

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Bảy, 5 tháng 4 2025, 3:36 PM
Kết thúc lúc	Thứ Bảy, 5 tháng 4 2025, 5:00 PM
Thời gian thực hiện	1 giờ 24 phút
Điểm	90,00/90,00
Điểm	10,00 trên 10,00 (100%)

Câu hỏi 1

Đúng

Đạt điểm 10,00 trên 10,00

Viết chương trình nhập vào một dãy số gồm N phần tử, mỗi phần tử trong khoảng từ 0 đến 9. Kiểm tra xem số nào là số xuất hiện nhiều nhất.

Đầu vào:

- Dòng đầu tiên chứa N ($N \leq 100$) là số lượng phần tử của dãy.
- Dòng thứ 2 chứa N số nguyên a_i cách nhau bởi dấu cách ($0 \leq a_i \leq 9$).

Đầu ra: In ra màn hình số xuất hiện nhiều nhất. Nếu có nhiều số thoả mãn, in ra số nhỏ nhất

For example:

Input	Result
10 1 2 0 1 2 3 2 9 6 2	2

Answer: (penalty regime: 0 %)

```
1 #include <algorithm>
2 #include <iostream>
3 #include <map>
4 #include <vector>
5 using namespace std;
6
7 int main() {
8     int n;
9     cin >> n;
10    vector<int> v(10, 0);
11    for (int i = 0; i < n; ++i) {
12        int x;
13        cin >> x;
14        ++v[x];
15    }
16    cout << max_element(begin(v), end(v)) - begin(v);
17 }
```

	Input	Expected	Got	
✓	10 1 2 0 1 2 3 2 9 6 2	2	2	✓

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 2

Đúng

Đạt điểm 10,00 trên 10,00

An có 2 dãy số A và B , B là hoán vị của A . Do sơ suất, An đã xóa đi một số phần tử trong dãy A . Bạn hãy giúp An xác định nhưng con số bị mất này nhé.

Đầu vào: Dòng đầu tiên chứa số nguyên n là số phần tử của dãy A sau khi đã bị xóa. Dòng thứ 2 chứa n phần tử của dãy A , dòng thứ 3 chứa số nguyên m là số phần tử của dãy B . Dòng thứ 4 chứa m phần tử của dãy B . ($n, m \leq 2 \times 10^5$, các phần tử $\leq 10^4$).

Đầu ra: In ra các phần tử bị xóa theo thứ tự tăng dần, nếu có nhiều phần tử bị xóa có giá trị bằng nhau, chỉ in ra giá trị đấy 1 lần.

Dữ liệu vào nhập từ bàn phím và kết quả được in ra màn hình.

For example:

Input	Result
10 203 204 205 206 207 208 203 204 205 206 13 203 204 204 205 206 207 205 208 203 206 205 206 204	204 205 206

Answer: (penalty regime: 0 %)

```
1 #include <algorithm>
2 #include <iostream>
3 #include <map>
4 #include <vector>
5 using namespace std;
6
7 int main() {
8     map<int, int> m;
9     int n, x;
10    for (cin >> n; n--;) {
11        cin >> x;
12        --m[x];
13    }
14    for (cin >> n; n--;) {
15        cin >> x;
16        ++m[x];
17    }
18    for (auto [i, c] : m) {
19        if (c) {
20            cout << i << ' ';
21        }
22    }
23 }
```

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 3

Đúng

Đạt điểm 10,00 trên 10,00

Ave là một người rất thích các dãy số. Một hôm rảnh rỗi, Ave viết ra giấy một dãy gồm N số nguyên và 1 số nguyên A . Ave muốn tìm một đoạn con liên tiếp $[L, R]$, sao cho trung bình cộng của đoạn này lớn hơn hoặc bằng A và có độ dài càng lớn càng tốt. Nếu có nhiều đoạn như vậy, Ave muốn chọn đoạn có đầu mút trái nhỏ nhất. Các bạn hãy giúp Ave giải bài toán này nhé.

Đầu vào

Dòng đầu 2 số N, A ($N \leq 100000$). N dòng sau, dòng thứ i là số a_i biểu diễn số nguyên thứ i trong dãy số ($|a_i| \leq 10000$).

Đầu ra

Dòng đầu là độ dài của đoạn tìm được. Dòng thứ 2 là 2 số gồm chỉ số đầu và chỉ số cuối của đoạn tìm được.

