);
        return vs;
}

int main() {
        string l1;
        while (getline(cin, l1)) {
                string l2; getline(cin, l2);
                vector<string> s1 = split(l1), s2 = split(l2);
                bool match = true;
                if (s1.size() != s2.size()) {
                        match = false;
                } else {
                        for (int i = 0; i < s1.size(); i++) {
                                match &= s1[i][0] == s2[i][0];
                        }
                }
                cout << (match ? "yes" : "no") << endl;
        }
}
```

1.199 UVa 13096: Standard Deviation

```
#include <iostream>
#include <cstdio>
#include <cmath>

using namespace std;
```

```
int main() {
	double n;
	while (cin >> n && n) {
		double sd = sqrt((n*n+n)/3.0);
		printf("%.6f\n", sd);
	}
}
```

1.200 UVa 13099: Tobby and the Line Game

```
#include <iostream>
#include <cstdio>
#include <cmath>

using namespace std;

int main() {
	double x1, y1, x2, y2;
	while (cin >> x1 >> y1 >> x2 >> y2) {
		double x = pow(x1-x2, 2);
		double y = pow(y1-y2, 2);
		double E = (x+y)/6;
		printf("%.8f\n", E);
	}
}
```

1.201 UVa 13107: Royale with Cheese

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

using namespace std;

string stringify(int i) {
    ostringstream os; os << i;
    return os.str();
}

int main() {
    string s;
    while (cin >> s) {
        string id = "", id2 = "";
        map<char,int> ns; int idx = 0;
        for (int i = 0; i < s.length(); i++) {
            if (ns.find(s[i]) == ns.end()) {
                ns[s[i]] = ++idx;
            }
        }
        for (int i = 0; i < s.length(); i++) {
            id += stringify(ns[s[i]]);
        }
        replace(id.begin(), id.end(), '2', '#');
        replace(id.begin(), id.end(), '5', '2');
        replace(id.begin(), id.end(), '#', '5');
        replace(id.begin(), id.end(), '6', '$');
        replace(id.begin(), id.end(), '9', '6');
        replace(id.begin(), id.end(), '$', '9');
        cout << id << endl;
    }
}
```

1.202 UVa 13108: Juanma and the Drinking Fountains

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

using namespace std;
typedef long long ll;

int c[201][201];
int binom(int n, int k) {
    int i, j;
    for (i = 0; i <= n; i++) {
        for (j = 0; j <= min(i,k); j++) {
            if (j == 0 || j == i) c[i][j] = 1;
            else c[i][j] = c[i-1][j-1] + c[i-1][j];
        }
    }
    return c[n][k];
}

int main() {
        binom(200,200);
        int t; cin >> t;
        while (t--) {
                int n; cin >> n;
                int res = 0;
                for (int i = 0; i <= 4; i++) res += c[n-1][i];
                cout << res << endl;
        }
}
```

1.203 UVa 13109: Elephants

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

using namespace std;

int main() {
        int t; cin >> t;
        while (t--) {
                int n, m, cnt = 0; cin >> n >> m;
                while (n--) {
                        int i; cin >> i;
                        if (i <= m) cnt++;
                }
                cout << cnt << endl;
        }
}
```

2 Java

2.1 UVa 343: What Base Is This?

```
import java.util.Scanner;
import java.math.BigInteger;

public class p343 {
        public static void main(String[] args) {
                Scanner s = new Scanner(System.in);
```

```
		while (s.hasNext() && s.hasNext()) {
			String s1 = s.next(); String s2 = s.next();
			boolean found = false;
			for (int i = 2; i <= 36; i++) {
				for (int j = 2; j <= 36; j++) {
					BigInteger n1, n2;
					try {
						n1 = new BigInteger(s1, i);
					} catch (NumberFormatException e) {
						continue;
					}
					try {
						n2 = new BigInteger(s2, j);
					} catch (NumberFormatException e) {
						continue;
					}
					if ((n1.toString()).equals(n2.toString())) {
						System.out.printf("%s (base %d) = %s (base %d)\n", ↩
						    s1, i, s2, j);
						found = true;
						break;
					}

				}
				if (found) break;
			}
			if (!found) System.out.printf("%s is not equal to %s in any base 2..36\n↩
			    ", s1, s2);
		}
	}
}
```

2.2 UVa 389: Basically Speaking

```
import java.util.Scanner;
import java.math.BigInteger;

public class p389 {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		while (s.hasNext() && s.hasNext() && s.hasNext()) {
			String num = s.next(); int from = s.nextInt(); int to = s.nextInt();
			BigInteger n;
			try {
				n = new BigInteger(num, from);
			} catch (Exception e) {
				continue;
			}
			String res = n.toString(to);
			if (res.length() > 7) System.out.println("  ERROR");
			else {
				for (int x = 1; x <= 7-res.length(); x++) System.out.print(" ");
				System.out.println(res.toUpperCase());
			}
		}
	}
}
```

