

Status	Finished
Started	Sunday, 1 December 2024, 10:10 PM
Completed	Tuesday, 3 December 2024, 8:18 PM
Duration	1 day 22 hours
Marks	100.00/100.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 10.00 out of 10.00

[DigitsSort]

Viết hàm sắp xếp các chữ số của số nguyên dương n ($n \leq 10^{100}$) theo thứ tự tăng dần.

Đầu vào

Đầu vào từ bàn phím gồm $T + 1$ dòng

- Dòng đầu tiên chứa số nguyên T là số lượng số cần kiểm tra;
- T dòng tiếp theo, mỗi dòng chứa một số nguyên n .

Đầu ra

In ra màn hình T dòng, mỗi dòng ghi ra số nguyên n sau khi đã sắp xếp.

For example:

Input	Result
1	12345
52341	

Answer: (penalty regime: 0 %)

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int main()
9  {
10     ios_base::sync_with_stdio(0);
11     cin.tie(0); cout.tie(0);
12     int T; cin >> T;
13     while(T--)
14     {
15         string s; cin >> s;
16         sort(s.begin(), s.end());
17         cout << s << el;
18     }
19     return 0;
20 }
21
```

	Input	Expected	Got	
✓	1 52341	12345	12345	✓
✓	3 982347656123 12134 120	122334566789 11234 012	122334566789 11234 012	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 2

Correct

Mark 10.00 out of 10.00

[PrimeNumbersSort]

Cho một mảng số nguyên.

Viết chương trình sắp xếp các phần tử số nguyên tố trong mảng đó theo thứ tự tăng dần (giữ nguyên vị trí của các phần tử khác trong mảng).

Đầu vào

Đầu vào từ bàn phím gồm hai dòng:

- Dòng đầu tiên chứa số nguyên n ($n \leq 100$) là số lượng phần tử của mảng;
- Dòng thứ hai chứa n số nguyên là các phần tử của mảng đó. Các phần tử được phân tách bởi một dấu cách.

Đầu ra

In ra màn hình các phần tử của mảng sau khi đã sắp xếp, phân tách các phần tử bởi **duy nhất một dấu cách**.

For example:

Input	Result
5	100 2 6 5 48
100 5 6 2 48	

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  const int mod=1e9+7;
7
8  int check(int n)
9  {
10     for(int i = 2; i < sqrt(n); i++)
11     {
12         if(n % i == 0) return 0;
13     }
14     return n > 1;
15 }
16 int main()
17 {
18     ios_base::sync_with_stdio(0);
19     cin.tie(0); cout.tie(0);
20     int n; cin >> n;
21     int a[n];
22     vector<int> vt1, vt2;
23     for(int i = 0; i < n; i++)
24     {
25         cin >> a[i];
26         if(check(a[i]))
27         {
28             vt1.push_back(a[i]);
29             vt2.push_back(i);
30         }
31     }
32     sort(vt1.begin(), vt1.end());
33     int m = vt1.size();
34     for(int i = 0; i < m; i++)
35     {
36         a[vt2[i]] = vt1[i];
37     }

```

```
38     for(auto &x : a) cout << x << " ";
39     return 0;
40 }
41
```

	Input	Expected	Got	
✓	5 100 5 6 2 48	100 2 6 5 48	100 2 6 5 48	✓
✓	8 100 5 6 2 48 7 200 300	100 2 6 5 48 7 200 300	100 2 6 5 48 7 200 300	✓
✓	10 100 5 6 2 48 7 200 300 17 20	100 2 6 5 48 7 200 300 17 20	100 2 6 5 48 7 200 300 17 20	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 3

Correct

Mark 10.00 out of 10.00

[CanPlaceFlowers]

Giả sử bạn là nhân viên của một công ty cây xanh, được giao cho chăm sóc một luống đất. Luống đất đó được chia thành nhiều ô đất nhỏ có kích cỡ bằng nhau.

Trước khi được giao cho bạn chăm sóc, trên luống hoa đã được trồng sẵn một số cây xanh tán rộng.

Một ngày đầu xuân, công ty gửi đến cho bạn một số cây hoa và yêu cầu bạn trồng vào những vị trí còn trống trong luống hoa.

Đây là những cây hoa hướng sáng và chúng không thể sống được dưới bóng râm của cây khác, đặc biệt là cây tán rộng. Vì vậy, bạn không thể trồng chúng tại ô bên cạnh những ô đã được trồng cây xanh tán rộng.

Và nếu trồng những cây hoa bên cạnh nhau, chúng sẽ tranh giành phân bón và nước và sau cùng, cả hai cây sẽ cùng chết.

Viết hàm `bool canPlaceFlowers (int flowerbed[], int n, int k)` kiểm tra xem có thể trồng hết số hoa mới được cấp vào trong luống đất của bạn hay không.