For example:

Input	Result
7 3	5
1	2 6
2	
3	
5	
3	
2	
1	

Answer: (penalty regime: 0 %)

```

1 #include <algorithm>
2 #include <cstdio>
3 #include <iostream>
4 #include <iterator>
5 #include <map>
6 #include <set>
7 #include <stack>
8 #include <string>
9 #include <vector>
10 using namespace std;
11
12 int main() {
13     int n, k;
14     cin >> n >> k;
15     vector<int64_t> pref(n + 1, 0);
16     for (int i = 0; i < n; ++i) {
17         int x;
18         cin >> x;
19         pref[i+1] = pref[i] + x;
20     }
21     int aL = -1;
22     int aR = -1;
23     for (int i = 1; i <= n; ++i) {
24         for (int j = n; j >= i; --j) {
25             if (j - i <= aR - aL) {
26                 break;
27             }
28             int64_t s = pref[j] - pref[i - 1];
29             if (s / (j - i + 1) >= k) {
30                 aL = i;
31                 aR = j;
32                 break;
33             }
34         }
35     }
36     cout << aR - aL + 1 << '\n' << aL << ' ' << aR;
37 }
```

Passed all tests! ✓

▸ **SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)**

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 4

Đúng

Đạt điểm 10,00 trên 10,00

Viết chương trình BeautifulStrings với nhiệm vụ đọc **một chuỗi kí tự** là chữ cái tiếng Anh (chữ thường) từ input chuẩn (bàn phím) kết thúc khi hết input, nhiệm vụ của bạn là in ra **số dãy con khác nhau** sinh ra từ việc cắt đi hai kí tự bất kì trong chuỗi kí tự đầu vào. Kích thước của input được giới hạn 250 (độ dài chuỗi kí tự đầu không quá 250).

Đầu vào:

Một chuỗi ký tự chữ thường là chữ cái tiếng Anh.

Đầu ra:

Số dãy con khác nhau sinh ra từ việc cắt đi hai ký tự bất kỳ trong chuỗi kí tự đầu vào.

For example:

Input	Result
abba	4

Answer: (penalty regime: 0 %)

```
1 #include <algorithm>
2 #include <iostream>
3 #include <map>
4 #include <set>
5 #include <string>
6 #include <vector>
7 using namespace std;
8
9 int main() {
10     string s;
11     cin >> s;
12     int n = s.length();
13     set<string> st;
14     for (int i = 0; i < n - 1; ++i) {
15         for (int j = i + 1; j < n; ++j) {
16             string t = s.substr(0, i) + s.substr(i+1, j-i-1) + s.substr(j+1);
17             st.emplace(t);
18         }
19     }
20     cout << st.size();
21 }
```

	Input
✓	abba
✓	abbaababbaedbac
✓	abbaababbaedbacabbaababbaedbacabbaababbaedbacabbaababbaedbahjkhjhcbbaababbaedbac
✓	abbaababbaedbacabbaababbaedbacabbaababbaedbacabbaababbaedbahjkhjhcbbaababbaedbacabbaababbaedbacabbaababbaedbacabbaabat
✓	abbaababbaedbacabbaababbaedbacabbaababbaedbacabbaababbaedbahjkhjhcbbaababbaedbacabbaababbaedbacabbaababbaedbacabbaabat

Passed all tests! ✓

► **SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)**

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 5

Đúng

Đạt điểm 10,00 trên 10,00

Cho một chuỗi ký tự chỉ bao gồm các ký tự '(', ')', '{', '</code>, <code>}', '[' and ']', xác định xem chuỗi đó có hợp lệ hay không. Các dấu ngoặc cần phải được đóng theo thứ tự, "()" và "{}[]" đều là chuỗi hợp lệ nhưng "]" and "([])" thì không. Viết hàm `bool isValid(string s)` trả về `true` nếu chuỗi ký tự chứa các dấu ngoặc là hợp lệ, ngược lại `false`.

For example:

Test	Input	Result
<pre>string s; cin >> s; cout << isValid(s);</pre>	{[()]}	1

Answer: (penalty regime: 0 %)

```
1 #include <stack>  
2  
3 bool isValid(string s) {  
4     stack<char> st;  
5     for (char c : s) {  
6         if (c == '{' || c == '(' || c == '[') {  
7             st.emplace(c);  
8             continue;  
9         }  
10        if (c == '}') {  
11            if (st.top() != '{') {  
12                return false;  
13            }  
14        } else if (c == ')') {  
15            if (st.top() != '(') {  
16                return false;  
17            }  
18        } else if (c == ']') {  
19            if (st.top() != '[') {  
20                return false;  
21            }  
22        }  
23        st.pop();  
24    }  
25    return st.empty();  
26 }
```

	Test	Input	Expected	Got	
✓	<pre>string s; cin >> s; cout << isValid(s);</pre>	{[()]}	1	1	✓
✓	<pre>string s; cin >> s; cout << isValid(s);</pre>	{[()]}	0	0	✓

	Test	Input	Expected	Got	
✓	<pre>string s; cin >> s; cout << isValid(s);</pre>	{[()]}	0	0	✓
✓	<pre>string s; cin >> s; cout << isValid(s);</pre>	[{()}]	0	0	✓
✓	<pre>string s; cin >> s; cout << isValid(s);</pre>	()[]{}	1	1	✓

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 6

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu: Cho đẳng thức $a + b = c$, trong 3 số này có 1 số bị mờ đi một chữ số (được thay bằng dấu ?). Hãy viết chương trình tìm chữ số đó.