2.3 UVa 424: Integer Inquiry

```
import java.math.BigInteger;
import java.util.Scanner;

public class p424 {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		String num = s.nextLine();
		BigInteger sum = BigInteger.ZERO;
		while (!"0".equals(num)) {
			BigInteger n = new BigInteger(num);
			sum = sum.add(n);
			num = s.nextLine();
		}
		System.out.println(sum);
	}
}
```

2.4 UVa 495: Fibonacci Freeze

```
import java.io.*;
import java.util.*;
import java.math.*;

public class p495 {
	static BigInteger[] F = new BigInteger[5010];

	static void pregen() {
		F[0] = BigInteger.ZERO;
		F[1] = BigInteger.ONE;
		for (int i = 2; i <= 5000; i++) {
			F[i] = F[i-1].add(F[i-2]);
		}
	}
	public static void main(String[] args) throws Exception {
		BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
		String ns;
		pregen();
		while ((ns = br.readLine()) != null) {
			int n = Integer.parseInt(ns);
			BigInteger fn = F[n];
			System.out.println("The␣Fibonacci␣number␣for␣" + n + "␣is␣" + fn.↩
			    toString());
		}
	}
}
```

2.5 UVa 623: 500!

```
import java.util.Scanner;
import java.math.BigInteger;

public class p623 {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		while (s.hasNextInt()) {
			int n = s.nextInt();
			BigInteger fac = BigInteger.valueOf(1);
```

```
                    for(int i = 1; i<=n; i++) {
                            fac = fac.multiply(BigInteger.valueOf(i));
                    }
                    System.out.println(n + "!");
                    System.out.println(fac);
                }
        }
}
```

2.6 UVa 713: Adding Reversed Numbers

```
import java.io.*;
import java.util.*;
import java.math.BigInteger;
import static java.lang.System.*;

public class p713 {
    static String reverse(String s) {
        String rev = "";
        for (int i = s.length()-1; i >= 0; i--) rev += s.charAt(i);
        return rev;
    }

    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int t = Integer.parseInt(br.readLine());
        while (t-- > 0) {
            String nums[] = br.readLine().split("\\s");
            BigInteger b1 = new BigInteger(reverse(nums[0]));
            BigInteger b2 = new BigInteger(reverse(nums[1]));
            BigInteger r = new BigInteger(reverse(b1.add(b2).toString()));
            out.println(r);
        }
    }
}
```

2.7 UVa 748: Exponentiation

```
import java.io.*;
import java.math.*;

public class p748 {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String w;
        while ((w = br.readLine()) != null) {
            String[] pairs = w.split("\\s+");
            BigDecimal d = new BigDecimal(pairs[0]);
            int p = Integer.parseInt(pairs[1]);
            BigDecimal r = d.pow(p);
            System.out.println(r.stripTrailingZeros().toPlainString().replaceFirst("^0\\.", "."↩
                ));
        }
    }
}
```

2.8 UVa 893: Y3K Problem

```
import java.util.Scanner;
```

```
import java.util.Calendar;
import java.util.GregorianCalendar;
import java.text.SimpleDateFormat;

public class p893 {
	public static void main(String[] args) {
		Scanner s = new Scanner(System.in);
		int n = s.nextInt();
		int d = s.nextInt();
		int m = s.nextInt();
		int y = s.nextInt();
		while (n != 0 && d != 0 && m != 0 && y != 0) {
			Calendar c = new GregorianCalendar(y, m-1, d);
			SimpleDateFormat sdf = new SimpleDateFormat("d␣M␣yyyy");
			c.add(Calendar.DAY_OF_MONTH, n);
			System.out.println(sdf.format(c.getTime()));
			n = s.nextInt();
			d = s.nextInt();
			m = s.nextInt();
			y = s.nextInt();
		}
	}
}
```

2.9 UVa 10071: Back to High School Physics

```
import java.util.Scanner;

public class Prac_PA {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        for (int i = 0; s.hasNextLine(); i++) {
            int v = s.nextInt();
            int t = s.nextInt();
            System.out.println(2*t*v);
        }
    }

}
```

2.10 UVa 10105: Polynomial Coefficients

```
import java.util.Scanner;
import java.util.ArrayList;

public class p10105 {
    public static int factl(int num) {
        int res = 1;
        if (num == 0) {
            return 1;
        } else {
            for (int i = 1; i <= num; i++) {
                res *= i;
            }
            return res;
        }
    }

    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int k = sc.nextInt();
        while (sc.hasNextInt()) {
                int facs = 1;
                ArrayList<Integer> coeffs = new ArrayList<>();
                for (int i = 1; i <= k; i++) {
                        int xn = sc.nextInt();
                        coeffs.add(xn);
                }
                for (int coeff : coeffs) {
                        facs *= factl(coeff);
                }
                int fn = factl(n);
                System.out.println(fn/facs);
                if (sc.hasNextInt()) {
                        n = sc.nextInt();
                        k = sc.nextInt();
                }
        }
    }

    }
```