Hàm nhận đầu vào là mảng *flowerbed* ứng với luống đất có *n* ô đất, đánh dấu những vị trí trong luống đất đã được trồng cây xanh trước đó (giá trị 1 ứng với vị trí cây đã được trồng và 0 ứng với vị trí đất trống) và số lượng hoa được phát thêm *k*. Hàm trả về *true* nếu có thể trồng hết toàn bộ *k* cây hoa vào luống đất và *false* trong trường hợp ngược lại.

For example:

Test	Input	Result
canPlaceFlowers(flowerbed, n, k)	10 1 0 0 0 1 1 1 1 1 0 1	true

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
11 const int mod=1e9+7;
12
13 bool canPlaceFlowers (int flowerbed[], int n, int k)
14 {
15     int ans = 0;
16     int b[n];
17     for(int i = 0; i < n; i++)
18     {
19         b[i]=flowerbed[i];
20     }
21     for(int i = 0; i < n; i++)
22     {
23         if(i - 1 >= 0 && flowerbed[i] == 1 )
24         {
25             b[i-1] = 1;
26         }
27         if(i + 1 < n && flowerbed[i] == 1 )
28         {
29             b[i+1] = 1;
30         }
31     }
32     int dem = 0;

```

```

33     for(int i = 0; i < n; i++)
34     {
35         if(b[i] == 0)
36         {
37             dem++;
38         }
39         else
40         {
41             ans += (dem+1) / 2;
42             dem = 0;
43         }
44     }
45     ans += (dem+1)/2;
46     if(ans >= k) return true;
47     else return false;
48 }
49 void Print()
50 {
51     Faster;
52     int n, k; cin >> n >> k;

```

	Test	Input	Expected	Got	
✓	canPlaceFlowers(flowerbed, n, k)	10 1 0 0 0 1 1 1 1 1 0 1	true	true	✓
✓	canPlaceFlowers(flowerbed, n, k)	10 1 1 0 1 0 1 0 0 1 0 1	false	false	✓
✓	canPlaceFlowers(flowerbed, n, k)	30 6 0 1 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0	true	true	✓
✓	canPlaceFlowers(flowerbed, n, k)	70 12 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 0 0 1 0 1 1 0 1 0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 0 0 0 0 1 0 1 0 1 0 0 0 1 1 1 1 0	false	false	✓
✓	canPlaceFlowers(flowerbed, n, k)	70 12 1 1 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 1 1 1 1 0 1 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 1 1 0 1 0 0 0 0 1 1 1 1 0 1 0 1 0 1 1 0 0 0 1 1 0 0 1 0 0 0 1	false	false	✓
✓	canPlaceFlowers(flowerbed, n, k)	80 15 1 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 1 0 0 1 0 0 0 1 0 0 1 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 1 1 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 1 0 1 1 1 0 1 0 0 0 1	false	false	✓
✓	canPlaceFlowers(flowerbed, n, k)	150 13 1 0 0 0 0 0 1 0 1 0 1 1 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 1 1 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 1 1 0 0 0 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 1 1 1 1 1	true	true	✓

	Test	Input	Expected	Got	
✓	canPlaceFlowers(flowerbed, n, k)	180 25 1 0 0 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 1 0 0 0 1 0 0 0 1 1 1 0 1 1 1 1 1 1 0 1 0 1 0 1 1 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 1 0 0 1 1 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 1 0 0 0 0 0 1 1 1 1 0 0 1 0 0 1 0 0 1 0 0 1 0	false	false	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 4

Correct

Mark 10.00 out of 10.00

[Intertwined]

Viết chương trình nhập vào số nguyên n là độ dài của hai mảng số nguyên a và b , hai dòng tiếp theo là các phần tử thuộc a và b . In ra màn hình dãy số nguyên đan xen các phần tử của hai mảng trên (phần tử đầu của dãy a rồi đến phần tử đầu dãy b ,...).

For example:

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

Answer: (penalty regime: 0 %)

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define ll long long
4 #define ull unsigned long long
5 #define el "\n"
6 #define se second
7 #define fi first
8 #define en end()
9 #define be begin()
10 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
11 const int mod=1e9+7;
12
13 int main()
14 {
15     Faster;
16     int n; cin >> n;
17     int a[n], b[n];
18     for(auto &x : a) cin >> x;
19     for(auto &x : b) cin >> x;
20     for(int i = 0; i < n; i++)
21     {
22         cout << a[i] << " " << b[i] << " ";
23     }
24     return 0;
25 }
26
```

	Input	Expected	Got	
✓	3 1 3 5 2 4 6	1 2 3 4 5 6	1 2 3 4 5 6	✓

	Input	Expected	Got	
✓	4 12 17 2 9 -1 3 4 2	12 -1 17 3 2 4 9 2	12 -1 17 3 2 4 9 2	✓
✓	1 1 2	1 2	1 2	✓
✓	2 2 2 2 2	2 2 2 2	2 2 2 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 5

Correct

Mark 10.00 out of 10.00

[NeverGiveYouUp]

Sau khi chấm xong điểm thi giữa kỳ môn Tin học cơ sở 4, các thầy/cô muốn sắp xếp điểm đã chấm theo thứ tự giảm dần. Hãy viết chương trình giúp các thầy/cô thực hiện việc này.

