

Status	Finished
Started	Saturday, 11 January 2025, 8:43 AM
Completed	Saturday, 11 January 2025, 9:15 AM
Duration	32 mins 14 secs
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

Đọc vào từ bàn phím cho đến khi đọc được số nguyên dương (>0), in số này ra.

Kiểm tra số này có phải **lớn hơn 50 và nhỏ hơn 100** không. Nếu đúng in ra **YES**, nếu sai in ra **NO**.

For example:

Input	Result
12	12 NO

Answer: (penalty regime: 0 %)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     while (true) {
6         int x;
7         cin >> x;
8         if (x > 0) {
9             cout << x << ' ';
10            if (x > 50 && x < 100) {
11                cout << "YES";
12            } else {
13                cout << "NO";
14            }
15            return 0;
16        }
17    }
18 }
```

	Input	Expected	Got	
✓	12	12 NO	12 NO	✓
✓	56	56 YES	56 YES	✓
✓	198	198 NO	198 NO	✓
✓	64	64 YES	64 YES	✓
✓	5	5 NO	5 NO	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

Cho 02 số nguyên $N (>0)$, $K (>0)$, in ra N số nguyên dương (>0) đầu tiên **không** là ước số của K , các số cách nhau bởi một dấu cách. Nếu đầu vào không hợp lệ in ra INVALID.

Answer: (penalty regime: 0 %)

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n, k;
6     cin >> n >> k;
7
8     if (n <= 0 || k <= 0) {
9         cout << "INVALID";
10        return 0;
11    }
12    for (int i = 2; ; ++i) {
13        if (k % i) {
14            cout << i << ' ';
15            if (--n == 0) {
16                return 0;
17            }
18        }
19    }
20 }
```

	Input	Expected	Got	
✓	3 7	2 3 4	2 3 4	✓
✓	16 7	2 3 4 5 6 8 9 10 11 12 13 14 15 16 17 18	2 3 4 5 6 8 9 10 11 12 13 14 15 16 17 18	✓
✓	97 13	2 3 4 5 6 7 8 9 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99	2 3 4 5 6 7 8 9 10 11 12 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99	✓
✓	-34 124	INVALID	INVALID	✓
✓	234 0	INVALID	INVALID	✓
✓	5 8	3 5 6 7 9	3 5 6 7 9	✓
✓	1 10	3	3	✓
✓	5 2	3 4 5 6 7	3 4 5 6 7	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3

Correct

Mark 1.00 out of 1.00

Nhập vào một dãy số nguyên gồm n ($n \leq 100$) phần tử. Đếm số phần tử chẵn và tính tổng các phần tử này. In ra màn hình số lượng [số chẵn](#) và tổng của chúng. Nếu dãy không có [số chẵn](#) nào thì không in gì.

Đầu vào

Đầu vào từ bàn phím gồm hai dòng:

- Dòng đầu tiên chứa số nguyên n ($n \leq 100$);
- Dòng thứ hai chứa n số nguyên là các phần tử của dãy, phân tách nhau bởi dấu cách.

Đầu ra

In ra màn hình gồm hai dòng:

- Dòng đầu tiên in ra các phần tử là [số chẵn](#) trong dãy, phân tách nhau bởi dấu cách;
- Dòng thứ hai in ra tổng của chúng.

Nếu dãy không có [số chẵn](#) nào thì không in gì.

For example:

