

Câu hỏi 1

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu

Viết hàm `int maxThree(int a, int b, int c)` nhận tính số nguyên lớn nhất trong ba số *a*, *b*, *c*.

Input

- Đầu vào của hàm là ba số nguyên

Output

- Đầu ra của hàm là số nguyên lớn nhất trong ba số nguyên đầu vào

Gợi ý

1. Bắt đầu viết hàm với việc chọn tên cho hàm. Tên của hàm phải đại diện cho chức năng của hàm đó, có thể dùng dấu gạch dưới `_` để phân cách các từ hoặc sử dụng ký tự in hoa để phân cách (cách viết kiểu "lạc đà" - camelCase).
2. Sau đó viết danh sách các tham số, mỗi tham số là một biến có kiểu và phân cách với nhau bởi dấu phẩy. Danh sách tham số đặt trong cặp ngoặc tròn `()`
3. Tiếp theo chọn kiểu trả về của hàm và viết phía trước tên hàm. Ví dụ: hàm `maxThree` trả về số nguyên nên kiểu trả về là `int`.
4. Viết phần thân hàm trong cặp ngoặc nhọn `{}`. Trong phần thân hàm ta viết lệnh tương tự như lệnh trong hàm `main()`, có thể khai báo thêm biến mới phục vụ việc tính toán.
5. Dùng lệnh `return` để trả kết quả sau khi tính toán xong.
6. Sử dụng hàm bằng cách gọi hàm và cung cấp danh sách tham số. Ví dụ: gọi `maxThree(3, 2, 1)` ta được kết quả là một số nguyên có giá trị bằng 3.

Lưu ý

1. Chỉ cần viết hàm như đề bài yêu cầu, không cần viết hàm `main()`, không cần viết các câu lệnh `#include`, `using namespace std ...`
2. Cách truyền tham số như trên gọi là truyền tham số bằng giá trị, nghĩa là các biến tham số đầu vào *a*, *b*, *c* của hàm có giá trị giống với giá trị được truyền vào từ bên ngoài (chứ *a*, *b*, *c* không phải là biến là biến bên ngoài).

For example:

Test	Input	Result
<pre>int a, b, c; cin >> a >> b >> c; cout << maxThree(a, b, c);</pre>	4 20 1	20

Answer: (penalty regime: 0 %)

```
1 | int maxThree(int a, int b, int c) {
2 |     int m = a;
3 |     if (m < b) {
4 |         m = b;
5 |     }
6 |     if (m < c) {
7 |         m = c;
8 |     }
9 |     return m;
10| }
```

	Test	Input	Expected	Got	
✓	<pre>int a, b, c; cin >> a >> b >> c; cout << maxThree(a, b, c);</pre>	4 20 1	20	20	✓
✓	<pre>int a, b, c; cin >> a >> b >> c; cout << maxThree(a, b, c);</pre>	21 18 5	21	21	✓
✓	<pre>int a, b, c; cin >> a >> b >> c; cout << maxThree(a, b, c);</pre>	23 87 36	87	87	✓
✓	<pre>int a, b, c; cin >> a >> b >> c; cout << maxThree(a, b, c);</pre>	41 -93 -37	41	41	✓
✓	<pre>int a, b, c; cin >> a >> b >> c; cout << maxThree(a, b, c);</pre>	38475 45862 78504	78504	78504	✓

Passed all tests! ✓

Đúng

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

Câu hỏi 2

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu

Viết hàm `void swap(int& a, int& b)` trao đổi hai biến số nguyên a, b .

Input

- Đầu vào của hàm là hai tham số đến biến kiểu nguyên

Output

- Hàm trao đổi giá trị hai biến số nguyên được đại diện bởi hai tham chiếu đầu vào

Gợi ý

1. Tham chiếu là cách C++ tạo ra đại diện của một biến khác. Do đó khi thay đổi giá trị do tham chiếu đại diện, giá trị của biến được đại diện cũng thay đổi theo. Truyền tham số bằng tham chiếu là phương pháp truyền tham số hay được dùng vì nó không sao chép dữ liệu như truyền tham số bằng giá trị.
2. Trong hàm swap, ta sử dụng cách làm quen thuộc để trao đổi giá trị

```
int tmp = a;
a = b;
b = tmp;
```

Lưu ý

1. Chỉ cần viết hàm như đề bài yêu cầu, không cần viết hàm `main()`, không cần viết các câu lệnh `#include, using namespace std ...`
2. Khi dùng tham chiếu, cần hết sức cẩn thận vì việc thay đổi giá trị bên trong hàm sẽ làm thay đổi giá trị biến bên ngoài.
3. Nếu không cần thay đổi biến bên ngoài mà lại muốn dùng tham chiếu, có thể thêm từ khóa `const` vào trước khai báo tham chiếu.

For example:

Test	Input	Result
<pre>int a, b; cin >> a >> b; swap(a, b); cout << a << " " << b;</pre>	4 20	20 4

Answer: (penalty regime: 0 %)

```
1 void swap(int& a, int& b) {
2     int t = a;
3     a = b;
4     b = t;
5 }
```

	Test	Input	Expected	Got	
✓	<pre>int a, b; cin >> a >> b; swap(a, b); cout << a << " " << b;</pre>	4 20	20 4	20 4	✓
✓	<pre>int a, b; cin >> a >> b; swap(a, b); cout << a << " " << b;</pre>	21 18	18 21	18 21	✓
✓	<pre>int a, b; cin >> a >> b; swap(a, b); cout << a << " " << b;</pre>	23 87	87 23	87 23	✓
✓	<pre>int a, b; cin >> a >> b; swap(a, b); cout << a << " " << b;</pre>	41 -93	-93 41	-93 41	✓
✓	<pre>int a, b; cin >> a >> b; swap(a, b); cout << a << " " << b;</pre>	38475 45862	45862 38475	45862 38475	✓

Passed all tests! ✓

Đúng

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

Câu hỏi 3

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu

Viết hàm `void printArrow(int n, bool left)` để in một hình mũi tên như ví dụ dưới đây, trong đó `n` là bậc của mũi tên, và `left` quy định hướng của mũi tên. Nếu `left` là `true` thì hướng của mũi tên sẽ quay sang trái, là `false` thì sẽ quay sang phải.

Input

- Tham số `n` chỉ kích thước của cánh mũi tên trên cùng và dưới cùng
- Tham số `left` chỉ hướng của mũi tên sang trái (true) hay sang phải (false)

Output

- Hàm chỉ làm nhiệm vụ in ra màn hình, không trả về giá trị gì

Gợi ý

- Do hàm không trả về giá trị gì, ta dùng kiểu void để khai báo với chương trình dịch
- Khi đó chỉ cần lệnh `return;` là có thể thoát khỏi hàm (không cần trả về giá trị nào)
- Trong phần thân hàm, thực hiện vòng lặp để in lần lượt các dòng theo yêu cầu
- Cần tính toán số lượng khoảng trắng trước dấu `*` đầu tiên trên mỗi dòng tùy thuộc vào hướng của mũi tên
- In số khoảng trắng cần thiết rồi in số dấu `*` cần thiết.