Đầu vào: số nguyên n là số lớp trên một dòng và n điểm của các sinh viên trong lớp ở dòng tiếp theo.

Đầu ra: điểm của các sinh viên đã được sắp thứ tự giảm dần.

For example:

Input	Result
5 98 63 21 26 53	98 63 53 26 21

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
11 const int mod=1e9+7;
12
13 int main()
14 {
15     Faster;
16     int n; cin >> n;
17     int a[n];
18     for(auto &x : a) cin >> x;
19     sort(a, a + n, greater<int>());
20     for(auto &x : a) cout << x << " ";
21     return 0;
22 }
23

```

	Input	Expected	Got	
✓	5 98 63 21 26 53	98 63 53 26 21	98 63 53 26 21	✓

	Input	Expected	Got	
✓	30 84 85 33 77 58 34 15 96 6 36 78 68 62 59 29 83 54 39 22 74 91 36 2 87 42 20 36 62 3 71	96 91 87 85 84 83 78 77 74 71 68 62 62 59 58 54 42 39 36 36 36 34 33 29 22 20 15 6 3 2	96 91 87 85 84 83 78 77 74 71 68 62 62 59 58 54 42 39 36 36 36 34 33 29 22 20 15 6 3 2	✓
✓	40 63 96 87 59 27 87 74 26 46 26 18 74 20 44 88 18 91 76 77 97 86 38 35 18 93 6 50 67 71 7 48 67 78 28 17 66 12 55 31 98	98 97 96 93 91 88 87 87 86 78 77 76 74 74 71 67 67 66 63 59 55 50 48 46 44 38 35 31 28 27 26 26 20 18 18 18 17 12 7 6	98 97 96 93 91 88 87 87 86 78 77 76 74 74 71 67 67 66 63 59 55 50 48 46 44 38 35 31 28 27 26 26 20 18 18 18 17 12 7 6	✓
✓	70 42 80 15 27 60 80 59 75 32 0 93 21 88 24 68 65 96 40 33 90 99 30 58 19 78 36 89 93 59 70 8 26 80 84 24 50 12 40 88 97 94 73 54 56 81 65 83 21 2 48 1 32 81 4 65 14 62 90 17 35 42 45 79 9 1 71 24 21 60 51	99 97 96 94 93 93 90 90 89 88 88 84 83 81 81 80 80 80 79 78 75 73 71 70 68 65 65 65 62 60 60 59 59 58 56 54 51 50 48 45 42 42 40 40 36 35 33 32 32 30 27 26 24 24 24 21 21 21 19 17 15 14 12 9 8 4 2 1 1 0	99 97 96 94 93 93 90 90 89 88 88 84 83 81 81 80 80 80 79 78 75 73 71 70 68 65 65 65 62 60 60 59 59 58 56 54 51 50 48 45 42 42 40 40 36 35 33 32 32 30 27 26 24 24 24 21 21 21 19 17 15 14 12 9 8 4 2 1 1 0	✓
✓	100 23 45 44 46 49 75 92 11 68 17 9 71 24 79 2 11 18 80 73 56 98 31 3 39 41 29 48 34 36 85 66 39 13 34 11 4 45 96 39 76 55 84 63 9 49 86 27 52 61 7 32 44 63 44 59 22 54 78 36 77 53 23 55 31 10 74 46 42 21 79 36 62 92 9 81 43 16 33 96 11 94 54 84 84 51 5 68 31 97 60 79 69 52 33 26 58 86 67 26 21	98 97 96 96 94 92 92 86 86 85 84 84 84 81 80 79 79 79 78 77 76 75 74 73 71 69 68 68 67 66 63 63 62 61 60 59 58 56 55 55 54 54 53 52 52 51 49 49 48 46 46 45 45 44 44 44 43 42 41 39 39 39 36 36 36 34 34 33 33 32 31 31 31 29 27 26 26 24 23 23 22 21 21 18 17 16 13 11 11 11 11 10 9 9 9 7 5 4 3 2	98 97 96 96 94 92 92 86 86 85 84 84 84 81 80 79 79 79 78 77 76 75 74 73 71 69 68 68 67 66 63 63 62 61 60 59 58 56 55 55 54 54 53 52 52 51 49 49 48 46 46 45 45 44 44 44 43 42 41 39 39 39 36 36 36 34 34 33 33 32 31 31 31 29 27 26 26 24 23 23 22 21 21 18 17 16 13 11 11 11 11 10 9 9 9 7 5 4 3 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 6

Correct

Mark 10.00 out of 10.00

[PointDescending]

Cuối kỳ, điểm tổng kết môn Toán của khối lớp 8 gồm $(n \setminus)$ học sinh được lưu trong một mảng $(1 \setminus)$ chiều.

Viết chương trình sắp xếp lại bảng điểm trên theo thứ tự giảm dần và in ra màn hình.

Kết quả làm tròn đến chữ số thập phân thứ hai.

For example:

Input	Result
15	9.83 8.81 7.87 7.45 5.00 4.96
2.91364701067 4.92844392133 0.784151561401 7.45297643936 0.361399762262 4.95631157295	4.93 4.27 3.90 2.91 2.68 1.34
9.83352340969 0.422081509266 7.86714702984 3.89690308818 4.99809841067 4.26767489958	0.78 0.42 0.36
1.34403895566 2.68317658315 8.80729068767	

Answer: (penalty regime: 0 %)

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 #define ll long long
4 #define ull unsigned long long
5 #define el "\n"
6 #define se second
7 #define fi first
8 #define en end()
9 #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13
14 int main()
15 {
16     Faster;
17     int n; cin >> n;
18     double a[n];
19     for(auto &x : a) cin >> x;
20     sort(a, a + n, greater<double>());
21     for(auto &x : a) cout << fixed << setprecision(2) << x << " ";
22     return 0;
23 }
24
```