Đầu vào:

- Dòng đầu chứa một chuỗi mô tả số nguyên không âm a ;
- Dòng thứ hai chứa một chuỗi mô tả số nguyên không âm b ;
- Dòng thứ ba chứa một chuỗi mô tả số nguyên không âm c ;
- Các chuỗi có không quá 100 ký tự số.

Đầu ra:

- Gồm một dòng chứa chữ số là kết quả tương ứng của bộ test trong dữ liệu vào.

For example:

Input	Result
1 ? 1	0

Answer: (penalty regime: 0 %)

Reset answer

```

1 #include <algorithm>
2 #include <iostream>
3 #include <iterator>
4 using namespace std;
5
6 string add(const string &a, const string &b) {
7     string x(rbegin(a), rend(a));
8     string y(rbegin(b), rend(b));
9     int n = x.length();
10    int m = y.length();
11    if (n < m) {
12        swap(x, y);
13        swap(n, m);
14    }
15    int carry = 0;
16    string res;
17    for (int i = 0; i < n; ++i) {
18        int cur = carry + x[i] - '0';
19        if (i < m) {
20            cur += y[i] - '0';
21        }
22        res.push_back(cur % 10 + '0');
23        carry = cur / 10;
24    }
25    if (carry) {
26        res.push_back('1');
27    }
28    reverse(begin(res), end(res));
29    return res;
30 }
31
32 int main() {
33     string a, b, c;
34     cin >> a >> b >> c;
35
36     size_t i = a.find('?');
37     if (i != string::npos) {
38         for (char x : "0123456789") {
39             a[i] = x;
40             if (add(a, b) == c) {
41                 cout << x;
42                 return 0;

```

```
43     }
44 }
45 }
46 i = b.find('?');
47 if (i != string::npos) {
48     for (char x : "0123456789") {
49         b[i] = x;
50         if (add(a, b) == c) {
51             cout << x;
52             return 0;
53         }
54     }
55 }
```

	Input	Expected	Got	
✓	1 ? 1	0	0	✓

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 7

Đúng

Đạt điểm 10,00 trên 10,00

Viết chương trình nhập vào một dãy số gồm N phần tử, mỗi phần tử trong khoảng từ 0 đến 9. Kiểm tra xem giá trị số nào không xuất hiện.

Đầu vào:

- Dòng đầu tiên chứa N ($N \leq 100$) là số lượng phần tử của dãy.
- Dòng thứ 2 chứa N số nguyên a_i cách nhau bởi dấu cách ($0 \leq a_i \leq 9$)

Đầu ra: In ra màn hình danh sách các số không xuất hiện trong dãy sắp xếp từ nhỏ đến lớn và cách nhau bởi dấu cách. Nếu tất cả các số đều xuất hiện thì in ra "None".

For example:

Input	Result
10 1 2 0 1 2 3 2 9 6 2	4 5 7 8

Answer: (penalty regime: 0 %)

```
1 #include <algorithm>
2 #include <iostream>
3 #include <iterator>
4 #include <map>
5 #include <set>
6 #include <stack>
7 #include <string>
8 #include <vector>
9 using namespace std;
10
11 int main() {
12     vector<int> v(10, 0);
13     int n, x;
14     for (cin >> n; n--;) {
15         cin >> x;
16         ++v[x];
17     }
18     bool ok = false;
19     for (int i = 0; i <= 9; ++i) {
20         if (v[i] == 0) {
21             ok = true;
22             cout << i << ' ';
23         }
24     }
25     if (!ok) {
26         cout << "None";
27     }
28 }
```

	Input	Expected	Got	
✓	10 1 2 0 1 2 3 2 9 6 2	4 5 7 8	4 5 7 8	✓

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 8

Đúng

Đạt điểm 10,00 trên 10,00

Tuyến phố thông minh Anna có n tòa nhà, tòa nhà thứ i có khoảng cách x_i mét so với đầu của tuyến phố. Thành phố quyết định sẽ đặt các trạm phát sóng wifi miễn phí tại các tòa nhà dọc theo tuyến phố thông minh. Mỗi thiết bị phát wifi có thể phủ sóng trong phạm vi k mét, tức là các tòa nhà có khoảng cách đến thiết bị không vượt quá k mét thì sẽ sử dụng được wifi từ thiết bị này. Bạn hãy viết chương trình tìm cách đặt các trạm wifi để số lượng trạm wifi là ít nhất và tất cả các tòa nhà đều có thể sử dụng wifi.