Lưu ý

- Trong bài này ta vẫn dùng cách truyền tham số bằng giá trị

For example:

Test	Input	Result
<pre>int n, left; cin >> n >> left; printArrow(n, left);</pre>	5 0	<pre>***** **** *** ** * ** *** **** *****</pre>
<pre>int n, left; cin >> n >> left; printArrow(n, left);</pre>	5 1	<pre>***** **** *** ** * ** *** **** *****</pre>

Answer: (penalty regime: 0 %)

```
1 #include <iostream>
2 using namespace std;
3
4 void printArrow(int n, bool left) {
5     if (left) {
6         int space = n - 1;
7         for (int ast = n; ast >= 1; --ast, --space) {
8             cout << string(space, ' ') << string(ast, '*') << "\n";
9         }
10    }
```

```

8      cout << string(space, ' ') << string(ast, '*') << '\n';
9  }
10  space += 2;
11  for (int ast = 2; ast <= n; ++ast, ++space) {
12      cout << string(space, ' ') << string(ast, '*') << '\n';
13  }
14
15  } else {
16      int space = 0;
17      for (int ast = n; ast >= 1; --ast, space += 2) {
18          cout << string(space, ' ') << string(ast, '*') << '\n';
19      }
20      space -= 4;
21      for (int ast = 2; ast <= n; ++ast, space -= 2) {
22          cout << string(space, ' ') << string(ast, '*') << '\n';
23      }
24  }
25  }

```

	Test	Input	Expected	Got	
✓	<pre>int n, left; cin >> n >> left; printArrow(n, left);</pre>	5 0	<pre>***** **** *** ** * ** *** **** *****</pre>	<pre>***** **** *** ** * ** *** **** *****</pre>	✓
✓	<pre>int n, left; cin >> n >> left; printArrow(n, left);</pre>	5 1	<pre>***** **** *** ** * ** *** **** *****</pre>	<pre>***** **** *** ** * ** *** **** *****</pre>	✓

[illegible]

Câu hỏi 4

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết hàm `long long int factorial(int n)` nhận đầu vào là số nguyên dương n , đầu ra là giá trị $n! = 1 \times 2 \times 3 \times \dots \times n$
hàm trả về long long nhưng mà trả về int vẫn đúng 1 số testcase

Input:

- Số nguyên dương n

Output:

- Kết quả giai thừa của n

Gợi ý:

- Khai báo biến lưu giá trị giai thừa, có giá trị là 1: `long long int result = 1;`
- Để tính giai thừa dùng vòng for biến chạy i từ 2 đến n : `result = result * i;`
- Trả về kết quả

Lưu ý:

- Cần chú ý khai báo biến, tránh tràn số
- Kiểu trả về của hàm là `long long int`
- Biến lưu kết quả là `long long int`

Answer: (penalty regime: 0 %)

```
1 long long factorial(int n) {  
2     long long ret = 1;  
3     for (int i = 2; i <= n; ++i) {  
4         ret *= i;  
5     }  
6     return ret;  
7 }
```

	Test	Input	Expected	Got	
✓	factorial(n)	4	24	24	✓
✓	factorial(n)	10	3628800	3628800	✓
✓	factorial(n)	14	87178291200	87178291200	✓
✓	factorial(n)	0	1	1	✓
✓	factorial(n)	20	2432902008176640000	2432902008176640000	✓

Passed all tests! ✓

Đúng

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

Câu hỏi 5

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết hàm `bool isPrime (int number)` kiểm tra xem số nguyên *number* có phải là một số nguyên tố không.

Nếu đúng, trả về *true*, ngược lại, trả về *false*.

Input:

- Số nguyên *number*

Output:

- Trả về *true* nếu *number* là số nguyên tố, ngược lại, trả về *false*.

Gợi ý:

- Kiểm tra số nguyên đầu vào có phải âm, 0 hoặc 1 không. Trường hợp này trả về *false*.
- Khởi tạo biến `bool checkPrime = true`; để gán cho biến kiểm tra ban đầu là đúng.
- Dùng vòng `for` duyệt *i* từ 2 đến căn bậc 2 của *number*, nếu *number* chia hết cho *i* thì *number* không phải số nguyên tố, chuyển biến `checkPrime = false`; dùng lệnh `break`; để kết thúc vòng `for` sớm. (Nếu kết thúc vòng `for` *number* không chia hết cho bất cứ số nào trong khoảng 2 đến căn bậc 2 của nó, thì *number* là số nguyên tố và `checkPrime` vẫn giữ giá trị *True*)
- Trả về biến `checkPrime`.

For example:

Test	Input	Result
<pre>int number; cin >> number; if(isPrime(number)){ cout << "Prime."; } else { cout << "Not a prime."; }</pre>	3	Prime.

Answer: (penalty regime: 0 %)

```
1 bool isPrime(int n) {
2     for (int i = 2; i*i <= n; ++i) {
3         if (n % i == 0) {
4             return false;
5         }
6     }
7     return n > 1;
8 }
```

	Test	Input	Expected	Got	
✓	<pre> int number; cin >> number; if(isPrime(number)){ cout << "Prime."; } else { cout << "Not a prime."; } </pre>	3	Prime.	Prime.	✓
✓	<pre> int number; cin >> number; if(isPrime(number)){ cout << "Prime."; } else { cout << "Not a prime."; } </pre>	12	Not a prime.	Not a prime.	✓
✓	<pre> int number; cin >> number; if(isPrime(number)){ cout << "Prime."; } else { cout << "Not a prime."; } </pre>	-12	Not a prime.	Not a prime.	✓
✓	<pre> int number; cin >> number; if(isPrime(number)){ cout << "Prime."; } else { cout << "Not a prime."; } </pre>	83	Prime.	Prime.	✓
✓	<pre> int number; cin >> number; if(isPrime(number)){ cout << "Prime."; } else { cout << "Not a prime."; } </pre>	0	Not a prime.	Not a prime.	✓

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:

Viết hàm `double calculate (double num1, char operat, double num2)` giả lập một máy tính bỏ túi.

Input:

- Tham số `num1` và `num2` tương ứng với 2 số hạng của phép toán
- Tham số `operat` tương ứng với toán tử. Biết các toán tử đơn giản thực hiện phép toán lần lượt là:
 - Phép trừ `-`
 - Phép cộng `+`
 - Phép chia `/`
 - Phép nhân `*`

Output:

- Kết quả của phép toán giữa số `num1` và `num2` với toán tử `operat`
- Kết quả làm tròn đến chữ số thập phân thứ hai.

Gợi ý:

- Dùng cấu trúc `if-else` hoặc `switch` để xét từng trường hợp của `operat`
- Với mỗi trường hợp, sử dụng phép toán tương ứng với `operat` để tính với 2 số hạng là `num1` và `num2` sau đó `return` luôn kết quả.

