

Question 1

Not answered

Mark 0.00 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	
---	--

## Question 2

Not answered

Mark 0.00 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 5 6 2 48	100 2 6 5 48

**Answer:** (penalty regime: 0 %)

1	
---	--

Question 3

Not answered

Mark 0.00 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 0 1	true

Answer: (penalty regime: 0 %)

1

**Question 4**

Not answered

Mark 0.00 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  $b, \dots$ ).

**For example:**

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

**Answer:** (penalty regime: 0 %)

1	
---	--

**Question 5**

Not answered

Mark 0.00 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 sĩ 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	
---	--

Question 6

Not answered

Mark 0.00 out of 10.00

[PointDescending]

Cuối kỳ, điểm tổng kết môn Toán của khối lớp 8 gồm  $n$  học sinh được lưu trong một mảng 1 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

**Question 7**

Not answered

Mark 0.00 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	
---	--

**Question 8**

Not answered

Mark 0.00 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	
---	--



**Question 9**

Not answered

Mark 0.00 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	
---	--

Question 10

Not answered

Mark 0.00 out of 10.00

[TheLastSurvivor]

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

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

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  $k$  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  $n$  và vòng đếm  $k$ , 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

[Back to Course](#)