Đầu vào

Dữ liệu vào từ bàn phím gồm 2 dòng. Dòng đầu tiên chứa 2 số nguyên n, k ($n, k \leq 10^5$). Dòng thứ hai chứa n số nguyên x_i ($x_i \leq 10^5$). Hai số nguyên trên cùng một dòng cách nhau bởi một dấu cách.

Đầu ra

In ra màn hình số lượng thiết bị phát wifi ít nhất để tất cả các tòa nhà đều có thể sử dụng wifi.

For example:

Input	Result
5 1 1 2 3 4 5	2
6 1 1 2 3 4 5 6	2
2 3 1 6	2

Answer: (penalty regime: 0 %)

```

1 #include <algorithm>
2 #include <cstdlib>
3 #include <iostream>
4 #include <iterator>
5 #include <map>
6 #include <set>
7 #include <stack>
8 #include <string>
9 #include <vector>
10 using namespace std;
11
12 int main() {
13     int n, k;
14     cin >> n >> k;
15     vector<int> a(n);
16     for (int &x : a) {
17         cin >> x;
18     }
19     sort(begin(a), end(a));
20     int ans = 0;
21     for (int i = 0; i < n; ) {
22         ++ans;
23         auto it1 = upper_bound(begin(a), end(a), a[i] + k) - 1;
24         auto it2 = upper_bound(begin(a), end(a), *it1 + k);
25         if (it2 == end(a)) {
26             cout << ans;
27             return 0;
28         }
29         i = it2 - begin(a);
30     }
31 }
```

