题目不难，当热身了。

连续自然数和

因为是连续，所以构成等差数列，设首项为a_1,末项为a_n.

则这一段的和为$\frac{(a_1+a_n)\times(a_n-a_1+1)}{2}$,我们要让这个等于$m$,

即$\frac{(a_1+a_n)\times(a_n-a_1+1)}{2}=m.$

不妨设$x=a_1+a_n, y=a_n-a_1+1$,则原式可化为$x\times y=2m.$

同时又有$(x-y)+1=2\times a_1$,将两式联立，可得$\frac{2m}{y}-y+1=2\times a_1.$

所以我们可以枚举y,进而求出a_1和a_n,然后验证一下式子是否成立就行了。

复杂度$O(m)$.

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;

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

    int m; cin >> m;
    vector<array<int, 2> > ans;

    auto calc = [](int a1, int an) {
        return (a1 + an) * (an - a1 + 1) / 2LL;
    };
    for (int y = 1; y <= 2 * m; ++y) {
        if (2 * m % y) continue;
        int tem = 2 * m / y - y + 1;
        if (tem <= 0 || tem % 2) continue;
        int a1 = tem / 2;
        int an = y + a1 - 1;
        if (a1 == an) continue;
        if (calc(a1, an) == m) {
            ans.push_back({a1, an});
        }
    }
    sort(ans.begin(), ans.end());
    for (auto &&[x, y] : ans) cout << x << ' ' << y << '\n';
    return 0;
```

```
}
```

笨小猴

按题意模拟即可。

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;

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

    string s; cin >> s;
    vector<int> cnt(26);
    for (int i = 0; i < s.size(); ++i) {
        cnt[s[i] - 'a']++;
    }
    int maxx = -1, minn = (int)(1e18);
    for (int i = 0; i < 26; ++i) {
        if (cnt[i] == 0) continue;
        maxx = max(maxx, cnt[i]);
        minn = min(minn, cnt[i]);
    }

    auto check = [](int x) -> bool {
        if (x <= 1) return 0;
        if (x == 2) return 1;
        for (int i = 2; i <= sqrt(x); ++i) {
            if (x % i == 0) return 0;
        }
        return 1;
    };
    if (check(maxx - minn)) {
        cout << "Lucky Word\n" << maxx - minn << '\n';
        return 0;
    }
    cout << "No Answer\n0\n";
    return 0;
}
```

迷宫

数据很小，所以直接dfs即可。

```
```

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;

int dx[] = {1, -1, 0, 0};
int dy[] = {0, 0, 1, -1};

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

    int n, m, _; cin >> n >> m >> _;
    array<int, 2> st, ed; cin >> st[0] >> st[1] >> ed[0] >> ed[1];
    vector pass(n + 1, vector(m + 1, 1));
    vector vis(n + 1, vector(m + 1, 0));
    while (_--) {
        int x, y; cin >> x >> y;
        pass[x][y] = 0;
    }

    int ans = 0;
    auto dfs = [&](auto dfs, int x, int y) -> void {
        if (x == ed[0] && y == ed[1]) {
            ans++;
            return;
        }

        for (int i = 0; i < 4; ++i) {
            int tox = x + dx[i];
            int toy = y + dy[i];
            if (1 <= tox && tox <= n && 1 <= toy && toy <= m &&
pass[tox][toy] && !vis[tox][toy]) {
                vis[tox][toy] = 1;
                dfs(dfs, tox, toy);
                vis[tox][toy] = 0;
            }
        }
    };
    vis[st[0]][st[1]] = 1;
    dfs(dfs, st[0], st[1]);
    cout << ans << '\n';
    return 0;
}
```

合并果子

贪心，每次把最小的两堆合并，这样累加的时候一定加的是最小的。用优先队列实现即可。

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;

int dx[] = {1, -1, 0, 0};
int dy[] = {0, 0, 1, -1};

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

    int n; cin >> n;
    priority_queue<int, vector<int>, greater<int> > q;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x;
        q.push(x);
    }
    int ans = 0;
    while (q.size() >= 2) {
        int x = q.top(); q.pop();
        int y = 0;
        if (!q.empty()) {
            y = q.top(); q.pop();
        }
        ans += x + y;
        q.push(x + y);
    }
    cout << ans << '\n';
    return 0;
}
```

产生数

数据比较小，所以可以跑一遍floyd，求出一个数能够转换成其他数的个数，然后乘法原理计数即可。

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
using i128 = __int128;

int dx[] = {1, -1, 0, 0};
int dy[] = {0, 0, 1, -1};

void print(i128 x) {
    if (x < 0) {
        putchar('-');
```

```
        x = -x;
    }
    if (x > 9)
        print(x / 10);
    putchar(x % 10 + '0');
}

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

    string n;
    int k; cin >> n >> k;
    vector pass(11, vector(11, 0));
    vector<int> a(10);
    for (int i = 1; i <= k; ++i) {
        int u, v; cin >> u >> v;
        pass[u][v] = 1;
    }
    for (int i = 1; i <= 9; ++i) {
        for (int l = 0; l <= 9; ++l) {
            for (int r = 1; r <= 9; ++r)
                if (pass[l][i] && pass[i][r]) pass[l][r] = 1;
        }
    }
    for (int i = 0; i <= 9; ++i) {
        pass[i][i] = 1;
        for (int j = 0; j <= 9; ++j)
            if (pass[i][j]) a[i]++;
    }
    i128 ans = 1;
    for (int i = 0; i < n.size(); ++i) {
        ans *= a[n[i] - '0'];
    }
    print(ans);
    return 0;
}
```