

Task

```
#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, p, q;
    cin >> n >> k >> p >> q;

    int alice=q+(p-1)*2;

    if (alice-k > 0){
        if ((alice-k)%2 == 0)
            cout << (alice-k+1)/2 << " " << 2;
        else
            cout << (alice-k+1)/2 << " " << 1;
    }
    else if (alice+k <= n){
        if ((alice+k)%2 == 0)
            cout << (alice+k+1)/2 << " " << 2;
        else
            cout << (alice+k+1)/2 << " " << 1;
    }
    else cout << -1;
}
```

Point3D

```
#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;

struct A {
    int x, y, z;
};

bool cmp(A a, A b) {
    if (a.x == b.x) {
        if (a.y == b.y) {
            return a.z < b.z;
        }
        return a.y > b.y;
    }
    return a.x < b.x;
}

void merge(vector<A> &v, int l, int m, int r) {
    vector<A> left(v.begin() + l, v.begin() + m + 1);
    vector<A> right(v.begin() + m + 1, v.begin() + r + 1);
    int i = 0, j = 0, k = l;
    while (i < left.size() && j < right.size()) {
        if (cmp(left[i], right[j])) {
            v[k++] = left[i++];
        } else {
            v[k++] = right[j++];
        }
    }
    while (i < left.size()) v[k++] = left[i++];
    while (j < right.size()) v[k++] = right[j++];
}

void mergeSort(vector<A> &v, int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        mergeSort(v, l, m);
        mergeSort(v, m + 1, r);
        merge(v, l, m, r);
    }
}

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);
    }

    vector<A> v;
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) {
        int x, y, z;
        cin >> x >> y >> z;
        v.push_back({x, y, z});
    }

    mergeSort(v, 0, n - 1);

```

```

    for (auto it : v) {
        cout << it.x << " " << it.y << " " << it.z << "\n";
    }
}

```

Tron2Mang

```

#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 T;
    cin >> T;

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

        vector<int> a(n), b(m);
        for (int i = 0; i < n; i++) cin >> a[i];
        for (int i = 0; i < m; i++) cin >> b[i];

        vector<int> merged;
        merged.reserve(n + m);

        int i = 0, j = 0;
        while (i < n && j < m) {
            if (a[i] < b[j]) merged.push_back(a[i++]);
            else merged.push_back(b[j++]);
        }
        while (i < n) merged.push_back(a[i++]);
        while (j < m) merged.push_back(b[j++]);

        for (int num : merged) cout << num << " ";
        cout << "\n";
    }

    return 0;
}

```

Find MEX

```
#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;

void merge(vector<int>& arr, int l, int m, int r) {
    int n1 = m - l + 1;
    int n2 = r - m;
    vector<int> L(n1), R(n2);

    for (int i = 0; i < n1; i++) L[i] = arr[l + i];
    for (int i = 0; i < n2; i++) R[i] = arr[m + 1 + i];

    int i = 0, j = 0, k = l;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) arr[k++] = L[i++];
        else arr[k++] = R[j++];
    }

    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];
}

void mergeSort(vector<int>& arr, int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);
        merge(arr, l, m, r);
    }
}

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

    mergeSort(a, 0, n - 1);
```

```

int mex = 0;
for (int i = 0; i < n; i++) {
    if (a[i] == mex) {
        mex++;
    } else if (a[i] > mex) {
        break;
    }
}

cout << mex << "\n";
return 0;
}

```

Point2D (template)

```

#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;

struct A {
    int x, y;
};

bool cmp(A a, A b) {
    if (a.x == b.x) {
        return a.y > b.y;
    }
    return a.x < b.x;
}

void merge(vector<A> &v, int l, int m, int r) {
    vector<A> left(v.begin() + l, v.begin() + m + 1);
    vector<A> right(v.begin() + m + 1, v.begin() + r + 1);
    int i = 0, j = 0, k = 1;
    while (i < left.size() && j < right.size()) {
        if (cmp(left[i], right[j])) {
            v[k++] = left[i++];
        } else {
            v[k++] = right[j++];
        }
    }
    while (i < left.size()) v[k++] = left[i++];
    while (j < right.size()) v[k++] = right[j++];
}

void mergeSort(vector<A> &v, int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        mergeSort(v, l, m);
        mergeSort(v, m + 1, r);
    }
}

```

```

        merge(v, l, m, r);
    }
}

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);
    }

    vector<A> v;
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) {
        int x, y;
        cin >> x >> y;
        v.push_back({x, y});
    }

    mergeSort(v, 0, n - 1);

    for (auto it : v) {
        cout << it.x << " " << it.y << "\n";
    }
}

```

VU33_MaxStr

```

#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;

    sort(s.rbegin(), s.rend());

    int sum = 0;
    for (auto c : s) sum += (c - '0');
}

