

T1

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

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);

    int T;
    cin >> T;
    while (T--) {
        int n;
        cin >> n;
        for (int i = 1; i <= n; ++i) {
            int _;
            cin >> _;
        }

        if (n == 1) cout << 1;
        else if (n == 2) cout << 2;
        else if (n == 3) cout << 6;
        else if (n % 4 == 0) cout << 4;
        else if (n % 4 == 1) cout << 2 * n;
        else if (n % 4 == 2) cout << n;
        else cout << 12;

        cout << '\n';
    }

    return 0;
}
```

T2

```
#include <bits/stdc++.h>
using namespace std;
const int N = 5e4 + 5;

int n, m, K, lover[13], col[N], ans[N]; //ans[i]表示所有人最快何时能都到达i点
vector<int> G[N];
int dis[N][32], vis[N][32];

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);

    int T;
    cin >> T;
    while (T--) {
        cin >> n >> m >> K;

        for (int i = 1; i <= n; ++i) //记得初始化
            G[i].clear();
    }
}
```

```

memset(ans, -1, sizeof ans);

for (int i = 1; i <= k; ++i) //每个人的位置
    cin >> lover[i];
for (int i = 1; i <= n; ++i) { //每个点的属性
    char c; cin >> c;
    if (c == 'J') col[i] = 0;
    if (c == 'M') col[i] = 1;
    if (c == 'S') col[i] = 2;
    if (c == 'H') col[i] = 3;
    if (c == 'T') col[i] = 4;
}
while (m--) {
    int u, v;
    cin >> u >> v;
    G[u].push_back(v);
    G[v].push_back(u);
}

for (int k = 1; k <= K; ++k) { //对每个人跑一遍bfs
    //bfs
    memset(dis, 0x3f, sizeof dis);
    memset(vis, 0, sizeof vis);

    int s = lover[k];
    queue<pair<int, int>> q;
    q.push({s, 1 << col[s]});
    dis[s][1 << col[s]] = 0;
    vis[s][1 << col[s]] = true;

    while (!q.empty()) {
        int u = q.front().first, s = q.front().second;
        q.pop();
        for (int v : G[u]) { //从{u,s}走到状态{v,T}
            if (s >> col[v] & 1) continue; //v的属性在前五步内走过
            int T = s | (1 << col[v]);
            if (T == (1 << 5) - 1) T = 0; //走了5种属性
            if (!vis[v][T]) {
                dis[v][T] = dis[u][s] + 1;
                vis[v][T] = true;
                q.push({v, T});
            }
        }
    }

    for (int u = 1; u <= n; ++u)
        ans[u] = max(ans[u], *min_element(dis[u], dis[u] + 32));
}

int res = 2e9;
for (int u = 1; u <= n; ++u)
    if (ans[u] != -1 && ans[u] < 1e9) res = min(res, ans[u]);
if (res < 1e9) cout << res << '\n';
else cout << "what a pity!\n";
}

return 0;
}

```

T3

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
const int N = 3e3 + 10;
int n, m, f[N], a[N][N], b[N][N], ans = 2e9;
int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);

    cin >> n >> m;
    for (int i = 1; i <= m; ++i)
        for (int j = 1; j <= n - 1; ++j)
            cin >> a[i][j];
    for (int i = 1; i <= n; ++i)
        for (int j = 1; j <= m - 1; ++j)
            cin >> b[i][j];

    memset(f, 0x3f, sizeof f);
    f[1] = 0;
    for (int i = 0; i <= m - 1; ++i) {
        if (i) for (int j = 1; j <= n; ++j)
            f[j] += b[j][i];
        for (int j = 2; j <= n; ++j)
            f[j] = min(f[j], f[j - 1] + a[i + 1][j - 1]);
        for (int j = n - 1; j >= 1; --j)
            f[j] = min(f[j], f[j + 1] + a[i + 1][j]);
        ans = min(ans, f[n]);
    }

    cout << ans;

    return 0;
}
```

T4

```
#include <bits/stdc++.h>
using namespace std;
const int N = 3e5 + 5;
int n, a[N], LG[N], lst[N], hs[N];

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

struct node {
    int mx, mn, g, mxp;
} st[N][20];

node calc(node a, node b) {
    node res;
    res.mx = max(a.mx, b.mx);
    res.mn = min(a.mn, b.mn);
```

```

        res.g = gcd(a.g, b.g);
        res.mxp = max(a.mxp, b.mxp);
        return res;
    }

int main(){
    ios::sync_with_stdio(0);
    cin.tie(0);

    cin >> n;
    for(int i = 1; i <= n; i++) {
        cin >> a[i];
        st[i][0] = {a[i], a[i], abs(a[i] - a[i - 1])};
        hs[i] = a[i];
    }
    sort(hs + 1, hs + 1 + n);
    int len = unique(hs + 1, hs + 1 + n) - hs - 1;
    for(int i = 1; i <= n; i++) {
        int id = lower_bound(hs + 1, hs + 1 + len, a[i]) - hs;
        st[i][0].mxp = lst[id], lst[id] = i;
    }

    for (int i = 2; i <= n; i++) LG[i] = LG[i / 2] + 1;
    for (int i = 1; i <= 18; i++)
        for (int j = 1; j + (1 << i) - 1 <= n; j++)
            st[j][i] = calc(st[j][i - 1], st[j + (1 << (i - 1))][i - 1]);

    int q, yescnt=0;
    cin >> q;
    while (q--) {
        int l, r, k;
        cin >> l >> r >> k;
        l ^= yescnt; r ^= yescnt; k ^= yescnt;

        node res = st[l][0];
        res.g = 0;
        if (l != r) {
            int k = LG[r - l];
            res = calc(res, calc(st[l + 1][k], st[r - (1 << k) + 1][k]));
        }

        if (res.mx - res.mn == (r - l) * k && (res.g == k || !res.g)
            && (res.mxp < l || !k)) cout << "yes\n", yescnt++; //注意公差为0时允许相等
        else cout << "no\n";
    }

    return 0;
}

```