	Input	Expected	Got	
✓	15 2.91364701067 4.92844392133 0.784151561401 7.45297643936 0.361399762262 4.95631157295 9.83352340969 0.422081509266 7.86714702984 3.89690308818 4.99809841067 4.26767489958 1.34403895566 2.68317658315 8.80729068767	9.83 8.81 7.87 7.45 5.00 4.96 4.93 4.27 3.90 2.91 2.68 1.34 0.78 0.42 0.36	9.83 8.81 7.87 7.45 5.00 4.96 4.93 4.27 3.90 2.91 2.68 1.34 0.78 0.42 0.36	✓
✓	15 7.21907452705 0.878577101963 6.03602923324 2.0344845004 7.82063737578 0.0973769359693 9.85169780453 4.99579173751 2.12118676212 6.5749396641 5.4901426112 5.7066330807 1.94822014317 0.486663835645 2.10218967251	9.85 7.82 7.22 6.57 6.04 5.71 5.49 5.00 2.12 2.10 2.03 1.95 0.88 0.49 0.10	9.85 7.82 7.22 6.57 6.04 5.71 5.49 5.00 2.12 2.10 2.03 1.95 0.88 0.49 0.10	✓
✓	64 3.95099465183 6.06838977332 0.0164275631446 1.86036619917 8.87952069096 5.20612836225 3.42043894504 1.1822061393 1.28909831655 3.0122058109 8.4557139917 7.59094093014 6.08204202224 4.61262238477 5.82718301466 0.447329310916 2.1478509568 4.01877005193 8.73892996038 6.76894172826 4.91166571455 3.89899577511 6.51128399693 1.10590894738 3.97633618145 1.02364355 9.1348243227 4.81412062382 0.903557551251 0.219100557646 2.44014397834 7.08629565333 5.4821901029 4.01407442789 9.41956675047 9.15692070719 8.54966376937 7.85165964829 3.86802883902 7.97138659653 5.16006208026 8.65787973474 9.91118920464 9.85744322835 5.88756616857 3.08065988506 5.48382637378 7.4467593248 0.650610983252 3.2452573593 8.48145998543 6.8816624943 4.22617759977 1.86792043726 8.22716898617 2.53837841855 7.55337075504 9.70518094912 1.0896204347 8.80427499457 5.8058084723 4.89467389872 4.66547006462 0.73433507502	9.91 9.86 9.71 9.42 9.16 9.13 8.88 8.80 8.74 8.66 8.55 8.48 8.46 8.23 7.97 7.85 7.59 7.55 7.45 7.09 6.88 6.77 6.51 6.08 6.07 5.89 5.83 5.81 5.48 5.48 5.21 5.16 4.91 4.89 4.81 4.67 4.61 4.23 4.02 4.01 3.98 3.95 3.90 3.87 3.42 3.25 3.08 3.01 2.54 2.44 2.15 1.87 1.86 1.29 1.18 1.11 1.09 1.02 0.90 0.73 0.65 0.45 0.22 0.02	9.91 9.86 9.71 9.42 9.16 9.13 8.88 8.80 8.74 8.66 8.55 8.48 8.46 8.23 7.97 7.85 7.59 7.55 7.45 7.09 6.88 6.77 6.51 6.08 6.07 5.89 5.83 5.81 5.48 5.48 5.21 5.16 4.91 4.89 4.81 4.67 4.61 4.23 4.02 4.01 3.98 3.95 3.90 3.87 3.42 3.25 3.08 3.01 2.54 2.44 2.15 1.87 1.86 1.29 1.18 1.11 1.09 1.02 0.90 0.73 0.65 0.45 0.22 0.02	✓
✓	50 0.0741714086618 5.43687901272 4.07503385353 2.4824629193 8.81970969244 7.75428463969 5.97683102192 8.00579224903 2.93905789007 9.48129815574 0.281702907428 9.74289323701 3.88958106316 3.34491007016 3.56483318362 4.86139136711 4.92545326481 7.24268013018 0.100441367638 2.28883514867 5.74507682197 8.03665952276 1.55448989326 3.717053409 5.80301526722 4.25425101717 0.0906161821488 2.45414969522 7.72282864665 5.92278384375 0.381565547688 9.400300637 9.93231718438 3.47965901516 1.76299262558 1.04882972569 9.55736675903 4.64661229694 0.170985237801 9.28945704467 9.39046872324 9.41032878728 9.01317566388 2.32060740125 5.3059668099 8.03099019768 3.05599863386 6.20095258786 1.56075483891 3.5773748678	9.93 9.74 9.56 9.48 9.41 9.40 9.39 9.29 9.01 8.82 8.04 8.03 8.01 7.75 7.72 7.24 6.20 5.98 5.92 5.80 5.75 5.44 5.31 4.93 4.86 4.65 4.25 4.08 3.89 3.72 3.58 3.56 3.48 3.34 3.06 2.94 2.48 2.45 2.32 2.29 1.76 1.56 1.55 1.05 0.38 0.28 0.17 0.10 0.09 0.07	9.93 9.74 9.56 9.48 9.41 9.40 9.39 9.29 9.01 8.82 8.04 8.03 8.01 7.75 7.72 7.24 6.20 5.98 5.92 5.80 5.75 5.44 5.31 4.93 4.86 4.65 4.25 4.08 3.89 3.72 3.58 3.56 3.48 3.34 3.06 2.94 2.48 2.45 2.32 2.29 1.76 1.56 1.55 1.05 0.38 0.28 0.17 0.10 0.09 0.07	✓