Input	Result
5	0 4 8
0 1 4 7 8	12

Answer: (penalty regime: 0 %)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n, total = 0, count = 0;
6     for (cin >> n; n--; ) {
7         int x;
8         cin >> x;
9         if (x % 2 == 0) {
10             cout << x << ' ';
11             ++count;
12             total += x;
13         }
14     }
15     if (count) {
16         cout << '\n' << total;
17     }
18 }
```

	Input	Expected	Got	
✓	5 0 1 4 7 8	0 4 8 12	0 4 8 12	✓
✓	7 -1 1 3 5 7 11 15			✓
✓	20 2 -8 -10 14 6 8 4 -24 16 18 -40 -14 34 50 46 48 24 -2 12 -22	2 -8 -10 14 6 8 4 -24 16 18 -40 -14 34 50 46 48 24 -2 12 -22 162	2 -8 -10 14 6 8 4 -24 16 18 -40 -14 34 50 46 48 24 -2 12 -22 162	✓
✓	63 6 -161 4 133 10 49 153 30 -63 36 107 15 109 141 22 77 -31 43 111 9 8 29 -143 55 67 -45 64 81 -35 -113 12 -151 51 0 73 105 157 -41 -101 3 -100 36 -91 26 -135 125 61 97 -27 -151 81 -41 163 145 105 -3 125 47 26 115 -81 153 -161	6 4 10 30 36 22 8 64 12 0 -100 36 26 26 180	6 4 10 30 36 22 8 64 12 0 -100 36 26 26 180	✓
✓	90 2 -8 7 14 10 5 3 -24 18 4 16 -32 45 48 33 -20 66 88 -42 0 1 6 22 -50 100 -15 8 -18 12 20 -40 90 78 36 -64 -22 25 30 9 55 70 13 44 42 80 -72 -10 19 17 23 31 35 -11 60 64 2 -6 76 -14 24 58 -90 28 -66 120 -8 26 38 -54 18 52 82 -86 -99 102 150 210 -240 300 11 14 -16 50 -48 110 8 130 0 -20 111	2 -8 14 10 -24 18 4 16 -32 48 -20 66 88 -42 0 6 22 -50 100 8 -18 12 20 -40 90 78 36 -64 -22 30 70 44 42 80 -72 -10 60 64 2 -6 76 -14 24 58 -90 28 -66 120 -8 26 38 -54 18 52 82 -86 102 150 210 -240 300 14 -16 50 -48 110 8 130 0 -20 1576	2 -8 14 10 -24 18 4 16 -32 48 -20 66 88 -42 0 6 22 -50 100 8 -18 12 20 -40 90 78 36 -64 -22 30 70 44 42 80 -72 -10 60 64 2 -6 76 -14 24 58 -90 28 -66 120 -8 26 38 -54 18 52 82 -86 102 150 210 -240 300 14 -16 50 -48 110 8 130 0 -20 1576	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.50 out of 1.50

Cho 01 chuỗi kí tự chỉ gồm các kí tự chữ cái thường từ 'a' đến 'z'. Đếm số lần xuất hiện của các kí tự này. Sau đó in ra **kí tự** và **số lượng** kí tự đó, kí tự nào không xuất hiện trong chuỗi thì không cần in. In ra theo thứ tự **sắp xếp tăng dần** của kí tự trong bảng mã ASCII.

For example:

Input	Result
hellocplusplus	c: 1 e: 1 h: 1 l: 4 o: 1 p: 2 s: 2 u: 2

Answer: (penalty regime: 0 %)

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     string s;
6     cin >> s;
7     int cnt[26] = {0};
8     for (size_t i = 0; i < s.length(); ++i) {
9         ++cnt[s[i] - 'a'];
10    }
11    for (int i = 0; i < 26; ++i) {
12        if (cnt[i]) {
13            cout << char(i + 'a') << ": " << cnt[i] << '\n';
14        }
15    }
16 }
```

	Input	Expected	Got	
✓	abcbacxyz	a: 2 b: 2 c: 2 x: 1 y: 1 z: 1	a: 2 b: 2 c: 2 x: 1 y: 1 z: 1	✓

	Input	Expected	Got	
✓	helloplusplus	c: 1 e: 1 h: 1 l: 4 o: 1 p: 2 s: 2 u: 2	c: 1 e: 1 h: 1 l: 4 o: 1 p: 2 s: 2 u: 2	✓
✓	denhunaycungsai	a: 2 c: 1 d: 1 e: 1 g: 1 h: 1 i: 1 n: 3 s: 1 u: 2 y: 1	a: 2 c: 1 d: 1 e: 1 g: 1 h: 1 i: 1 n: 3 s: 1 u: 2 y: 1	✓
✓	homnayangi	a: 2 g: 1 h: 1 i: 1 m: 1 n: 2 o: 1 y: 1	a: 2 g: 1 h: 1 i: 1 m: 1 n: 2 o: 1 y: 1	✓
✓	daihoccongnghez	a: 1 c: 2 d: 1 e: 1 g: 2 h: 2 i: 1 n: 2 o: 2 z: 1	a: 1 c: 2 d: 1 e: 1 g: 2 h: 2 i: 1 n: 2 o: 2 z: 1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.50/1.50.

Question 5

Correct

Mark 1.50 out of 1.50

Cho 02 số nguyên $N (>0)$, $K (>0)$, và mảng $N (<100)$ số nguyên. Đếm trong mảng có bao nhiêu số nguyên có K chữ số. Viết hàm `int kDigitsNumbers(int *a, int n, int k)` trả về số đếm được này.