Lưu ý:

- Trong bài này ta vẫn dùng cách truyền tham số bằng giá trị

For example:

Test	Input	Result
<code>calculate (num1, operat, num2)</code>	<code>2 + 3</code>	<code>5.00</code>

Answer: (penalty regime: 0 %)

```
1 double calculate(double num1, char operat, double num2) {
2     if (operat == '+') {
3         return num1 + num2;
4     } if (operat == '-') {
5         return num1 - num2;
6     } if (operat == '*') {
7         return num1 * num2;
8     } if (operat == '/') {
9         return num1 / num2;
10    }
11    return 0;
12 }
```

	Test	Input	Expected	Got	
✓	calculate (num1, operat, num2)	2 + 3	5.00	5.00	✓
✓	calculate (num1, operat, num2)	12 / 4	3.00	3.00	✓
✓	calculate (num1, operat, num2)	23.89 * 23.12	552.34	552.34	✓
✓	calculate (num1, operat, num2)	82 - 12	70.00	70.00	✓
✓	calculate (num1, operat, num2)	12 * -23.004	-276.05	-276.05	✓

Passed all tests! ✓

Đúng

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

Câu hỏi 7

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết hàm `void readArray(int a[], int n)` để đọc vào n số nguyên từ bàn phím và lưu vào mảng số nguyên a và hàm `void printArray(int a[], int n)` để in các phần tử của mảng a có số lượng phần tử là n , các phần tử cách nhau bởi dấu cách.

Input:

- Một dãy số nguyên n phần tử.

Output:

- In ra dãy số nguyên đã nhập từ bàn phím

Gợi ý:

- Bạn chỉ cần viết các hàm được yêu cầu, hàm `main` để đọc dữ liệu và chạy chương trình đã có sẵn.
- Trong 2 hàm này bạn không cần phải return lại arr, khi mảng được truyền vào hàm, thay đổi các giá trị của mảng trong hàm thì khi ra ngoài hàm, các giá trị trong mảng sẽ bị thay đổi theo (đến bài con trở sẽ hiểu)

For example:

Test	Input	Result
<pre>int n = 5; int a[n]; readArray(a, n); printArray(a, n);</pre>	1 2 3 4 5	1 2 3 4 5

Answer: (penalty regime: 0 %)

```
1 void readArray(int a[], int n) {
2     for (int i = 0; i < n; ++i) {
3         cin >> a[i];
4     }
5 }
6
7 void printArray(int a[], int n) {
8     for (int i = 0; i < n; ++i) {
9         cout << a[i] << ' ';
10    }
11 }
```

	Test	Input	Expected	Got	
✓	<pre>int n = 5; int a[n]; readArray(a, n); printArray(a, n);</pre>	1 2 3 4 5	1 2 3 4 5	1 2 3 4 5	✓
✓	<pre>int n = 10; int a[n]; readArray(a, n); printArray(a, n);</pre>	-1 -2 -3 -4 -5 -6 -7 -8 -9 -10	-1 -2 -3 -4 -5 -6 -7 -8 -9 -10	-1 -2 -3 -4 -5 -6 -7 -8 -9 -10	✓
✓	<pre>int n = 50; int a[n]; readArray(a, n); printArray(a, n);</pre>	-945 740 -865 918 -766 613 449 148 367 287 -502 -43 -84 657 617 179 -763 537 801 -562 279 -459 851 -28 -986 -808 -255 360 -44 702 17 -519 -90 -910 -838 300 -390 757 -287 731 -129 -958 -955 512 -931 201 213 995 65 62	-945 740 -865 918 -766 613 449 148 367 287 -502 -43 -84 657 617 179 -763 537 801 -562 279 -459 851 -28 -986 -808 -255 360 -44 702 17 -519 -90 -910 -838 300 -390 757 -287 731 -129 -958 -955 512 -931 201 213 995 65 62	-945 740 -865 918 -766 613 449 148 367 287 -502 -43 -84 657 617 179 -763 537 801 -562 279 -459 851 -28 -986 -808 -255 360 -44 702 17 -519 -90 -910 -838 300 -390 757 -287 731 -129 -958 -955 512 -931 201 213 995 65 62	✓
✓	<pre>int n = 100; int a[n]; readArray(a, n); printArray(a, n);</pre>	948 530 492 -484 -692 -552 661 153 -838 -978 559 -626 -983 66 594 -408 497 -11 -587 -877 -54 -280 -683 554 -819 -306 -328 245 -510 -390 585 -40 787 -712 809 -888 -971 -70 832 -42 156 -136 -119 -717 -694 151 -22 555 530 -456 422 961 115 -411 -57 -273 597 911 -864 -157 -765 315 642 939 -494 -410 -253 -664 781 -393 -469 -205 713 984 724 997 -95 954 809 -468 -298 -591 -532 609 311 -597 198 -718 436 183 -721 608 638 -576 455 13 867 -477 127 -983	948 530 492 -484 -692 -552 661 153 -838 -978 559 -626 -983 66 594 -408 497 -11 -587 -877 -54 -280 -683 554 -819 -306 -328 245 -510 -390 585 -40 787 -712 809 -888 -971 -70 832 -42 156 -136 -119 -717 -694 151 -22 555 530 -456 422 961 115 -411 -57 -273 597 911 -864 -157 -765 315 642 939 -494 -410 -253 -664 781 -393 -469 -205 713 984 724 997 -95 954 809 -468 -298 -591 -532 609 311 -597 198 -718 436 183 -721 608 638 -576 455 13 867 -477 127 -983	948 530 492 -484 -692 -552 661 153 -838 -978 559 -626 -983 66 594 -408 497 -11 -587 -877 -54 -280 -683 554 -819 -306 -328 245 -510 -390 585 -40 787 -712 809 -888 -971 -70 832 -42 156 -136 -119 -717 -694 151 -22 555 530 -456 422 961 115 -411 -57 -273 597 911 -864 -157 -765 315 642 939 -494 -410 -253 -664 781 -393 -469 -205 713 984 724 997 -95 954 809 -468 -298 -591 -532 609 311 -597 198 -718 436 183 -721 608 638 -576 455 13 867 -477 127 -983	✓

