[H007] - Alpha Problem

```cpp
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define fi first
#define se second
const int N = 1e6 + 9;
const int N2 = N * 10;
const int mod = 1e9 + 7;
const int inf = LLONG_MAX;



signed main(){
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    if (fopen("TASK.INP", "r")){
    freopen("TASK.INP", "r", stdin);
    freopen("TASK.OUT", "w", stdout);}

    int n, k;
    cin >> n >> k;

    stack<int> st;

    while (n != 0){
        st.push(n % k);
        n /= k;
    }

    while (!st.empty()){
        cout << st.top();
        st.pop();
    }

}
```

[H006] - Word Merging

```cpp
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define fi first
#define se second
const int N = 1e6 + 9;
const int N2 = N * 10;
const int mod = 1e9 + 7;
const int inf = LLONG_MAX;
```

```
signed main(){
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    if (fopen("TASK.INP", "r")){
    freopen("TASK.INP", "r", stdin);
    freopen("TASK.OUT", "w", stdout);}

    string s;
    cin >> s;

    stack<char> st;

    for (int i=0; i<s.size(); i++){
        if (!st.empty()){
            if (st.top() == s[i]){
                st.pop();
            }
            else{
                st.push(s[i]);
            }
        }
        else{
            st.push(s[i]);
        }
    }

    cout << st.size();

}
```

[H004] - Wood Cutting

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define fi first
#define se second
const int N = 1e6 + 9;
const int N2 = N * 10;
const int mod = 1e9 + 7;
const int inf = LLONG_MAX;



signed main(){
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    if (fopen("TASK.INP", "r")){
    freopen("TASK.INP", "r", stdin);
    freopen("TASK.OUT", "w", stdout);}

    int n, S;
    cin >> n >> S;

    priority_queue<int, vector<int>, greater<int>> pq;
```

```
    for (int i = 0; i < n; ++i) {
        int a;
        cin >> a;
        pq.push(a);
    }

    int ans = 0;

    while (pq.size() > 1) {
        int a = pq.top(); pq.pop();
        int b = pq.top(); pq.pop();
        int cost = a + b;
        ans += cost;
        pq.push(cost);
    }

    cout << ans << '\n';
}
```

[H003] - Isosceles Triangle

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define fi first
#define se second
const int N = 1e6 + 9;
const int N2 = N * 10;
const int mod = 1e9 + 7;
const int inf = LLONG_MAX;



signed main(){
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    if (fopen("TASK.INP", "r")){
    freopen("TASK.INP", "r", stdin);
    freopen("TASK.OUT", "w", stdout);}

    int n;
    cin >> n;

    map<int, int> m;
    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;
        m[x]++;
    }

    vector<pair<int, int>> v;

    for (auto it : m) {
        if (it.fi > 0) {
            v.push_back({it.fi, it.se});
        }
```

```
    }

    int M = v.size();
    vector<int> p(M + 1, 0);
    for (int i = 0; i < M; i++) {
        p[i + 1] = p[i] + v[i].se;
    }

    int ans = 0;
    for (int i = 0; i < M; i++) {
        int x = v[i].fi;
        int fx = v[i].se;
        if (fx >= 3) {
            ans += fx * (fx - 1) * (fx - 2) / 6;
        }

        if (fx >= 2) {
            int a = fx * (fx - 1) / 2;
            auto b = lower_bound(v.begin(), v.end(), make_pair(2 * x, -inf));

            int c = distance(v.begin(), b);
            int d = p[c], e = d - fx;
            if (e < 0) {
                e = 0;
            }
            ans += a * e;
        }
    }

    cout << ans;
}
```

[H002] - Difference in Height

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define fi first
#define se second
const int N = 1e6 + 9;
const int N2 = N * 10;
const int mod = 1e9 + 7;
const int inf = LLONG_MAX;



signed main(){
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    if (fopen("TASK.INP", "r")){
    freopen("TASK.INP", "r", stdin);
    freopen("TASK.OUT", "w", stdout);}

    int n;
    cin >> n;
    vector<int> v;
```

```
    vector<int> p(n + 1);
    for (int i=0; i<n; i++){
        int x;
        cin >> x;
        v.push_back(x);
    }
    sort(v.begin(), v.end());
    vector<int> v2;
    v2.push_back(0);
    for (int i=0; i<n; i++){
        v2.push_back(v[i]);
    }
    for (int i=1; i<=n; i++){
        p[i] = p[i-1] + v2[i];
    }
    int ans = 0;
    for (int i=1; i<=n; i++){
        int tmp1 = p[i-1] - p[0];
        int tmp2 = p[n] - p[i];
        ans += (i - 1) * v2[i] - tmp1;
        ans += tmp2 - (n - i) * v2[i];
    }
    cout << ans / 2;

}
```

[H001] - Tom's Currency

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define fi first
#define se second
const int N = 1e6 + 9;
const int N2 = N * 10;
const int mod = 1e9 + 7;
const int inf = LLONG_MAX;



signed main(){
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    if (fopen("TASK.INP", "r")){
    freopen("TASK.INP", "r", stdin);
    freopen("TASK.OUT", "w", stdout);}

    int n;
    cin >> n;

    map<int, int> m;

    for (int i=0; i<n; i++){
        int x;
        cin >> x;
        m[x]++;
```

```
    }

    cout << m.size() << "\n";

    for (auto it : m){
        cout << it.fi << " ";
    }

}
```

Hashing: VQ44_FLOWERS

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

int n, k;
vector<int> a;


vector<int> get_ans(const vector<int>& A,int K){
        map<int,int> Hash;

    for (int i=0; i<A.size(); i++){
        Hash[A[i]]++;
    }
    vector<int> ans;
    for (auto it : Hash){
        Hash[it.first]--;
        ans.push_back(it.first);
        K--;
        if (K == 0){
            return ans;
        }
    }
    for (auto it : Hash){
        for (int i=0; i<it.second; i++){
            ans.push_back(it.first);
            K--;
            if (K == 0){
                return ans;
            }
        }
    }
        return ans;
}

int main(){
        ios::sync_with_stdio(false); cin.tie(nullptr);
    if (fopen("TASK.INP", "r")){
        freopen("TASK.INP", "r", stdin);
        freopen("TASK.OUT", "w", stdout);
    }
        cin >> n >> k;

        a.resize(n);
        for (int i = 0; i < n; i++) cin >> a[i];
```

```
        vector<int> ans = get_ans(a,k);

        for (const int& x: ans) cout << x << ' ';
        return 0;
}
```

Hashing: KiemKe

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


int N;

int count_distinct(const vector<string>& ids){

        map<string,int> Hash;

        for (auto it : ids){
        Hash[it]++;
    }
    return Hash.size();

}

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

        cin >> N;
        vector<string> ids(N);

        for (int i = 0; i < N; i++) cin >> ids[i];

        cout << count_distinct(ids);

        return 0;
}
```