```

```

    if (sum % 3 == 0) {
        cout << s << "\n";
        return 0;
    }

    int remainder = sum % 3;
    vector<int> count(10, 0);
    for (auto c : s) count[c - '0']++;

    bool removed = false;
    for (int i = remainder; i < 10; i += 3) {
        if (count[i] > 0) {
            count[i]--;
            removed = true;
            break;
        }
    }

    if (!removed) {
        int removeCount = 2;
        for (int i = 3 - remainder; i < 10; i += 3) {
            while (count[i] > 0 && removeCount > 0) {
                count[i]--;
                removeCount--;
            }
            if (removeCount == 0) break;
        }
    }

    string result = "";
    for (int i = 9; i >= 0; i--) {
        result += string(count[i], '0' + i);
    }

    if (result.empty() || result[0] == '0') result = "0";
    cout << result << "\n";

    return 0;
}

```

VQ44_FLOWERS(template)

```

#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;

void mer(vector<int>& arr, int l, int m, int r) {
    vector<int> left(arr.begin() + l, arr.begin() + m + 1);
    vector<int> right(arr.begin() + m + 1, arr.begin() + r + 1);
}

```

```

    int i = 0, j = 0, k = 1;
    while (i < left.size() && j < right.size()) {
        if (left[i] < right[j]) arr[k++] = left[i++];
        else arr[k++] = right[j++];
    }
    while (i < left.size()) arr[k++] = left[i++];
    while (j < right.size()) arr[k++] = right[j++];
}

void merSort(vector<int>& arr, int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        merSort(arr, l, m);
        merSort(arr, m + 1, r);
        mer(arr, l, m, r);
    }
}

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;
    vector<int> arr(n);
    vector<int> check(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    merSort(arr, 0, n - 1);

    arr.push_back(-123456789);
    for (int i = 0; i < n; i++) {
        // cout << arr[i] << " ";
        if (arr[i] != arr[i + 1]) {
            check[i] = 1;
            cout << arr[i] << " ";
            k--;
            if (k == 0) break;
        }
    }
    // cout << "\n";
    for (int i = 0; i < n; i++) {
        if (check[i] != 1 && k > 0) {
            cout << arr[i] << " ";
            k--;
            if (k == 0) break;
        }
    }
    return 0;
}

```



```
#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;

void mer(vector<string> &arr, int l, int m, int r) {
    vector<string> left(arr.begin() + l, arr.begin() + m + 1);
    vector<string> right(arr.begin() + m + 1, arr.begin() + r + 1);
    int i = 0, j = 0, k = l;
    while (i < left.size() && j < right.size()) {
        if (left[i] <= right[j]) {
            arr[k++] = left[i++];
        } else {
            arr[k++] = right[j++];
        }
    }
    while (i < left.size()) arr[k++] = left[i++];
    while (j < right.size()) arr[k++] = right[j++];
}

void merSort(vector<string> &arr, int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        merSort(arr, l, m);
        merSort(arr, m + 1, r);
        mer(arr, l, m, r);
    }
}

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<string> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    merSort(arr, 0, n - 1);

    int unique_count = 1;
    for (int i = 1; i < n; i++) {
        if (arr[i] != arr[i - 1]) unique_count++;
    }
}
```

```

        cout << unique_count << "\n";

        return 0;
    }

```

MergeSort

```

#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;

void merge(vector<int> &arr, int l, int m, int r) {
    vector<int> left(arr.begin() + l, arr.begin() + m + 1);
    vector<int> right(arr.begin() + m + 1, arr.begin() + r + 1);
    int i = 0, j = 0, k = l;
    while (i < left.size() && j < right.size()) {
        if (left[i] < right[j]) arr[k++] = left[i++];
        else arr[k++] = right[j++];
    }
    while (i < left.size()) arr[k++] = left[i++];
    while (j < right.size()) arr[k++] = right[j++];

    for (int i = 0; i < arr.size(); i++) {
        if (i == l) cout << "[ ";
        cout << arr[i] << " ";
        if (i == r) cout << "] ";
    }
    cout << "\n";
}

void mergeSort(vector<int> &arr, int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);
        merge(arr, l, m, r);
    }
}

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

mergeSort(arr, 0, n - 1);
return 0;
}

```

InsertionSort

```

#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;

void print(vector<int> arr) {
    for (int x : arr) {
        cout << x << " ";
    }
    cout << "\n";
}

void insertionSort(vector<int> &arr, int n) {
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
            print(arr);
        }
        arr[j + 1] = key;
        print(arr);
    }
}

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

```

```

        insertionSort(arr, n);

    return 0;
}

```

BubbleSort

```

#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;

void prinbt(vector<int> &arr) {
    for (int x : arr) {
        cout << x << " ";
    }
    cout << "\n";
}

void bubbleSort(vector<int> &arr, int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                swap(arr[j], arr[j + 1]);
                prinbt(arr);
            }
        }
    }
}

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

    bubbleSort(arr, n);

    return 0;
}

```

SelectionSort

```
#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> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

    for (int i = 0; i < n - 1; i++) {
        int mini = i;
        for (int j = i + 1; j < n; j++) {
            if (a[j] <= a[mini]) {
                mini = j;
            }
        }
        if (mini != i) {
            swap(a[i], a[mini]);
            for (int k = 0; k < n; k++) {
                cout << a[k] << " ";
            }
            cout << "\n";
        }
    }

    return 0;
}
```