	Test	Input	Expected	Got	
✓	<pre>int n = 1000; int a[n]; readArray(a, n); printArray(a, n);</pre>	<pre>-309 390 423 541 26 318 -733 -649 631 774 -285 -850 -837 -68 -967 -56 510 -76 168 -822 351 74 -459 206 872 368 746 -745 554 462 531 -820 -297 -709 -217 732 -190 -136 -797 -561 497 -738 -697 195 916 58 -853 401 -440 219 608 229 -242 -185 141 -931 660 -598 -462 317 -721 -42 -750 -84 25 -53 835 727 -555 -626 681 -133 -697 -562 118 -443 -620 -268 720 -346 -180 785 376 618 557 -794 -18 961 814 122 644 -456 830 785 899 -843 -981 721 351 -700 343 923 644 77 -494 -862 467 -530 431 -762 -624 -529 -60 332 -349 -665 -865 183 734 -555 111 -672 -272 785 -143 -485 -146 -873 949 -628 -37 -640 996 -993 97 928 -712 37 -382 299 247 -871 898 -54 568 -377 -385 596 683 900 856 -743 22 101 -512 255 -388 779 465 -106 79 876 -911 -470 532 109 636 -423 870 573 698 -588 -284 -912 -745 -553 176 108 -622 -307 -407 -428 21 -84 -853 -195 -675 82 -154 750 -465 -731 75 -53 -629 724 482 854 501 556 725 -26 -605 122 -746 -822 535 13 -914 -597 -215 437 -224 -889 -979 -686 -345 7 902 -340 -971 256 -581 -972 -204 -999 267 -750 -757 -187 -847 754 410 -170 836 -435 -149 -178 291 45 488 38 308 -449 777 -240 236 -241 108 167 100 -981 -242 137 56 44 -811 -50 -220 -108 -438 -724 237 -252 -50 450 896 -877 -339 -25 504 -117 652 336 674 -342 -264 244 -942 656 -494 -87 333 776 -665 -213 -519 -643 -725 -650 453 258 631 225 163 781 -509 302 453 484 613 -801 -323 -501 357 880 -801 320 -936 280 355 466 -874 614 59 364 803 -916 909 647 481 319 508 868 -577 -43 116 31 715 -763 -986 -775 -148 -995 55 603 25 -896 -484 779 62 -363 377 378 335 -596 10 120 -681 224 991 -205 161 735 444 280 134 948 37 -677 -43 -57 -87 455 612 -320 161 -485</pre>	<pre>-309 390 423 541 26 318 -733 -649 631 774 -285 -850 -837 -68 -967 -56 510 -76 168 -822 351 74 -459 206 872 368 746 -745 554 462 531 -820 -297 -709 -217 732 -190 -136 -797 -561 497 -738 -697 195 916 58 -853 401 -440 219 608 229 -242 -185 141 -931 660 -598 -462 317 -721 -42 -750 -84 25 -53 835 727 -555 -626 681 -133 -697 -562 118 -443 -620 -268 720 -346 -180 785 376 618 557 -794 -18 961 814 122 644 -456 830 785 899 -843 -981 721 351 -700 343 923 644 77 -494 -862 467 -530 431 -762 -624 -529 -60 332 -349 -665 -865 183 734 -555 111 -672 -272 785 -143 -485 -871 898 -54 568 -377 -385 596 683 900 856 -743 22 101 -512 255 -388 779 465 -106 79 876 -911 -470 532 109 636 -423 870 573 698 -588 -284 -912 -745 -553 176 108 -622 -307 -407 -428 21 -84 -853 -195 -675 82 -154 750 -465 -731 75 -53 -629 724 482 854 501 556 725 -26 -605 122 -746 -822 535 13 -914 -597 -215 437 -224 -889 -979 -686 -345 7 902 -340 -971 256 -581 -972 -204 -999 267 -750 -757 -187 -847 754 410 -170 836 -435 -149 -178 291 45 488 38 308 -449 777 -240 236 -241 108 167 100 -981 -242 137 56 44 -811 -50 -220 -108 -438 -724 237 -252 -50 450 896 -877 -339 -25 504 -117 652 336 674 -342 -264 244 -942 656 -494 -87 333 776 -665 -213 -519 -643 -725 -650 453 258 631 225 163 781 -509 302 453 484 613 -801 -323 -501 357 880 -801 320 -936 280 355 466 -874 614 59 364 803 -916 909 647 481 319 508 868 -577 -43 116 31 715 -763 -986 -775 -148 -995 55 603 25 -896 -484 779 62 -363 377 378 335 -596 10 120 -681 224 991 -205 161 735 444 280 134 948 37 -677 -43 -57 -87 455 612 -320 161 -485 38 89 744 -467 475 -127 819 -406 -964 -420 -462 96 -252 777 -368 -565 634 -926 505 918 -484 -154 309 596 -894 235 -115 -745 -601 -570 560 -376 74 817 -150 -983 -196 983 -582</pre>	<pre>-309 390 423 541 26 318 -733 -649 631 774 -285 -850 -837 -68 -967 -56 510 -76 168 -822 351 74 -459 206 872 368 746 -745 554 462 531 -820 -297 -709 -217 732 -190 -136 -797 -561 497 -738 -697 195 916 58 -853 401 -440 219 608 229 -242 -185 141 -931 660 -598 -462 317 -721 -42 -750 -84 25 -53 835 727 -555 -626 681 -133 -697 -562 118 -443 -620 -268 720 -346 -180 785 376 618 557 -794 -18 961 814 122 644 -456 830 785 899 -843 -981 721 351 -700 343 923 644 77 -494 -862 467 -530 431 -762 -624 -529 -60 332 -349 -665 -865 183 734 -555 111 -672 -272 785 -143 -485 -146 -873 949 -628 -37 -640 996 -993 97 928 -712 37 -382 299 247 -871 898 -54 568 -377 -385 596 683 900 856 -743 22 101 -512 255 -388 779 465 -106 79 876 -911 -470 532 109 636 -423 870 573 698 -588 -284 -912 -745 -553 176 108 -622 -307 -407 -428 21 -84 -853 -195 -675 82 -154 750 -465 -731 75 -53 -629 724 482 854 501 556 725 -26 -605 122 -746 -822 535 13 -914 -597 -215 437 -224 -889 -979 -686 -345 7 902 -340 -971 256 -581 -972 -204 -999 267 -750 -757 -187 -847 754 410 -170 836 -435 -149 -178 291 45 488 38 308 -449 777 -240 236 -241 108 167 100 -981 -242 137 56 44 -811 -50 -220 -108 -438 -724 237 -252 -50 450 896 -877 -339 -25 504 -117 652 336 674 -342 -264 244 -942 656 -494 -87 333 776 -665 -213 -519 -643 -725 -650 453 258 631 225 163 781 -509 302 453 484 613 -801 -323 -501 357 880 -801 320 -936 280 355 466 -874 614 59 364 803 -916 909 647 481 319 508 868 -577 -43 116 31 715 -763 -986 -775 -148 -995 55 603 25 -896 -484 779 62 -363 377 378 335 -596 10 120 -681 224 991 -205 161</pre>	✓