	Input	Expected	Got	
✓	1 1 1	1	1	✓

	Input	Expected	Got	
✓	80888 20 32236 2373 25551 74679 91506 81123 46756 48107 62887 62197 85535 97856 71700 33748 32178 84999 2047 87294 46891 8337 24848 47516 41957 57954 45840 14465 86575 41126 76679 45301 19595 27775 31594 9022 58366 34854 56547 61143 54571 37148 95229 23510 69272 19839 9711 40737 94050 96812 46322 63067 58059 10558 81951 13803 21330 59542 51193 24819 85941 83279 73664 6938 35056 23626 58922 63477 804 65666 4730 71354 11345 86685 69447 96896 62009 20090 91616 25898 70858 39439 42061 38284 86696 35344 4275 6220 1012 86278 25031 66345 12761 78733 27024 99978 30561 48946 45546 13309 61063 83306 35251 74259 96726 76820 54169 16105 87625 75465 80235 59255 4809 5060 47143 29215 80649 22031 90665 29115 63287 19980 93205 39787 66333 21702 20847 39612 55852 1659 75601 73595 66644 34706 8338 28755 23702 85893 68782 74871 52833 86312 3086 87292 73837 72968 7944 5757 18484 9565 90450 7334 31508 24795 3429 86387 1743 3829 93557 5658 37253 11762 62119 57836 41393 31916 21342 94083 87771 13533 94968 80460 98786 64487 4138 65318 5037 35179 85798 453 17617 36653 56167 60139 57986 73737 66196 55303 3759 98110 30863 58657 51417 8307 17440 85502 94405 65281 82884 51711 13943 77914 67089 69030 41020 94885 31150 46420 29097 71075 66268 45915 90358 27806 83630 31950 34898 59203 70776 18048 52740 50540 44038 69341 29674 53646 8123 56254 19244 21253 56642 65767 72867 59743 3677 53345 37918 69610 16581 70100 60214 69203 86467 14060 92972 58901 88524 69880 87238 60838 60095 19292 24298 2385 74697 60851 72452 22047 16666 82763 19484 93541 10934 83854 49870 17051 32157 81848 90014 17852 3666 29817 32409 82824 87604 14527 99902 18784 72303 84511 90069 98718 1613 63587 77176 61737 46164 54059 69045 24457 42997 3780 56932 45032 26001 84697 57411 88858 29083 50560 28026 93969 32712 56386 20077 75948 22276 44193 93154 8214 45447 83780 61416 94314 57178 72356 26542 40899 18 86037 10434 63102 83080 279 10936 93532 20677 46295 91284 17436 80783 18302 5328 34123 64347 4250 51570 30552 36492 9116 49760 20195 15106 1805 95480 68102 17150 57561 18610 9217 36885 77541 21260 68807 31286 15441 3986 4142 61429 13505 24113 11691 97100 61036 11421 5124 7734 31096 17422 35903 54246 8571 74750 34328 55907 56624 86039 32220 59358 93070 13858 28596 18568 65098 35763 53872 92005 37518 46004 96890 87417 85894 7467 20068 68673 97249 46723 1909 17310 1083 8159 9544 31530 35567 82062 36497 64062 5582 46211 74305 84550 32088 18104 94803 27902 99826 55409 69021 58964 11529 32094 51719 50432 78477 11997 91826 1105 42665 90511 23 44152 40685 23108 57267 32984 11101 51767 64612 94647 56272 72099 47781 94821 54236 11208 2984 17829 75290 36303 69878 24432 20079 95706 7665 17120 39265 16995 75750 48765 39536 69691 72265 37930 77645 4373 4592 76586 39540 94546 98832 81051 2630 84011 89496 84792 11692 75952 61053 80545 94977 7667 61597 5461 19014 7524 20747 59154 3016 24125 35581 2817 58478 7658 81034 6853 58273 62753 91208 14287 69144 84577 55293 99268 20605 44120 78514 53765 30297 25992 32089 65556 10603 91501 5088 18079 45529 95320 12273 22136 33860 26594 83781 91159 32949 59339 62054 3170 85097 80297 14917 64207 94659 74308 56022 16176 73566 87295 85820 78288 77907 95744 24117 40881 25109 51837 17842 70175 19524 10440 80457 21038 56041 28826 12869 94514 30902 136 7734 99399 82779 39560 61540 64831 16443 41074 83720 78406 89700 95060 70549 44779 49764 18021 77238 22209 38330 26581 44731 22157 42951 1324 97507 86194 33037 68095 49655 51577 88626 14694 8546 4072 28969 38464 75109 532 73839 5713 55695 7342 69465 81582 28361 72620 64814 30951 4749 30389 98562 34436 64031 56555 17935 86016 72135 47706 63933 14849 96430 15855 28895 5701 5636 74038 18331 39926 41878 91804 23753 31008 74329 79832 4753 49523 12863 84867 30864 56127 56028 19240 29183 80113 56172 565 31149 50434 76002 19583 74159 33804 79017 40911 95669 55997 82586 35457 82897 64479 38107 18747 31117 78992 22182 60187 44368 35178 37633 53188 22133 40397 180 87493 45562 54636 98059 78863 72998 79495 93961 73397 50033 75082 98664 ...snip... 87 90611 70666 39087 3346 43393 86398 76083 83347 80821 61880 33557 91723 53836 8931 19474 25577 6814 48027 38248 13667 94450 62657 26776 83332 4046 84924 64546 30109 37924 15001 74055 77819 25913 35453 70686 65385 86142 7911 49740 48917 95482 29878 80330 67804 44003 61129 33436 45353 99316 93272 93203 54532 73140 1397 35452 52456 9405 6128 13256 96106 75933 74179 65227 756 26184 48781 8366 94330 98741 79833 68993 27309 859 67669 15936 76777 62632 64884 13069 89626 36994 23414 66640 74656 355 60370 265 96604 36808 68395 98368 43332 19857 11611 44489 63667 67940 95826 38242 33210 59845 65736 4274 78595 51530 87528 29085 76549 92635 41475 52828 836 33111 58920 26334 55252 8937 42289 59194 63353 18211 4193 90294 90910 94168 30475 46415 12193 37131 19648 42217 59758 32262 77828 92098 15635 94631 68155 38965 15118 87447 71738 62747 68972 83726 86031 52105 66737 11697 6698 43336 84344 50711 65346 70686 21066 78388 93975 11491 87905 65157 42403 49926 59152 55038 66980 17862 84459 9946 35506 30843 67801 29125 71015 30704 36894 99639 42445 53832 42376 90306 40620 66685 53935 8956 4514 41329 99704 57198 30669 73950 54103	2440	2440	✓

