

**算法实现：**

**2.6**

代码：

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int fac(int n) {

int result = 1;

for (int i = 1; i <= n; ++i) result \*= i;

return result;

}

int find(const vector<int>& arr) {

int order = 0;

int n = arr.size();

for (int i = 0; i < n; ++i) {

int cnt = 0;

for (int j = i + 1; j < n; ++j)

if (arr[j] < arr[i]) cnt++;

order += cnt \* fac(n - i - 1);

}

return order;

}

void next(vector<int>& arr) {

int n = arr.size();

int i = n - 2;

while (i >= 0 && arr[i] >= arr[i + 1]) i--;

for (int j = n - 1; j > i; --j)

if (arr[j] > arr[i]) {

swap(arr[i], arr[j]);

break;

}

reverse(arr.begin() + i + 1, arr.end());

return;

}

int main() {

int n;

cin >> n;

vector<int> num(n);

for (int i = 0; i < n; ++i) cin >> num[i];

int cnt = find(num);

next(num);

cout << cnt << endl;

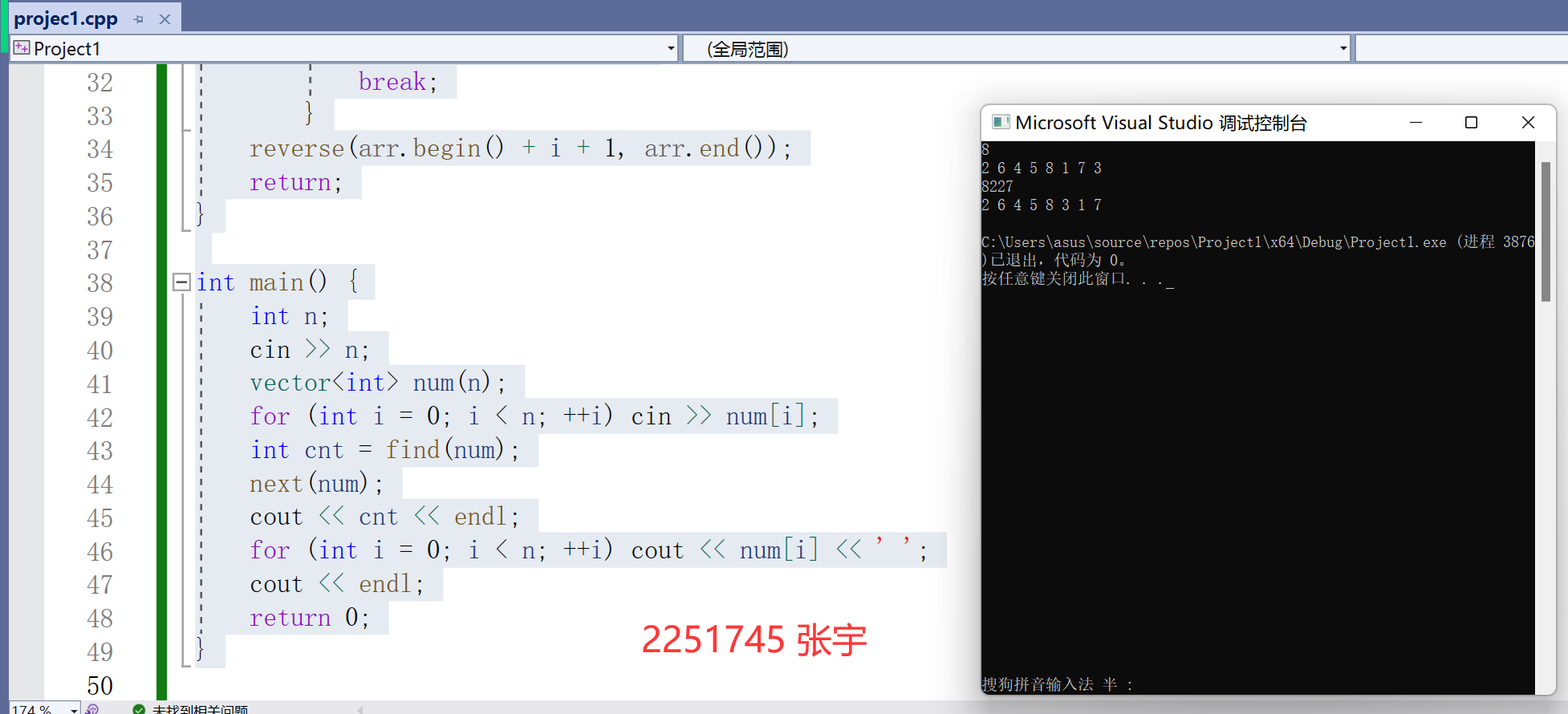
for (int i = 0; i < n; ++i) cout << num[i] << ' ';

cout << endl;

return 0;

}

测试截图：



**2.9**

代码：

#include <iostream>

using namespace std;

void hanoi(int n, char start, char mid, char end) {

if (n == 0) return;

hanoi(n - 1, start, end, mid);

cout << n << " " << start << " " << mid << endl;

hanoi(n - 1, end, mid, start);

}

int main() {

int n;

cin >> n;

hanoi(n, 'A', 'B', 'C');

return 0;

}

测试截图：

