```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <string>
#include <vector>
#include <algorithm>
#include <set>
#include <map>
#include <cmath>
#include <ctime>
#include <functional>
#include <sstream>
#include <fstream>
#include <valarray>
#include <complex>
#include <queue>
#include <cassert>
#include <bitset>
using namespace std;

#ifdef LOCAL
#define debug_flag true
#else
#define debug_flag false
#endif

#define dbg(args...) { if (debug_flag) { _print(_split(#args, ',').begin(), args); ↩
  cerr << endl; } else { void(0);} }

vector<string> _split(const string& s, char c) {
    vector<string> v;
    stringstream ss(s);
    string x;
    while (getline(ss, x, c))
        v.emplace_back(x);
    return v;
}

void _print(vector<string>::iterator) {}
template<typename T, typename... Args>
void _print(vector<string>::iterator it, T a, Args... args) {
    string name = it->substr((*it)[0] == ' ', it->length());
    if (isalpha(name[0]))
        cerr << name << " = " << a << " ";
    else
        cerr << name << " ";
    _print(++it, args...);
}

#ifdef LOCAL
#define eprintf(...) fprintf(stderr, __VA_ARGS__)
#else
#define eprintf(...) 42;
#endif
```

```
typedef long long int int64;

const int N = (int)2e5;
const int K = 15;

int n;
int a[N];
bool good_dp[N];
int good_cnt;

bool good(int i)
{
    if (i == n - 1)
        return true;
    if (i % 2 == 0)
        return a[i] < a[i + 1];
    return a[i] > a[i + 1];
}

void update(int i)
{
    if (good(i))
    {
        if (!good_dp[i])
        {
            good_dp[i] = true;
            good_cnt++;
        }
    }
    else
    {
        if (good_dp[i])
        {
            good_dp[i] = false;
            good_cnt--;
        }
    }
}

void do_swap(int i1, int i2)
{
    swap(a[i1], a[i2]);
    for (int dx = -1; dx <= 1; dx++)
    {
        if (0 <= i1 + dx && i1 + dx < n)
            update(i1 + dx);
        if (0 <= i2 + dx && i2 + dx < n)
            update(i2 + dx);
    }
}

int main()
{
#ifdef LOCAL
    freopen("input.txt", "r", stdin);
#endif
```

```
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);

    vector<int> cand_list;
    cand_list.reserve(3 * n);
    for (int i = 0; i < n; i++)
    {
        update(i);
        if (!good(i))
        {
            cand_list.push_back(i);
            if (i - 1 >= 0)
                cand_list.push_back(i - 1);
            if (i + 1 < n)
                cand_list.push_back(i + 1);
        }
    }

    sort(cand_list.begin(), cand_list.end());
    cand_list.erase(unique(cand_list.begin(), cand_list.end()), cand_list.end());

    if ((int)cand_list.size() > K)
    {
        printf("0\n");
        return 0;
    }

    vector<pair<int, int> > ans_list;
    ans_list.reserve(3 * n);
    for (int i1 : cand_list)
    {
        for (int i2 = 0; i2 < n; i2++)
        {
            do_swap(i1, i2);
            if (good_cnt == n)
                ans_list.emplace_back(i1, i2);
            do_swap(i1, i2);
        }
    }

    for (auto &p : ans_list)
        if (p.first > p.second)
            swap(p.first, p.second);

    sort(ans_list.begin(), ans_list.end());
    ans_list.erase(unique(ans_list.begin(), ans_list.end()), ans_list.end());

    int ans = (int)ans_list.size();

    printf("%d\n", ans);

    return 0;
}
```