Input	Expected	Got
64567 28619 83134 93611 68874 97486 2068 97981 7819 43525 67739 60529 57180 82170 21105 15212 27593 38197 36866 30677 53668 48770 99898 79759 95760 62805 46633 11630 72450 20358 17962 13257 34425 21736 91002 3081 13359 88272 85311 84423 5465 55512 29684 76964 26818 99993 59525 58187 22728 61387 20035 86396 61417 53016 93841 5004 363 34568 94306 71792 32478 70439 19958 99415 42883 74146 99734 7180 56942 95486 91475 3115 69287 64417 61921 25625 15362 24593 43062 56832 51851 2886 81482 28057 78835 23394 88481 40525 82967 4452 85561 32458 376 17860 17860 20665 10262 82996 11158 99400 10581 43870 46443 9619 42614 77618 40903 83440 21145 29330 43152 45196 12998 35439 78276 97272 80750 84564 84274 27270 99394 66876 3622 62234 56958 48232 24257 71592 40246 82829 83332 94609 23232 46916 88353 65191 75616 73838 33561 50925 30968 82553 28425 1816 5253 69724 4380 17785 19227 39931 73805 82434 75029 80490 65128 45645 59844 3119 7826 37693 9789 34154 61418 20074 72325 13086 13195 31678 24292 41115 50679 72745 18646 56771 11758 56299 87233 84762 22477 90148 5359 39933 90584 84329 5770 82044 73961 22571 88239 4982 38395 92187 34748 98505 19483 59767 37801 34240 44606 66129 12655 5222 21618 75179 37368 20704 36698 9599 13892 90062 97274 55830 88438 49020 88996 61709 14484 99485 48617 19254 68538 21084 4339 70670 2233 82609 19223 93671 36823 79961 13264 52411 17177 71424 81688 53370 79723 65596 77588 7737 24048 64576 21844 31826 88755 53855 86729 13493 36766 54895 57705 66506 71565 20026 93369 9810 5976 37708 57148 45453 91695 63919 26174 17743 7605 47001 12394 91646 9503 41581 7406 17243 16956 84160 67834 42337 2526 63030 31908 88128 48073 62236 18113 84347 12775 65077 5898 81454 99860 45665 14750 97998 68266 54262 66784 68223 60346 19210 76950 41220 36233 46604 58430 82634 16768 77382 37552 54252 97414 80516 62833 81109 17333 65974 92249 94260 42153 85668 35998 21742 41605 34507 47131 12644 34769 42399 18416 33982 14691 78416 53097 80431 23476 45703 88733 50985 90865 87538 33381 52996 17410 93172 77536 83564 10091 45256 88799 93511 38793 78416 23442 95727 85346 20890 64332 69074 2556 92089 96970 203 98023 9733 84398 12126 83277 40610 11101 87510 9778 47932 6091 8128 32092 21846 21314 34967 32108 88115 65736 20072 26796 70387 48572 84920 68100 51291 15339 61570 99140 53967 86819 76142 73598 52281 98881 94275 8477 86521 5229 74999 25685 33239 33615 60311 66525 12866 99752 39646 90819 54154 65064 34661 20597 9120 55578 31269 75251 92254 79177 60945 98974 19644 6536 79595 94239 90590 16535 55924 63533 43268 67141 36009 27176 16490 15685 5530 31140 21661 29617 48351 79198 5170 21081 97919 72887 69516 14935 34353 40265 42820 16930 98868 92751 35358 80500 15371 1756 43502 35803 19873 34880 58775 47778 38075 21785 26422 53382 29287 31935 96216 41388 37711 41540 93810 79860 55189 60767 56292 33581 8420 81613 97731 26610 52455 17704 85159		