Test	Input	Expected	Got
	38 89 744 -467 475 -127 819	-876 -808 -655 -142 214 744 149	735 444 280 134 948 37 -677
	-406 -964 -420 -462 96 -252	357 937 -26 984 407 -135 307 -657	-43 -57 -87 455 612 -320 161
	777 -368 -565 634 -926 505 918	56 675 -417 410 -595 -273 -443	-485 38 89 744 -467 475 -127
	-484 -154 309 596 -894 235	-505 -863 955 77 -188 21 -335 230	819 -406 -964 -420 -462 96
	-115 -745 -601 -570 560 -376	966 -254 -689 -94 -784 -960 562	-252 777 -368 -565 634 -926
	74 817 -150 -983 -196 983 -582	799 452 -666 -616 478 710 644	505 918 -484 -154 309 596
	-876 -808 -655 -142 214 744	-608 -710 -952 -469 40 -593 -908	-894 235 -115 -745 -601 -570
	149 357 937 -26 984 407 -135	111 -164 127 -462 53 564 421 40	560 -376 74 817 -150 -983
	307 -657 56 675 -417 410 -595	543 -284 -543 -376 480 -137 -319	-196 983 -582 -876 -808 -655
	-273 -443 -505 -863 955 77	-748 845 -724 898 761 690 997 586	-142 214 744 149 357 937 -26
	-188 21 -335 230 966 -254 -689	-819 -16 -573 799 -812 12 -832 47	984 407 -135 307 -657 56 675
	-94 -784 -960 562 799 452 -666	837 545 356 -672 759 629 653 -513	-417 410 -595 -273 -443 -505
	-616 478 710 644 -608 -710	-128 912 -165 56 -923 845 259 555	-863 955 77 -188 21 -335 230
	-952 -469 40 -593 -908 111	-958 -148 -572 -560 294 -619 687	966 -254 -689 -94 -784 -960
	-164 127 -462 53 564 421 40	-515 963 -90 -264 992 881 870	562 799 452 -666 -616 478 710
	543 -284 -543 -376 480 -137	-775 -497 -645 -724 401 25 828	644 -608 -710 -952 -469 40
	-319 -748 845 -724 898 761 690	653 176 247 -311 -912 469 26 973	-593 -908 111 -164 127 -462
	997 586 -819 -16 -573 799 -812	89 -513 939 636 602 279 383 -473	53 564 421 40 543 -284 -543
	12 -832 47 837 545 356 -672	-293 717 659 -79 -204 399 811 834	-376 480 -137 -319 -748 845
	759 629 653 -513 -128 912 -165	720 -117 189 -459 30 -327 -659	-724 898 761 690 997 586 -819
	56 -923 845 259 555 -958 -148	-827 523 -288 -235 580 290 156	-16 -573 799 -812 12 -832 47
	-572 -560 294 -619 687 -515	-476 671 -550 936 82 502 -947 386	837 545 356 -672 759 629 653
	963 -90 -264 992 881 870 -775	-378 343 -436 697 66 -736 -272	-513 -128 912 -165 56 -923
	-497 -645 -724 401 25 828 653	-865 14 -226 700 709 -362 -467	845 259 555 -958 -148 -572
	176 247 -311 -912 469 26 973	-861 -530 27 -751 997 -632 510 62	-560 294 -619 687 -515 963
	89 -513 939 636 602 279 383	472 -864 -569 -169 745 -754 -786	-90 -264 992 881 870 -775
	-473 -293 717 659 -79 -204 399	391 251 -462 -126 -292 -337 -385	-497 -645 -724 401 25 828 653
	811 834 720 -117 189 -459 30	334 -250 163 834 -479 678 944 605	176 247 -311 -912 469 26 973
	-327 -659 -827 523 -288 -235	740 932 895 774 -779 -868 926	89 -513 939 636 602 279 383
	580 290 156 -476 671 -550 936	-760 -153 226 497 -112 -660 232	-473 -293 717 659 -79 -204
	82 502 -947 386 -378 343 -436	-802 279 42 173 -78 -681 -317	399 811 834 720 -117 189 -459
	697 66 -736 -272 -865 14 -226	-892 -1000 588 42 989 -106 25	30 -327 -659 -827 523 -288
	700 709 -362 -467 -861 -530 27	-835 -685 -665 -850 311 -248 -815	-235 580 290 156 -476 671
	-751 997 -632 510 62 472 -864	-319 -500 194 -184 -634 -621 508	-550 936 82 502 -947 386 -378
	-569 -169 745 -754 -786 391	319 -875 -484 -535 495 -82 -699 2	343 -436 697 66 -736 -272
	251 -462 -126 -292 -337 -385	-998 -397 71 -416 -426 -67 -218	-865 14 -226 700 709 -362
	334 -250 163 834 -479 678 944	-196 221 963 472 329 -302 166 901	-467 -861 -530 27 -751 997
	605 740 932 895 774 -779 -868	-677 -254 28 196 -787 156 -361 19	-632 510 62 472 -864 -569
	926 -760 -153 226 497 -112	-515 398 20 -312 810 -123 849 345	-169 745 -754 -786 391 251
	-660 232 -802 279 42 173 -78	813 -933 272 490 -154 -505 742	-462 -126 -292 -337 -385 334
	-681 -317 -892 -1000 588 42	913 -921 -467 -539 291 -190 336	-250 163 834 -479 678 944 605
	989 -106 25 -835 -685 -665	43 -597 -581 828 570 386 -542 217	740 932 895 774 -779 -868 926
	-850 311 -248 -815 -319 -500	-905 946 196 261 -565 101 -928	-760 -153 226 497 -112 -660
	194 -184 -634 -621 508 319	995 -241 529 249 436 111 546 -275	232 -802 279 42 173 -78 -681
	-875 -484 -535 495 -82 -699 2	514 288 -68 -298 -133 -778 701	-317 -892 -1000 588 42 989
	-998 -397 71 -416 -426 -67	-230 47 -942 -770 396 764 -61 749	-106 25 -835 -685 -665 -850
	-218 -196 221 963 472 329 -302	116 -909 -242 962 -591 406 962	311 -248 -815 -319 -500 194
	166 901 -677 -254 28 196 -787	-582 -819 766 207 -95 -225 -667	-184 -634 -621 508 319 -875
	156 -361 19 -515 398 20 -312	-406 722 989 828 944 498 -367 931	-484 -535 495 -82 -699 2 -998
	810 -123 849 345 813 -933 272	347 -600 -594 -819 780 756 770	-397 71 -416 -426 -67 -218
	490 -154 -505 742 913 -921	-638 -506 -163 -587 -73 -660 380	-196 221 963 472 329 -302 166
	-467 -539 291 -190 336 43 -597	-421 -438 -302 -532 619 22 57	901 -677 -254 28 196 -787 156
	-581 828 570 386 -542 217 -905	-258 917 -479 929 954 938 -293	-361 19 -515 398 20 -312 810
	946 196 261 -565 101 -928 995	-684 1 -638 -431 -316 0 945 -697	-123 849 345 813 -933 272 490
	-241 529 249 436 111 546 -275	-434 -748 -584 -738 594 361 32	-154 -505 742 913 -921 -467