	Input	Expected	Got	
✓	100 6.02363457817 1.03751478679 3.95424895573 4.64900568732 6.15685232504 4.69477931288 4.33045289687 9.92522336289 6.37452834406 4.37406622044 4.54879061203 5.3607432809 8.58396699028 3.17022021645 3.45607804851 7.36153049517 8.32142061282 3.37115848485 0.103793377186 9.84777856167 8.77701821236 3.96936345037 2.68479477243 9.48345075194 1.95830625875 6.26443961343 0.949558579474 4.04772624778 7.35577040304 9.54990329081 9.11916102569 4.85494494196 2.35840339262 2.22828914282 2.08092852132 6.52345163876 8.63687991577 9.12791740576 5.08087893368 7.64565085932 4.42890184999 6.78153014939 1.73978560159 3.77106567011 6.43985504336 3.18426649128 4.5667865619 1.62650108724 7.7146180378 7.88459882156 4.53775202402 1.01008238859 3.45300534204 2.00786312967 7.35295492788 3.31062097221 7.76281552081 8.90183087789 7.84318960999 7.65656776588 8.79804462553 7.71913461401 9.67299678433 2.16311521986 8.56978594934 5.67198280582 1.82764831079 4.81067511068 3.77373580496 0.852715091963 0.427489331616 0.224019853832 1.56351271089 1.70027245321 2.39758975492 0.825901690989 3.42203402741 7.75712495366 9.7978288058 6.54866661693 4.55355101011 4.0751057731 0.346668819026 4.07818823455 6.4086252309 2.91937249902 0.863522553346 6.86891048101 4.34035464693 7.70519903902 4.42834639067 8.01887159755 4.63653370644 5.83722675209 1.00548717073 1.27120910259 0.520381954476 3.8778262136 3.69788997338 2.1005319756	9.93 9.85 9.80 9.67 9.55 9.48 9.13 9.12 8.90 8.80 8.78 8.64 8.58 8.57 8.32 8.02 7.88 7.84 7.76 7.76 7.72 7.71 7.71 7.66 7.65 7.36 7.36 7.35 6.87 6.78 6.55 6.52 6.44 6.41 6.37 6.26 6.16 6.02 5.84 5.67 5.36 5.08 4.85 4.81 4.69 4.65 4.64 4.57 4.55 4.55 4.54 4.43 4.43 4.37 4.34 4.33 4.08 4.08 4.05 3.97 3.95 3.88 3.77 3.77 3.70 3.46 3.45 3.42 3.37 3.31 3.18 3.17 2.92 2.68 2.40 2.36 2.23 2.16 2.10 2.08 2.01 1.96 1.83 1.74 1.70 1.63 1.56 1.27 1.04 1.01 1.01 0.95 0.86 0.85 0.83 0.52 0.43 0.35 0.22 0.10	9.93 9.85 9.80 9.67 9.55 9.48 9.13 9.12 8.90 8.80 8.78 8.64 8.58 8.57 8.32 8.02 7.88 7.84 7.76 7.76 7.72 7.71 7.71 7.66 7.65 7.36 7.36 7.35 6.87 6.78 6.55 6.52 6.44 6.41 6.37 6.26 6.16 6.02 5.84 5.67 5.36 5.08 4.85 4.81 4.69 4.65 4.64 4.57 4.55 4.55 4.54 4.43 4.43 4.37 4.34 4.33 4.08 4.08 4.05 3.97 3.95 3.88 3.77 3.77 3.70 3.46 3.45 3.42 3.37 3.31 3.18 3.17 2.92 2.68 2.40 2.36 2.23 2.16 2.10 2.08 2.01 1.96 1.83 1.74 1.70 1.63 1.56 1.27 1.04 1.01 1.01 0.95 0.86 0.85 0.83 0.52 0.43 0.35 0.22 0.10	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 7