Answer: (penalty regime: 0 %)

```
1 int kDigitsNumbers(int* a, int n, int k) {
2     int count = 0;
3     for (int i = 0; i < n; ++i) {
4         string s = to_string(abs(a[i]));
5         count += int(s.length()) == k;
6     }
7     return count;
8 }
```

	Input	Expected	Got	
✓	5 2 1 23 456 78 900	2	2	✓
✓	1 3 123	1	1	✓
✓	4 2 1000 12345 6789 54321	0	0	✓
✓	5 3 100 234 567 890 123	5	5	✓
✓	6 1 1 2 3 4 5 6	6	6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.50/1.50.

Question 6

Correct

Mark 1.50 out of 1.50

Xây dựng [cấu trúc](#) (struct) phân số gồm 2 thành phần dữ liệu kiểu số nguyên tử số (num), mẫu số (den) và hoàn thành nốt các hàm xử lý trên kiểu dữ liệu này.

Tính tổng 2 đại lượng phân số `PhanSo tong (const PhanSo& ps1, const PhanSo& ps2);`

So sánh lớn hơn 2 phân số (trả về đúng nếu ps1 lớn hơn ps2) `bool ssLonHon(const PhanSo& ps1, const PhanSo& ps2);`

Chú ý: Không cần viết đầy đủ cả chương trình. Không sửa, xóa code đã cho trước.

For example:

Input	Result
1 2 1 3	5 6 true

Answer: (penalty regime: 0 %)

Reset answer

```

1 #include <iostream>
2
3 using namespace std;
4
5 struct PhanSo {
6     int num, den;
7 };
8
9 int gcd(int a, int b) {
10     return b == 0? a : gcd(b, a % b);
11 }
12
13 PhanSo tong(const PhanSo &ps1, const PhanSo &ps2) {
14     // your code here
15     int num = ps1.num * ps2.den + ps2.num * ps1.den;
16     int den = ps1.den * ps2.den;
17     bool neg = (num * den) < 0;
18     num = abs(num);
19     den = abs(den);
20     int g = gcd(num, den);
21     num /= g;
22     den /= g;
23     if (neg) {
24         num = -num;
25     }
26     return {num, den};
27 };
28
29 bool ssLonHon(const PhanSo &ps1, const PhanSo &ps2) {
30     // your code here
31     double v1 = double(ps1.num) / ps1.den;
32     double v2 = double(ps2.num) / ps2.den;
33     return v1 > v2;
34 };
35
36 int main() {
37     int num, den;
38     cin >> num >> den;
39     PhanSo ps1 = {num, den};
40     cin >> num >> den;
41     PhanSo ps2 = {num, den};
42
43     PhanSo ps3 = tong(ps1, ps2);
44     cout << ps3.num << " " << ps3.den << " " << boolalpha << ssLonHon(ps1, ps2) << endl;
45     return 0;
46 }
47

```

	Input	Expected	Got	
✓	1 2 1 3	5 6 true	5 6 true	✓
✓	-1 2 1 3	-1 6 false	-1 6 false	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.50/1.50.

Question 7

Correct

Mark 1.50 out of 1.50

Cho dãy n số nguyên ($1 \leq n \leq 1000$), bạn hãy tìm xem trong dãy có phần tử nào cân bằng được dãy số này không? Phần tử nào thỏa mãn tổng các phần tử bên trái nó bằng tổng các phần tử bên phải nó thì được gọi là Phần tử cân bằng. Nếu bên trái hoặc bên phải không có phần tử nào thì tổng coi là 0. Bạn chỉ phải viết hàm `bool isBalance (int *a, int n)` mà không cần viết cả chương trình. Trong các ví dụ dưới đây, số đầu tiên là số n – số lượng phần tử của dãy, n số sau là các phần tử của dãy đó.