	Test	Input	Expected	Got	
		514 288 -68 -298 -133 -778 701 -230 47 -942 -770 396 764 -61 749 116 -909 -242 962 -591 406 962 -582 -819 766 207 -95 -225 -667 -406 722 989 828 944 498 -367 931 347 -600 -594 -819 780 756 770 -638 -506 -163 -587 -73 -660 380 -421 -438 -302 -532 619 22 57 -258 917 -479 929 954 938 -293 -684 1 -638 -431 -316 0 945 -697 -434 -748 -584 -738 594 361 32 -393 -84 24 72 -245 593 -118 -654 -221 -909 601 -509 -520 -107 -390 604 -109 -39 132 -404 -326 332 -733 831 -246 380 660 -345 -662 -517 -226 154 -735 638 -324 354 928 -172 18 486 729 -983 -541 -272 -46 -251 208 -989 353 235 421 53 -952 -613 -44 914 830 639 833 713 -711 -564 519 -776 486 780 600 -145 773 994 782 574 428 -198 874 78 -103 -871 -594 -460 867 849 -276 766 -778 76 -958 453 -162 -240 994 -511 69 -863 827 -340 -410 -353 154 260 -255 914 -375 -617 -229 979 415 821 301 -318 446 -788 -335 -372 697 711 -725 749 137 387 699 883 -395 56 180 -265 -518 -965 -502 -85 668 763 -199 -669 -712 132 -145 -29 585 135 562 -816 -198 -753 -225 289 -800 -580 -301 52 347 305 162 -815 623 875 498 815 -865 720 -470 311 574 -481 -470 293 -580 -613 719 612 -852 -603 526 -6 -695 14 -820 99 464 -562 -161 -869 -854 835 -103 432	-393 -84 24 72 -245 593 -118 -654 -221 -909 601 -509 -520 -107 -390 604 -109 -39 132 -404 -326 332 -733 831 -246 380 660 -345 -662 -517 -226 154 -735 638 -324 354 928 -172 18 486 729 -983 -541 -272 -46 -251 208 -989 353 235 421 53 -952 -613 -44 914 830 639 833 713 -711 -564 519 -776 486 780 600 -145 773 994 782 574 428 -198 874 78 -103 -871 -594 -460 867 849 -276 766 -778 76 -958 453 -162 -240 994 -511 69 -863 827 -340 -410 -353 154 260 -255 914 -375 -617 -229 979 415 821 301 -318 446 -788 -335 -372 697 711 -725 749 137 387 699 883 -395 56 180 -265 -518 -965 -502 -85 668 763 -199 -669 -712 132 -145 -29 585 135 562 -816 -198 -753 -225 289 -800 -580 -301 52 347 305 162 -815 623 875 498 815 -865 720 -470 311 574 -481 -470 293 -580 -613 719 612 -852 -603 526 -6 -695 14 -820 99 464 -562 -161 -869 -854 835 -103 432	-539 291 -190 336 43 -597 -581 828 570 386 -542 217 -905 946 196 261 -565 101 -928 995 -241 529 249 436 111 546 -275 514 288 -68 -298 -133 -778 701 -230 47 -942 -770 396 764 -61 749 116 -909 -242 962 -591 406 962 -582 -819 766 207 -95 -225 -667 -406 722 989 828 944 498 -367 931 347 -600 -594 -819 780 756 770 -638 -506 -163 -587 -73 -660 380 -421 -438 -302 -532 619 22 57 -258 917 -479 929 954 938 -293 -684 1 -638 -431 -316 0 945 -697 -434 -748 -584 -738 594 361 32 -393 -84 24 72 -245 593 -118 -654 -221 -909 601 -509 -520 -107 -390 604 -109 -39 132 -404 -326 332 -733 831 -246 380 660 -345 -662 -517 -226 154 -735 638 -324 354 928 -172 18 486 729 -983 -541 -272 -46 -251 208 -989 353 235 421 53 -952 -613 -44 914 830 639 833 713 -711 -564 519 -776 486 780 600 -145 773 994 782 574 428 -198 874 78 -103 -871 -594 -460 867 849 -276 766 -778 76 -958 453 -162 -240 994 -511 69 -863 827 -340 -410 -353 154 260 -255 914 -375 -617 -229 979 415 821 301 -318 446 -788 -335 -372 697 711 -725 749 137 387 699 883 -395 56 180 -265 -518 -965 -502 -85 668 763 -199 -669 -712 132 -145 -29 585 135 562 -816 -198 -753 -225 289 -800 -580 -301 52 347 305 162 -815 623 875 498 815 -865 720 -470 311 574 -481 -470 293 -580 -613 719 612 -852 -603 526 -6 -695 14 -820 99 464 -562 -161 -869 -854 835 -103 432	

Passed all tests! ✓

Đúng

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

Câu hỏi 8

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết hàm `void readArray(int array[], int n)` để đọc n số nguyên nhập từ bàn phím vào mảng `array`

Hàm `readArray` được sử dụng để khởi tạo hai mảng `array1` và `array2`

Viết hàm `bool compareArray(int array1[], int array2[], int n)` nhận đầu vào là 2 mảng nguyên có n phần tử trên, đầu ra là `true` nếu 2 mảng giống hệt nhau, `false` nếu khác nhau

Input:

- Một dãy số nguyên n phần tử.

Output:

- In ra `true` nếu 2 mảng giống hệt nhau, ngược lại in ra `false`.

Gợi ý:

- Bạn chỉ cần viết các hàm được yêu cầu, hàm `main` để đọc dữ liệu, tạo 2 mảng và chạy chương trình đã có sẵn.
- Trong 2 hàm này bạn không cần phải return lại arr, khi mảng được truyền vào hàm, thay đổi các giá trị của mảng trong hàm thì khi ra ngoài hàm, các giá trị trong mảng sẽ bị thay đổi theo (đến bài con trở sẽ hiểu)
- Bài này chỉ cần dùng 1 vòng lặp `for` với câu lệnh `if else` để so sánh các vị trí tương ứng ở 2 mảng, chỉ cần 1 vị trí khác nhau là ta sẽ trả về `false`,

For example:

Test	Input	Result
<pre>int n; cin >> n; int array1[n], array2[n]; readArray(array1, n); readArray(array2, n); if(compareArray(array1, array2, n)){ cout << "true"; } else { cout << "false"; }</pre>	<pre>10 -150 190 -866 -840 -972 -649 596 402 -548 -507 -913 -483 568 -988 -455 -324 -370 -953 253 -344</pre>	false

Answer: (penalty regime: 0 %)

<pre>1 void readArray(int a[], int n) { 2 for (int i = 0; i < n; ++i) { 3 cin >> a[i]; 4 } 5 } 6 7 bool compareArray(int a[], int b[], int n) { 8 for (int i = 0; i < n; ++i) { 9 if (a[i] != b[i]) { 10 return false; 11 } 12 } 13 return true; 14 }</pre>	
---	--

	Test	Input	Expected	Got	
✓	<pre> int n; cin >> n; int array1[n], array2[n]; readArray(array1, n); readArray(array2, n); if(compareArray(array1, array2, n)){ cout << "true"; } else { cout << "false"; } </pre>	<pre> 10 -150 190 -866 -840 -972 -649 596 402 -548 -507 -913 -483 568 -988 -455 -324 -370 -953 253 -344 </pre>	false	false	✓
✓	<pre> int n; cin >> n; int array1[n], array2[n]; readArray(array1, n); readArray(array2, n); if(compareArray(array1, array2, n)){ cout << "true"; } else { cout << "false"; } </pre>	<pre> 10 -556 591 44 -238 100 -421 -785 -742 -400 -232 -614 -49 -428 772 911 198 -283 -338 -962 900 </pre>	false	false	✓
✓	<pre> int n; cin >> n; int array1[n], array2[n]; readArray(array1, n); readArray(array2, n); if(compareArray(array1, array2, n)){ cout << "true"; } else { cout << "false"; } </pre>	<pre> 10 431 -666 808 809 843 -130 -502 -511 -129 -876 431 -666 808 809 843 -130 -502 -511 -129 -876 </pre>	true	true	✓