Correct

Mark 10.00 out of 10.00

[ReversedArray]

Viết chương trình nhập vào một mảng số thực có độ dài (n) và in ra màn hình dãy đó theo thứ tự đảo ngược.

Kết quả làm tròn đến chữ số thập phân thứ hai.

For example:

Input	Result
4 1 2 3 4	4.00 3.00 2.00 1.00

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13
14 int main()
15 {
16     Faster;
17     int n; cin >> n;
18     double a[n];
19     for(auto &x : a) cin >> x;
20     for(int i = n - 1; i >= 0; i--) cout << fixed << setprecision(2) << a[i] << " ";
21     return 0;
22 }
23

```

	Input	Expected	Got	
✓	4 1 2 3 4	4.00 3.00 2.00 1.00	4.00 3.00 2.00 1.00	✓
✓	5 91 23 -12.324 21.12323 21	21.00 21.12 -12.32 23.00 91.00	21.00 21.12 -12.32 23.00 91.00	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 8

Correct

Mark 10.00 out of 10.00

[ConditionalSort]

Sắp xếp là nhu cầu thường thấy khi làm việc với dữ liệu, dù là kiểu kí tự hay kiểu số.

Viết hàm `void sort(float array[], int size, bool isAscending)` nhận tham số đầu vào là một mảng số thực có kích cỡ `\(size\)`.

Nếu `\(isAscending\)` nhận giá trị `\(true\)`, sắp xếp mảng theo thứ tự tăng dần, ngược lại, sắp xếp mảng theo thứ tự giảm dần.

For example:

Test	Input	Result
sort(array, n, true)	4 2 3 1 6	1.00 2.00 3.00 6.00

Answer: (penalty regime: 0 %)