	Input	Expected	Got	
✓	9740 70709 73513 30978 73173 526 76230 98275 72253 49426 30394 78574 51513 37721 99007 79850 77881 12175 53776 83773 98274 93953 7491 14084 46388 49751 21568 98875 88463 9820 17960 96188 98817 56644 40022 66587 56268 74791 32337 79275 72610 46882 73927 33022 20315 67293 6240 33673 39002 50042 88955 72538 66503 78707 12968 91848 49296 69118 3281 42535 2485 48210 33685 10807 50539 60209 9312 32447 28072 86697 12555 73434 38953 42926 73758 54444 12586 7495 60734 64308 18098 12560 89459 12704 65330 15508 40002 65639 53328 40087 72735 74032 34959 26712 85847 5263 27898 82361 12944 35866 70369 10805 48653 51503 20730 25912 94081 42323 94376 3346 2459 53604 48137 90576 750 38330 26513 65451 65991 43055 64214 48973 11560 53271 27744 45823 79279 43483 87358 83823 52556 69968 65492 73703 7182 40890 21383 79731 65430 37677 81619 11783 47669 25655 27063 95379 82170 21216 79143 91289 52839 6992 21118 95515 83972 31357 38304 81992 58992 77086 76012 77644 36383 10506 41719 44154 18737 20386 67868 5529 72947 28477 12926 65471 72873 85077 12925 22425 62937 63088 88230 47767 7945 20023 35151 85182 24653 72076 58234 73912 93276 40680 42423 17183 28957 26343 52300 42601 88639 24381 81490 86624 12476 64005 48612 72645 26970 306 17669 97786 49979 80175 82377 48831 60707 77753 18421 78275 57840 98299 3463 56091 29548 10291 24457 73736 75385 13436 82919 42962 9237 53213 27760 20951 6726 79228 82130 49879 92287 62815 104 3922 18901 59002 26925 26399 71491 68406 71599 47842 79542 25743 26397 80751 29961 23648 13461 87444 80031 70823 60140 88755 44379 81316 58079 18771 37121 66733 70167 72692 71657 32655 81543 39048 21735 61189 25781 3065 53621 89358 27056 56253 28377 74143 25259 80631 14770 9850 68626 73931 28164 82641 75427 28946 8303 45585 41472 7686 2196 43254 7353 97445 17832 7429 66925 33469 9226 10926 95748 9257 56947 83562 91 65545 3295 72634 36874 16012 44002 64381 67853 12585 67636 26086 59427 25887 12090 51132 30894 50867 33481 11113 36360 97921 50416 56395 98815 64272 40493 22725 98101 77668 38967 94014 1224 99776 42635 94855 21294 17891 18679 28853 40796 27353 51359 7025 15723 34445 15401 16783 7904 48972 99528 94338 92488 14834 4019 86532 3459 79670 10922 14966 57541 75815 24012 58889 61160 55894 23945 30997 58791 86424 49147 59840 92615 89239 33303 28429 7667 8640 59454 25598 12182 37566 10432 17351 45083 80486 20027 6797 38988 75327 3117 18760 22134 12263 12717 8403 38418 23680 60183 36006 81563 51446 71697 10878 76713 81721 75329 59430 28941 29366 36573 7127 59871 99019 89241 41495 88878 50653 28167 32669 37406 6294 6381 63681 11216 20284 70550 34136 38919 31376 16204 88591 65324 78517 4840 3263 70088 93530 71980 65420 11039 17643 3877 18754 65654 42695 41057 38345 50334 94110 81787 74775 17042 93716 93507 34225 87709 87897 15860 22170 83622 23442 76425 94013 8816 41724 9210 51339 92604 14162 74413 59641 51311 77216 59822 65152 47481 27044 31482 600 7975 60999 10566 87285 43736 63637 92754 57730 73386 8668 53694 19649 52712 72275 82765 51948 38019 58323 22858 36201 46246 21844 64152 51263 38741 30336 1058 87995 13463 11462 75620 7640 25811 51850 71704 98482 97534 25572 6508 97470 71732 4091 63018 94011 12866 21914 40760 81970 93440 31827 29523 34322 65010 27921 2363 51393 39738 39172 69745 2175 19517 85623 54850 91430 72062 31382 43945 36952 34877 71038 47583 25718 60893 64679 5843 2577 67491 68556 77840 65325 95195 88199 56657 65216 87114 41729 67490 76692 68415 70980 54231 39144 62943 2054 64104 79834 19657 25714 41966 97171 45084 20892 71178 92845 41629 32034 63012 23789 91365 25394 56299 2596 49516 58769 66229 10771 30363 93964 48830 85771 34249 82267 86871 25741 34695 68076 40331 79617 59943 55738 94526 35671 54366 75860 42099 96568 64768 47945 94466 37407 6709 81718 50107 13628 64164 88623 97298 7266 44400 13790 96976 37192 68046 6138 4439 99825 9706 47179 42945 46724 88050 96641 33245 26794 28877 4425 37032 51396 17525 31584 81177 7337 47848 20680 94283 32383 68302 84340 75964 87256 7999 97011 91741 18270 41492 71476 18969 ...snip... 905 44779 61638 95160 28371 77859 26422 52787 61443 66006 96476 11637 71334 93731 10205 17333 59024 10343 80510 95516 25557 73512 47335 94665 49695 95634 4182 16697 11677 27487 98605 23810 32222 34205 18850 30248 45366 98547 57300 78634 28154 28160 95181 42700 49223 54567 90147 70894 85005 59704 3009 14301 6491 6232 24327 93417 25995 39502 60977 2516 6853 74487 66485 44319 23305 33165 11096 67401 70611 83149 68155 23051 19018 53239 85654 76322 26702 89925 9534 86741 79654 27590 79840 25584 51678 33374 17259 89634 56204 56493 27424 56028 22099 13940 36489 78460 73535 72707 12343 78757 33819 31478 66073 95440 26786 29113 60472 79125 41229 65907 23904 75179 23110 95815 49692 22423 91502 43920 75443 25838 15551 92785 12370 66026 96058 33347 92381 17050 12017 45414 82751 58993 56571 85897 19828 19580 30414 72217 95483 75737 45196 22289 28483 57850 60966 79216 51444 4408 92904 87254 73670 11702 88381 39595 94163 6740 85455 34412 19297 48121 54175 88766 26174 87721 5247 64929 75615 50435 96127 73413 19380 90558 17229 46490 71041 38123 25576 33615 33737 32204 98507 94198 40150 20629 28647 29813 91707 6133 76911 51207 92308 15970	1	1	✓