	Test	Input	Expected	Got	
✓	<pre> int n; cin >> n; int array1[n], array2[n]; readArray(array1, n); readArray(array2, n); if(compareArray(array1, array2, n)){ cout << "true"; } else { cout << "false"; } </pre>	<pre> 100 -3759 -8169 3571 -3839 -3176 340 1190 9121 5582 6829 -6319 8484 6516 8609 986 -4636 -5339 -3219 -4975 4127 3428 -9669 2179 8473 -6775 -9863 5472 4502 7527 6437 2482 4061 9045 -3079 583 -883 8513 -9267 145 -2810 170 2257 -4359 9454 -9635 -8237 2766 -8288 6135 -6885 -9550 83 2851 -8343 -9771 8874 8136 -7681 -3722 4963 -8841 448 9045 -5185 3816 -8720 -5840 -7353 -1369 7263 -9351 7916 -5871 9276 -9726 -5048 4700 -4434 8176 4845 -6607 -5582 -7044 660 -5311 1925 -1656 6567 -8730 1556 5184 3087 -2852 8807 -7260 -1229 -344 7730 -5124 5334 7923 4774 8930 9203 -9461 497 -4524 -5570 8675 -8431 -6647 -3914 -8593 2392 4102 9140 -2262 4647 -3330 -7626 -8461 2738 -6635 2814 3795 4074 -9295 5962 -2375 5607 8015 564 -7574 25 2862 -6719 -4209 -6989 -9885 -9069 -8024 852 -3270 -9031 3133 -476 4051 -7400 -1948 2203 3371 -9731 1233 -2684 6533 -6942 6910 9146 8258 -2545 2012 -5903 -5315 -5344 9346 2400 4976 5878 -386 -5586 3230 -9297 -5894 2637 1190 7202 -3144 6523 1166 -2880 9893 6628 -982 8075 -5639 -2843 -5286 4082 -5101 -5999 -7918 -1323 538 -4177 -8599 5612 9851 -6190 -1211 -1847 </pre>	false	false	✓

	Test	Input	Expected	Got	
✓	<pre> int n; cin >> n; int array1[n], array2[n]; readArray(array1, n); readArray(array2, n); if(compareArray(array1, array2, n)){ cout << "true"; } else { cout << "false"; } </pre>	<pre> 250 14246 -10966 764 4571 308 -20148 -11733 -3131 -1236 10528 230 -5417 -1523 -2034 6873 -19923 -6743 -1972 -11362 -15054 -581 -18333 15445 -7942 -13973 21338 10320 4155 -16902 -14886 1222 11313 4495 5490 -5567 773 -9797 9058 22954 439 -8264 -15345 12605 -10010 -14459 22566 5722 -17286 2781 3578 15855 -3745 -16905 20754 9535 18372 -7254 8845 -1180 20304 10022 -9247 -18016 14974 -5007 9618 -7422 18091 -21697 13339 -23873 -13607 -11275 7569 10076 18541 -12596 -18891 8647 15428 -2070 -24459 -7899 8409 5360 -20571 6323 -19529 -12542 -23704 -10773 15924 9974 7224 9138 24374 -2608 6932 -8470 -3687 14870 -7432 -18697 -11686 373 -6426 -3308 -6471 10256 14722 21728 -16418 -22722 11570 -20381 5178 11025 19577 12727 -13214 1055 22410 7379 8309 700 -14496 -14670 -24239 10123 3792 9591 716 -12426 20767 -8078 -19970 -7895 5123 -21723 -21651 -11367 14399 -14645 -15547 17600 24591 3820 -2282 8123 9325 24177 13076 21566 -6912 -16279 10026 -12746 -12135 23912 -1384 -821 9175 8057 13938 -20481 -932 -9536 -7091 -8721 -20612 16126 -1266 37 -21707 19547 8668 -15574 20526 13255 2770 -12305 16952 22568 -4463 4809 8616 10914 -12999 4611 8784 -6604 -7962 12010 12314 -13254 -22956 8922 -22401 -3423 12622 -23737 9642 -830 -24483 4143 7334 21152 -9024 20805 -15733 21340 -24941 15246 2871 24413 5660 -5771 -11708 -12897 -12161 -17708 -23201 -19055 22729 -10344 -14912 4868 17404 12295 8555 11136 -20226 15362 -5676 -23264 -3991 1132 -16238 -21708 2495 -2640 15641 -9282 14847 -10361 -13479 8173 1244 -21741 -806 -3784 -11810 21185 -13489 16619 22252 11133 3914 41 -2208 20683 -9193 15663 -1195 -8995 -3739 20612 22186 -891 -17701 -2088 -21408 19598 -3798 1825 -2607 -15599 8786 20835 -16845 -18923 -20391 -1147 -11345 -2975 2812 -322 14166 -4043 12275 20068 -12688 -4191 -4039 12656 10535 -24810 16317 -13975 -7352 17358 17087 -15661 18591 263 -2476 18206 11836 4245 21000 -6626 -23005 23172 12302 -18026 24715 -17144 2889 -1154 -17698 17308 -16148 3419 -5523 23767 -20127 -19943 -7670 -12448 21120 18021 -23530 4922 -1423 -6358 19581 -19913 -7087 -16912 10487 24417 -14664 19971 -17854 -15116 20656 -7707 11542 -2923 -23584 -24368 -11019 2971 12018 -4743 -18266 -20711 -14500 -8814 12055 14192 11400 -11943 20286 -14676 17400 -624 12479 2592 -19349 8794 -11555 17671 6755 1591 -19374 17543 -5366 -15117 -3307 -6105 -7917 -16095 18340 827 -12810 -932 19360 4062 -17287 -11234 -23665 -2055 11365 7731 8845 21403 3677 10903 15131 6621 5628 17929 1932 -4519 -20358 8917 -14152 -23034 15654 10923 13733 -9397 23708 19547 20206 -919 7157 5520 -6838 -20238 -18064 14284 6613 -7291 -4795 -9951 -778 -21093 -20505 6188 -12602 985 10312 -13179 -8662 -7985 -21100 24225 21051 -5695 -13908 5273 -19811 -16289 10634 19423 -7910 -19195 -197 2324 -2243 -15107 -19703 -1003 -2840 -19396 21114 -2450 10538 2319 8034 6549 20920 24969 -722 569 7100 -252 -2273 -19426 14561 -3069 13500 -5102 16809 9660 -4923 -5 8949 575 5776 -24962 2766 -18050 -5993 2404 23841 6238 -12149 12816 24469 10338 2681 12161 -19732 -6321 -10410 -4586 -2730 </pre>	false	false	✓

Passed all tests! ✓

Đúng

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

Câu hỏi 9

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

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.

Input:

- Tham số `array` là mảng được cho
- Tham số `size` là số phần tử của `array`
- Tham số `isAscending` chỉ thị mảng cần được sắp xếp tăng dần (`true`) hay không (`false`)

Output:

- Hàm chỉ làm nhiệm vụ sắp xếp mảng, không trả về giá trị gì.

Gợi ý:

- Viết một hàm `void sortAscending(float array[], int size)` cài đặt thuật toán sắp xếp yêu thích để sắp xếp 1 mảng theo thứ tự tăng dần.
- Viết một hàm `void reverseArray(float array[], int size)` cài đặt thuật toán đảo ngược thứ tự các phần tử của mảng, phần tử tại vị trí `i` đổi chỗ với phần tử tại vị trí `size - 1 - i`.
- Hàm `sort` sử dụng hàm `sortAscending` để sắp xếp mảng, sau đó kiểm tra nếu `isAscending == false` thì gọi thêm hàm `reverseArray`.