```

1  #include <iostream>
2
3  void sort(float array[], int size, bool isAscending) {
4      for (int i = 0; i < size - 1; i++) {
5          for (int j = i + 1; j < size; j++) {
6              if (isAscending) {
7                  if (array[i] > array[j]) {
8                      std::swap(array[i], array[j]);
9                  }
10             } else {
11                 if (array[i] < array[j]) {
12                     std::swap(array[i], array[j]);
13                 }
14             }
15         }
16     }
17 }
18
19 void Print() {
20     int size;
21     bool isAscending;
22
23     std::cin >> size;
24
25     float array[size];
26
27     for (int i = 0; i < size; i++) {
28         std::cin >> array[i];
29     }
30
31     std::cin >> isAscending;
32
33     sort(array, size, isAscending);
34
35     for (int i = 0; i < size; i++) {
36         std::cout << array[i] << " ";
37     }
38
39 }
40

```

	Test	Input	Expected	Got	
✓	sort(array, n, true)	4 2 3 1 6	1.00 2.00 3.00 6.00	1.00 2.00 3.00 6.00	✓
✓	sort(array, n, true)	10 10.6579176171 -7.44530488421 -15.2651157251 -1.10895469456 -18.4388600308 4.9880183654 3.94724364022 16.2997304164 -3.30393854802 5.07280092833	-18.44 -15.27 -7.45 -3.30 -1.11 3.95 4.99 5.07 10.66 16.30	-18.44 -15.27 -7.45 -3.30 -1.11 3.95 4.99 5.07 10.66 16.30	✓
✓	sort(array, n, false)	20 -14.9901942752 19.9444193075 5.63863739243 -1.20820585544 -8.46026052206 21.5954131858 4.20688426553 3.13787226372 9.28173072626 10.8288242252 11.998278715 -16.6635788226 4.88816372857 5.06370205692 4.94745541204 -4.11308394768 -8.87581617955 -1.79038201065 -12.2754818716 8.15407638295	21.60 19.94 12.00 10.83 9.28 8.15 5.64 5.06 4.95 4.89 4.21 3.14 -1.21 -1.79 -4.11 -8.46 -8.88 -12.28 -14.99 -16.66	21.60 19.94 12.00 10.83 9.28 8.15 5.64 5.06 4.95 4.89 4.21 3.14 -1.21 -1.79 -4.11 -8.46 -8.88 -12.28 -14.99 -16.66	✓
✓	sort(array, n, false)	25 14.8934039584 -0.243600485587 -3.08316266829 -7.30696165787 8.28652890274 -0.593930434044 -27.669322874 2.54311532448 0.530580383512 -3.11106906328 -5.35475589182 -13.5145275252 7.95320983418 -12.4114171679 -24.1169997076 13.2799786583 -12.8804061711 -13.0915248803 4.28232477424 4.01351356736 -8.01188399809 23.8761836184 4.42337570597 -4.00886028421 -6.67621010736	23.88 14.89 13.28 8.29 7.95 4.42 4.28 4.01 2.54 0.53 -0.24 -0.59 -3.08 -3.11 -4.01 -5.35 -6.68 -7.31 -8.01 -12.41 -12.88 -13.09 -13.51 -24.12 -27.67	23.88 14.89 13.28 8.29 7.95 4.42 4.28 4.01 2.54 0.53 -0.24 -0.59 -3.08 -3.11 -4.01 -5.35 -6.68 -7.31 -8.01 -12.41 -12.88 -13.09 -13.51 -24.12 -27.67	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 9

Correct

Mark 10.00 out of 10.00

[SymmetricArray]

Một mảng số nguyên có độ dài (n) được cho là đối xứng nếu phần tử thứ (k) trong mảng $((1 \leq k \leq n))$ có giá trị bằng phần tử thứ $(n - k + 1)$.

Viết chương trình nhập vào mảng số nguyên có (n) phần tử, kiểm tra xem mảng đó có đối xứng hay không.

Nếu có, in ra màn hình "Symmetric array.", ngược lại, in ra màn hình "Asymmetric array."

For example:

Input	Result
4 2 3 3 2	Symmetric array.
5 1 2 3 1 3	Asymmetric array.

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13 int main()
14 {
15     Faster;
16     int n; cin >> n;
17     int a[n];
18     for(auto &x : a) cin >> x;
19     int i = 0, j = n - 1;
20     while(i++ <= j--){
21         if(a[i] != a[j])
22         {
23             cout << "Asymmetric array.";
24             return 0;
25         }
26     }
27     cout << "Symmetric array.";
28     return 0;
29 }
30
31
32

```

	Input	Expected	Got	
✓	4 2 3 3 2	Symmetric array.	Symmetric array.	✓
✓	5 1 2 3 1 3	Asymmetric array.	Asymmetric array.	✓
✓	20 25 65 76 33 68 39 86 83 22 28 38 20 4 4 62 69 99 82 35 10	Asymmetric array.	Asymmetric array.	✓
✓	21 44 80 29 67 40 65 40 28 39 64 15 64 39 28 40 65 40 67 29 80 44	Symmetric array.	Symmetric array.	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 10

Correct

Mark 10.00 out of 10.00

[TheLastSurvivor]

Trong một trò chơi sinh tử, khi $(n \setminus)$ người bắt buộc phải chấp nhận tham gia một trò chơi, trong đó, chỉ có duy nhất một người có quyền sống sót.

Trong trò chơi này, $(n \setminus)$ người ngồi thành một vòng tròn quanh một chiếc bàn. Một khẩu súng được đặt ở giữa bàn và trong đó có $(n - 1 \setminus)$ viên đạn đã được nạp sẵn.

Khẩu súng sẽ được chuyển theo chiều kim đồng hồ bắt đầu từ vị trí số $(1 \setminus)$ (vị trí 12 giờ). Khi khẩu súng được chuyển qua $(k \setminus)$ người thì người thứ $(k + 1 \setminus)$ sẽ bị hành quyết $((k < n \setminus))$

Quy luật sẽ lặp đi lặp lại với số người còn lại trong vòng tròn đó, bắt đầu với người tiếp theo, theo cùng một chiều quay, và bỏ qua cùng số người $(k \setminus)$ cho đến khi nào chỉ còn một người sống sót.