	Input	Expected	Got	
	69516 83192 17179 6183 83374 25691 21637 4764 62515 68443 11216 36148 98292 50542 2043 21952 3099 43520 4200 93289 8671 67554 28703 84081 21829 72028 13158 48558 74660 53139 70554 60504 87112 81790 59024 94978 32470 11711 93653 88848 10569 42625 90272 77637 35895 87031 19811 95893 16535 4413 27531 25139 26332 55220 75247 11056 33963 79591 38445 86767 67959 57539 59156 24766 42764 85114 80320 22431 35392 23931 68119 26204 41736 81298 60142 92748 14557 5778 52606 13381 15568 9921 31836 43069 89574 50855 53195 77819 99321 3121 76655 36293 19818 75294 5960 27516 22543 85039 56297 77826 89726 78821 44945 35723 52964 94704 40357 92472 29551 19616 7073 87976 90056 90354 231 64438 31906 85722 86064 84627 1222 91733 2017 66697 49505 93745 65190 18906 55321 76076 35088 28477 46864 92384 64966 66321 7104 24991 8808 82851 9035 39684 73566 73079 69066 2141 78248 83335 99219 60803 92940 72300 83815 59641 12141 49758 20957 22320 14214 5130 22935 75644 77745 96417 70897 54398 27384 75782 70288 48405 38803 29669 68819 19240 49314 37117 45966 53263 71334 94144 44746 4969 80159 24288 77038 70450 5899 88262 42981 47946 22731 48385 26120 81372 74674 20601 6973 23697 95636 33579 51253 81068 56913 45309 34893 66647 92812 93382 18352 71110 54314 22894 91011 40456 76626 16926 14836 55770 10089 26486 61827 10193 9231 71519 86350 38050 91741 58098 65158 85018 14534 95915 42878 53869 19825 83248 77030 57426 57399 13515 63679 15571 68051 9711 15883 58578 43546 94389 47297 13605 59258 66186 85900 79833 84501 83307 44642 61735 33054 76027 87805 56583 6406 9193 13900 66673 64562 40489 40751 98349 37907 46047 31146 24676 51941 3153 92020 51125 45201 61799 31424 86072 46468 82428 72821 87532 37648 3147 72564 4167 34942 89208 63187 30221 28974 42709 21661 15535 90153 4894 92317 32014 81515 36395 66671 48571 52844 27244 65980 90082 71738 57561 56837 32236 14773 28088 44044 74620 77155 31873 73396 82946 60946 56076 70797 48648 16858 70000 68587 85091 66140 42194 92875 49088 45270 36115 17942 15477 49506 47433 79739 46529 72873 44141 44862 85453 66327 23584 73061 95369 87191 71847 98413 17427 19945 30859 75756 18995 16333 16535 91512 50172 92567 29704 60849 74736 99836 93794 66484 52740 22731 35339 60205 39725 24480 21827 54591 64746 35461 11076 1015 88599 90971 51987 37295 26622 54406 92681 90649 55447 12078 79465 60484 58928 64056 65486 11420 97416 55287 92208 66218 53071 36609 532 68031 62600 71376 48145 44431 74847 2066 6715 86053 88044 98076 13971 62286 30932 31485 98950 35056 67589 16143 32894 63828 44397 20953 85330 7172 11317 42230 55009 85490 93994 72392 45333 12476 17498 63626 58940 24835 86508 49931 89760 99060 34569 43637 31968 59708 3793 73717 53365 55264 43330 6218 75577 58845 70497 75965 32116 13075 64263 14330 50610 76348 80498 32112 89089 97438 2939 82898 47900 69486 65775 30175 13707 576 66797 36351 24917 46227 41368 17481			
✓	6 1 1 2 3 4 5 6	2	2	✓
✓	2 3 1 6	2	2	✓
✓	5 3 1 4 5 9 15	3	3	✓

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 9

Đúng

Đạt điểm 10,00 trên 10,00

Cho một dãy A gồm N số nguyên và số nguyên K . Đếm số lượng cặp số nguyên trong dãy A có chênh lệch bằng K .

Đầu vào

Dữ liệu vào từ bàn phím gồm 2 dòng. Dòng đầu tiên chứa 2 số nguyên N và K ($0 < N \leq 10^5, 0 \leq K \leq 10^9$). Dòng thứ 2 chứa N số nguyên a_i thuộc dãy A ($a_i < 2^{31} - 1$).

Đầu ra

In ra màn hình một dòng duy nhất chứa số lượng cặp số nguyên có chênh lệch là K .

For example:

Input	Result
5 2 1 5 3 4 2	3

Answer: (penalty regime: 0 %)

```
1 #include <algorithm>
2 #include <iostream>
3 #include <iterator>
4 #include <map>
5 #include <set>
6 #include <stack>
7 #include <string>
8 #include <vector>
9 using namespace std;
10
11 int main() {
12     int n, k;
13     cin >> n >> k;
14     map<int, int> mp;
15     for (int i = 0; i < n; ++i) {
16         int x;
17         cin >> x;
18         ++mp[x];
19     }
20     int ans = 0;
21     for (auto [i, c] : mp) {
22         if (k == 0) {
23             ans += c * (c - 1) / 2;
24         } else if (mp.count(i + k)) {
25             ans += c * mp[i + k];
26         }
27     }
28     cout << ans;
29 }
```

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Trở lại Khoá học