2.11 UVa 10106: Product

```
import java.math.BigInteger;
import java.util.Scanner;

public class p10106 {
        public static void main(String[] args) {
                Scanner s = new Scanner(System.in);

                while (s.hasNextBigInteger() && s.hasNextBigInteger()) {
                        BigInteger x = s.nextBigInteger(), y = s.nextBigInteger(), prod;
                        prod = x.multiply(y);
                        System.out.println(prod);
                }

        }
}
```

2.12 UVa 10193: All You Need is Love

```
import java.io.*;
import java.math.*;
import static java.lang.System.*;

public class p10193 {
    public static void main(String[] args) throws Exception{
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int t = Integer.parseInt(br.readLine()), c = 0;
        while (t-- > 0) {
            BigInteger b1 = new BigInteger(br.readLine(), 2);
            BigInteger b2 = new BigInteger(br.readLine(), 2);
            BigInteger g = b1.gcd(b2);
            out.printf("Pair␣#%d:␣", ++c);
            if ("1".equals(g.toString())) out.println("Love␣is␣not␣all␣you␣need!");
            else out.println("All␣you␣need␣is␣love!");
```

```
        }
    }
}
```

2.13 UVa 10494: If We Were a Child Again

```
import java.io.*;
import java.math.*;
import static java.lang.System.*;

public class p10494 {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String line;
        while ((line = br.readLine()) != null) {
            String[] nums = line.split("\\s+");
            BigInteger n = new BigInteger(nums[0]), res = BigInteger.ZERO;
            BigInteger p = new BigInteger(nums[2]);
            if (nums[1].equals("/")) res = n.divide(p);
            else if (nums[1].equals("%")) res = n.mod(p);
            out.println(res.toString());
        }
    }
}
```

2.14 UVa 10523: Very Easy !!!

```
import java.util.Scanner;
import java.math.BigInteger;

public class p10523 {
        public static void main(String[] args) {
                int n, a;
                Scanner s = new Scanner(System.in);

                while (s.hasNextInt() && s.hasNextInt()) {
                        n = s.nextInt();
                        a = s.nextInt();
                        BigInteger sum = new BigInteger("0");
                        BigInteger temp = new BigInteger("1");
                        for (int i = 1; i <= n; i++) {
                                temp = temp.multiply(BigInteger.valueOf(a));
                                temp = temp.pow(i);
                                temp = temp.multiply(BigInteger.valueOf(i));
                                sum = sum.add(temp);
                                temp = BigInteger.valueOf(1);
                        }
                        System.out.println(sum);
                }
        }
}
```

2.15 UVa 10814: Simplifying Fractions

```
import java.util.Scanner;
import java.math.BigInteger;

public class p10814 {
        public static void main(String[] args) {
```

```
                Scanner s = new Scanner(System.in);
                int t = s.nextInt();
                for (int i = 1; i <= t; i++) {
                        BigInteger n = s.nextBigInteger();
                        s.next();
                        BigInteger d = s.nextBigInteger();
                        BigInteger g = n.gcd(d);
                        BigInteger n2 = n.divide(g);
                        BigInteger d2 = d.divide(g);
                        System.out.printf("%s␣/␣%s\n", n2.toString(), d2.toString());
                }
        }
}
```

2.16 UVa 10925: Krakovia

```
import java.util.Scanner;
import java.math.BigInteger;

public class p10925 {
        public static void main(String[] args) {
                Scanner s = new Scanner(System.in);
                int n = s.nextInt();
                int f = s.nextInt();
                int c = 0;
                while (!(n == 0 && f == 0)) {
                        c++;
                        BigInteger sum = BigInteger.valueOf(0);
                        for (int i = 1; i <= n; i++) {
                                BigInteger price = new BigInteger(s.next());
                                sum = sum.add(price);
                        }
                        BigInteger split = sum.divide(BigInteger.valueOf(f));

                        System.out.println("Bill␣#" + c + "␣costs␣" + sum + ":␣each␣friend␣↩
                            should␣pay␣" + split);
                        System.out.println("");
                        n = s.nextInt();
                        f = s.nextInt();
                }
        }
}
```

2.17 UVa 11172: Relational Operator

```
import java.util.Scanner;

public class RelOps {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int count = s.nextInt();
        for (int i = 0; i < count; i++) {
            int a = s.nextInt();
            int b = s.nextInt();
            if (a < b) {
                System.out.println("<");
            } else if (a > b) {
                System.out.println(">");
            } else {
```

```
                System.out.println("=");
            }
        }
    }

}
```