Lưu ý:

- Tìm hiểu thêm về hàm `sort` trong thư viện `algorithm`

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 void sort(float a[], int size, bool isAscending) {
2     for (int i = 0; i < size - 1; ++i) {
3         for (int j = i+1; j < size; ++j) {
4             if ((isAscending && a[j] < a[i]) || (!isAscending && a[j] > a[i])) {
5                 swap(a[i], a[j]);
6             }
7         }
8     }
9 }
```


	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! ✓

Đúng

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

Câu hỏi 10

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết hàm `string toLower (string s)` chuyển đổi chuỗi ký tự `s` thành chuỗi ký tự có tất cả các ký tự là chữ cái in thường. Hàm nhận đầu vào là một chuỗi `s` bất kỳ và trả về chuỗi đã được chuyển về dạng chữ thường (lowercase).

Input:

- Hàm nhận vào một chuỗi ký tự có kiểu dữ liệu là `string`.

Output:

- Hàm trả về một chuỗi ký tự có kiểu dữ liệu là `string` sau khi đã chuyển về chữ thường.

Gợi ý:

- Cách 1: để chuyển ký tự `char c` về viết thường dùng hàm `tolower(c)`
- Cách 2: để chuyển ký tự hoa `char c` về viết thường dùng công thức `c + 'a' - 'A'`

For example:

Test	Input	Result
<code>toLower(s)</code>	Convert all the StrIng to LowerCase.	convert all the string to lowercase.

Answer: (penalty regime: 0 %)

```
1 string toLower(string s) {
2     string t;
3     for (char c : s) {
4         if ('A' <= c && c <= 'Z') {
5             t.push_back(c + 32);
6         } else {
7             t.push_back(c);
8         }
9     }
10    return t;
11 }
```

	Test	Input	Expected	Got	
✓	toLowerCase(s)	Convert all the StrIng to LowerCase.	convert all the string to lowercase.	convert all the string to lowercase.	✓
✓	toLowerCase(s)	Also, C++, The Complete Reference, Second Edition, by Herbert Schildt, is an excellent book for answering those programming questions.	also, c++, the complete reference, second edition, by herbert schildt, is an excellent book for answering those programming questions.	also, c++, the complete reference, second edition, by herbert schildt, is an excellent book for answering those programming questions.	✓
✓	toLowerCase(s)	Digital Forest - Psychedelic Trance Music Festival	digital forest - psychedelic trance music festival	digital forest - psychedelic trance music festival	✓
✓	toLowerCase(s)	Digital Fortress is a techno-thriller novel written by American author Dan Brown and published in 1998 by St. Martin's Press. The book explores the theme of government surveillance of electronically stored information on the private lives of citizens, and the possible civil liberties and ethical implications of using such technology.	digital fortress is a techno-thriller novel written by american author dan brown and published in 1998 by st. martin's press. the book explores the theme of government surveillance of electronically stored information on the private lives of citizens, and the possible civil liberties and ethical implications of using such technology.	digital fortress is a techno-thriller novel written by american author dan brown and published in 1998 by st. martin's press. the book explores the theme of government surveillance of electronically stored information on the private lives of citizens, and the possible civil liberties and ethical implications of using such technology.	✓
✓	toLowerCase(s)	Tom M. Mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.[17] This definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper Computing Machinery and Intelligence, in which the question Can machines think? is replaced with the question Can machines do what we (as thinking entities) can do?.[18] In Turing's proposal the various characteristics that could be possessed by a thinking machine and the various implications in constructing one are exposed.	tom m. mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: a computer program is said to learn from experience e with respect to some class of tasks t and performance measure p if its performance at tasks in t, as measured by p, improves with experience e.[17] this definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. this follows alan turing's proposal in his paper computing machinery and intelligence, in which the question can machines think? is replaced with the question can machines do what we (as thinking entities) can do?.[18] in turing's proposal the various characteristics that could be possessed by a thinking machine and the various implications in constructing one are exposed.	tom m. mitchell provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: a computer program is said to learn from experience e with respect to some class of tasks t and performance measure p if its performance at tasks in t, as measured by p, improves with experience e.[17] this definition of the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. this follows alan turing's proposal in his paper computing machinery and intelligence, in which the question can machines think? is replaced with the question can machines do what we (as thinking entities) can do?.[18] in turing's proposal the various characteristics that could be possessed by a thinking machine and the various implications in constructing one are exposed.	✓

Passed all tests! ✓

Đúng

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

Câu hỏi 11

Đúng

Đạt điểm 10,00 trên 10,00

Pig Latin là một trò chơi ngôn ngữ trong đó các từ tiếng Anh bị thay đổi, thường là bằng cách thêm một hậu tố giả hoặc bằng cách di chuyển phần bắt đầu hoặc phụ âm đầu tiên hoặc cụm phụ âm của một từ về cuối từ và thêm một âm tiết để tạo ra một hậu tố. Mục đích là để che dấu các từ khỏi những người không biết luật.

Yêu cầu:

Hãy viết hàm `string pigLatin(string word)` để chuyển đổi một từ tiếng Anh đầu vào thành một từ trong ngôn ngữ Pig Latin với các quy tắc sau:

- Nếu từ bắt đầu bởi một phụ âm, chuyển phụ âm xuống cuối của từ và thêm "ay" vào cuối từ. Theo đó, "ball" trở thành "allbay".
- Nếu từ bắt đầu bởi một nguyên âm, thêm "way" vào cuối từ. Theo đó, "one" trở thành "oneway".

Input:

- Hàm nhận vào một chuỗi ký tự có kiểu dữ liệu là `string`.
- Bạn có thể giả sử từ đầu vào là một từ viết thường và không chứa các ký tự đặc biệt.

Output:

- Hàm trả về một chuỗi ký tự có kiểu dữ liệu là `string` sau khi đã chuyển đổi.

Gợi ý:

- Có thể dùng phép cộng (+) để nối string.
- Kiểm tra so sánh ký tự đầu tiên với các nguyên âm do số lượng ít hơn.
- Nếu bắt đầu là nguyên âm: trả về chuỗi đầu vào cộng thêm "way".
- Tương tự với phụ âm.
- Dùng hàm `s.substr(1)` để lấy ra chuỗi từ ký tự thứ 2 (index = 1) của chuỗi s.

Lưu ý:

- Chỉ cần viết hàm như đề bài yêu cầu, không cần viết hàm `main()`, không cần viết các câu lệnh `#include`, `using namespace std` ...

For example:

Test	Input	Result
<pre>string word; cin >> word; cout << pigLatin(word);</pre>	ball	allbay

Answer: (penalty regime: 0 %)

```
1 #include <string>
2 using namespace std;
3
4 string pigLatin(string word) {
5     const string s = "aeiou";
6     if (s.find(word[0]) == string::npos) {
7         return word.substr(1) + word[0] + "ay";
8     } return word + "way";
9 }
```

	Test	Input	Expected	Got	
✓	string word; cin >> word; cout << pigLatin(word);	ball	allbay	allbay	✓
✓	string word; cin >> word; cout << pigLatin(word);	zero	erozay	erozay	✓
✓	string word; cin >> word; cout << pigLatin(word);	all	allway	allway	✓
✓	string word; cin >> word; cout << pigLatin(word);	frozen	rozenfay	rozenfay	✓
✓	string word; cin >> word; cout << pigLatin(word);	optave	optaveway	optaveway	✓

Passed all tests! ✓

Đúng

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

Trở lại Khoá học