Giả sử bạn là một người trong trò chơi sinh tử này, hãy viết chương trình nhận đầu vào là số người tham gia trò chơi $(n \setminus)$ và vòng đếm $(k \setminus)$, tính ra vị trí an toàn trong vòng tròn để bạn là người sống sót cuối cùng.

For example:

Input	Result
5 3 Gabriela Krantz Wanita Strickler Micki Whyte Louisa Fell Latosha Boor	Gabriela Krantz

Answer: (penalty regime: 0 %)

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define ull unsigned long long
5  #define el "\n"
6  #define se second
7  #define fi first
8  #define en end()
9  #define be begin()
10 #define sz size()
11 #define Faster ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
12 const int mod=1e9+7;
13 int main()
14 {
15     Faster;
16     int n, k; cin >> n >> k;
17     cin.ignore();
18     string a[n + 1];
19     vector<string> vt;
20     for(int i = 0; i < n; i++)
21     {
22         getline(cin, a[i]);
23         vt.push_back(a[i]);
24     }
25     int it = 0;
26     while(vt.size() != 1)
27     {
28         it += k;
29         int m = vt.size();
30         while(it > m - 1) it -= vt.size();

```

```
31         vt.erase(vt.be + it);
32     }
33     cout << vt[0];
34     return 0;
35 }
36
37
```

	Input	Expected	Got	
✓	15 13 Louisa Fell Kyle Flenniken Joni Kellog Shanti Newhart Karissa Wagener Marion Batdorf Analisa Stinson Florene Cianci Myrtle Gunter Armandina Seagraves Darla Palencia Hortense Parkey Cindy Avilla Epifania Deluca Robbin Tung	Hortense Parkey	Hortense Parkey	✓
✓	5 3 Gabriela Krantz Wanita Strickler Micki Whyte Louisa Fell Latosha Boor	Gabriela Krantz	Gabriela Krantz	✓

	Input	Expected	Got	
✓	25 10 Candis Felter Florene Cianci Shena Carls Wanita Strickler Kyle Flenniken Analisa Stinson Millicent Reppert Lean Broadhurst Armandina Seagraves Marion Batdorf Theressa Kemmer Robbin Tung Cindy Avilla Sarah Tumlin Justine Hoke Lavada Mcninch Carman Capshaw Jana Granberry Micki Whyte Delpha Johnson Epifania Deluca Jinny Stadler Kimbery Pittenger Joni Kellog Darla Palencia	Candis Felter	Candis Felter	✓

	Input	Expected	Got	
✓	50 23 Jana Granberry Garnet Sheilds Theressa Kemmer Latosha Boor Joni Kellog Carman Capshaw Myrtle Gunter Robbin Tung Micki Whyte Kimbery Pittenger Marion Batdorf Cindy Avilla Leana Monaghan Sarah Tumlin Audrey Kidd Shanti Newhart Astrid Farish Verlie Schier Jenine McGavock Shena Carls Millicent Reppert Analisa Stinson Dollie Burns Epifania Deluca Gabriela Krantz Herlinda Swisher Candis Felter Armandina Seagraves Karissa Wagener Florene Cianci Allene Talton Justine Hoke Lean Broadhurst Sana Walker Apolonia Kobel Jinny Stadler Hortense Parkey Blair Bonetti Rosendo Eversole Wanita Strickler Rosann Frederickson Sofia Alva Lavada McNinch Lizzie Galle Darla Palencia Katie Mccusker Kyle Flenniken Delpha Johnson	Joni Kellog	Joni Kellog	✓

	Input	Expected	Got	
	Louisa Fell Melynda Landrum			
✓	5 3 Theressa Kemmer Kyle Flenniken Louisa Fell Blair Bonetti Robbin Tung	Theressa Kemmer	Theressa Kemmer	✓
✓	15 5 Lean Broadhurst Micki Whyte Jana Granberry Rosann Frederickson Jenine McGavock Katie McCusker Darla Palencia Kimberly Pittenger Robbin Tung Lizzie Galle Delpha Johnson Verlie Schier Joni Kellogg Shanti Newhart Theressa Kemmer	Darla Palencia	Darla Palencia	✓
✓	13 7 Kimberly Pittenger Cindy Avilla Micki Whyte Latosha Boor Sarah Tumlin Louisa Fell Herlinda Swisher Candis Felter Leana Monaghan Armandina Seagraves Shanti Newhart Verlie Schier Florene Cianci	Florene Cianci	Florene Cianci	✓

	Input	Expected	Got	
✓	13 8 Florene Cianci Myrtle Gunter Louisa Fell Carman Capshaw Shanti Newhart Kyle Flenniken Lean Broadhurst Millicent Reppert Marion Batdorf Cindy Avilla Armandina Seagraves Sarah Tumlin Jana Granberry	Armandina Seagraves	Armandina Seagraves	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

[Back to Course](#)