2.18 UVa 11185: Ternary

```
import java.math.BigInteger;
import java.util.Scanner;
import static java.lang.System.*;

public class p11185 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        while (s.hasNextInt()) {
            int n = s.nextInt();
            if (n < 0) break;
            BigInteger b = BigInteger.valueOf(n);
            out.println(b.toString(3));
        }
    }
}
```

2.19 UVa 11356: Dates

```
import java.util.*;
import java.text.SimpleDateFormat;
import static java.lang.System.*;

public class p11356 {
        public static void main(String[] args) {
                HashMap<String, Integer> months = new HashMap<String, Integer>();
                months.put("January", 1);
                months.put("February", 2);
                months.put("March", 3);
                months.put("April", 4);
                months.put("May", 5);
                months.put("June", 6);
                months.put("July", 7);
                months.put("August", 8);
                months.put("September", 9);
                months.put("October", 10);
                months.put("November", 11);
                months.put("December", 12);
                Scanner s = new Scanner(System.in);
                int t = Integer.parseInt(s.nextLine()), i = 0;
                while (t-- > 0) {
                        String dt = s.nextLine();
                        String[] tokens = dt.split("-");
                        int year = Integer.parseInt(tokens[0]);
                        String month = tokens[1];
                        int day = Integer.parseInt(tokens[2]);

                        Calendar c = new GregorianCalendar(year, months.get(month)-1, day);
                        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MMMM-dd");

                        int incr = Integer.parseInt(s.nextLine());
```

```
                    c.add(Calendar.DAY_OF_MONTH, incr);
                    out.println("Case␣" + ++i + ":␣" + sdf.format(c.getTime()));
                }
        }
}
```

2.20 UVa 11547: Automatic Answer

```
import java.util.Scanner;

public class Test2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int count = input.nextInt();
        for (int i = 0; i < count; i++) {
            int res = (((((input.nextInt()*567)/9)+7492)*235)/47)-498;
            int modd = (res/10)%10;
            System.out.println(Math.abs(modd));
        }
    }
}
```

2.21 UVa 11636: Hello World!

```
import java.util.Scanner;

public class p11636 {
        public static void main (String[] args) {
                Scanner sc = new Scanner(System.in);
                int num = sc.nextInt();
                int i = 1;
                while (num >= 0) {
                        double power = Math.ceil(Math.log(num)/Math.log(2));
                        int intp = (int) power;
                        System.out.println("Case␣" + i + ":␣" + intp);
                        num = sc.nextInt();
                        i++;
                }
        }
}
```

2.22 UVa 11879: Multiple of 17

```
import java.util.Scanner;
import java.math.BigInteger;

public class p11879 {
        public static void main(String[] args) {
                Scanner s = new Scanner(System.in);
                String num = s.nextLine();
                BigInteger n = new BigInteger(num);
                while (!(num.equals("0"))) {
                        if (n.mod(BigInteger.valueOf(17)) == BigInteger.ZERO) System.out.println↩
                            ("1");
                        else System.out.println("0");
                        num = s.nextLine();
                        n = new BigInteger(num);
                }
        }
```

```
}
```

2.23 UVa 12250: Language Detection

```
import java.util.Scanner;

public class Testing {
    public static void main(String[] args) {
        Scanner inp = new Scanner(System.in);
        String lang = null;
        int i = 0;
        while (!"#".equals(inp.next())) {
            switch (inp.next()) {
                case "HELLO": lang = "ENGLISH";
                    break;
                case "HOLA": lang = "SPANISH";
                    break;
                case "HALLO": lang = "GERMAN";
                    break;
                case "BONJOUR": lang = "FRENCH";
                    break;
                case "CIAO": lang = "ITALIAN";
                    break;
                case "ZDRAVSTVUJTE": lang = "RUSSIAN";
                    break;
                default: lang = "UNKNOWN";
            }
            i++;
            System.out.println("Case␣" + (i+1) + ":␣" + lang);
        }
    }
}
```

2.24 UVa 12930: Bigger or Smaller

```
import java.io.*;
import java.math.*;

public class p12930 {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String ln; int c = 0;
        while ((ln = br.readLine()) != null) {
            BigDecimal d1 = new BigDecimal(ln.split("\\s+")[0]);
            BigDecimal d2 = new BigDecimal(ln.split("\\s+")[1]);
            System.out.print("Case␣" + ++c + ":␣");
            switch (d1.compareTo(d2)) {
                case -1: System.out.println("Smaller"); break;
                case 0: System.out.println("Same"); break;
                case 1: System.out.println("Bigger"); break;
            }
        }
    }
}
```