For example:

Input	Result
5 4 3 2 4 3	true
6 1 4 3 2 6 7	false

Answer: (penalty regime: 0 %)

```
1 bool isBalance(int* a, int n) {
2     for (int i = 0; i < n; ++i) {
3         int sumleft = 0;
4         for (int j = 0; j < i; ++j) {
5             sumleft += a[j];
6         }
7         int sumright = 0;
8         for (int j = i+1; j < n; ++j) {
9             sumright += a[j];
10        }
11        if (sumleft == sumright) {
12            return true;
13        }
14    }
15    return false;
16 }
17
```

	Input	Expected	Got	
✓	5 4 3 2 4 3	true	true	✓
✓	6 1 4 -6 3 6 -7	true	true	✓
✓	6 1 4 3 2 6 7	false	false	✓
✓	5 -3 6 -1 0 2	true	true	✓
✓	10 0 0 0 1 2 3 4 5 6 0	false	false	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.50/1.50.

Question 8

Correct

Mark 1.00 out of 1.00

Sử dụng mảng 2 chiều quản lý ma trận kích thước $N \times N$ ($N < 10$). Dữ liệu của mảng này là các giá trị **true** / **false** (kiểu **bool**) để thể hiện ô màu trắng / đen tương ứng. **Ví dụ:** mảng **true** (trắng) / **false** (đen)

	0	1	2	3	4	5
0	F	T	F	T	F	T
1	F	T	T	F	T	T
2	F	T	T	T	T	T
3	T	T	F	T	F	T
4	T	T	T	F	T	T
5	T	T	F	T	T	T

Sinh viên thực hiện các yêu cầu sau:

(a) Viết hàm Tìm kiếm tất cả các ma trận con có kích thước **2x2**, trong đó tất cả 4 ô vuông của ma trận **2x2** này đều màu trắng hoặc màu đen. Với mỗi ma trận con tìm được, in ra chỉ số của ô góc trên trái của ma trận con này và các cặp chỉ số này được sắp xếp tăng dần theo thứ tự chỉ số thứ nhất rồi chỉ số thứ hai. **Ví dụ:** với ma trận như hình trên cần in ra chỉ số của 5 ma trận con tương ứng là (1,1); (1,4); (3,0); (4,0); (4,4).

```
void findSubMatrix(int **a, int n);
```

(b) Sắp xếp các cặp chỉ số tìm được ở (a) theo thứ tự tăng dần của giá trị tổng chỉ số (cộng 2 chỉ số của cặp; được tổng; sắp xếp tăng dần theo tổng này), sau đó in ra các cặp chỉ số này. **Ví dụ:** với ma trận như hình trên các cặp chỉ số in ra theo thứ tự sau (1,1) tổng 2; (3,0) tổng 3; (4,0) tổng 4; (1,4) tổng 5; và (4,4) tổng 8.

```
void sortBySum(int **a, int n);
```

For example:

Test	Input	Result
int n;	6	1 1
cin >> n;	0 1 0 1 0 1	1 4
int** a = new int*[n];	0 1 1 0 1 1	3 0
for (int i = 0; i < n; ++i) {	0 1 1 1 1 1	4 0
a[i] = new int[n];	1 1 0 1 0 1	4 4
}	1 1 1 0 1 1	
	1 1 0 1 1 1	
for (int i = 0; i < n; ++i) {		
for (int j = 0; j < n; ++j) {		
cin >> a[i][j];		
}		
}		
findSubMatrix(a, n);		

Test	Input	Result
int n;	6	1 1
cin >> n;	0 1 0 1 0 1	3 0
int** a = new int*[n];	0 1 1 0 1 1	4 0
for (int i = 0; i < n; ++i) {	0 1 1 1 1 1	1 4
a[i] = new int[n];	1 1 0 1 0 1	4 4
}	1 1 1 0 1 1	
	1 1 0 1 1 1	
for (int i = 0; i < n; ++i) {		
for (int j = 0; j < n; ++j) {		
cin >> a[i][j];		
}		
}		
sortBySum(a, n);		

Answer: (penalty regime: 0 %)

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  void findSubMatrix(int** a, int n) {
7      vector<pair<int, int>> v;
8      for (int i = 0; i < n-1; ++i) {
9          for (int j = 0; j < n-1; ++j) {
10             int s = a[i][j] + a[i+1][j] + a[i][j+1] + a[i+1][j+1];
11             if (s == 0 || s == 4) {
12                 v.emplace_back(i, j);
13             }
14         }
15     }
16     sort(begin(v), end(v));
17     for (auto [i, j] : v) {
18         cout << i << ' ' << j << '\n';
19     }
20 }
21
22 void sortBySum(int** a, int n) {
23     vector<pair<int, int>> v;
24     for (int i = 0; i < n-1; ++i) {
25         for (int j = 0; j < n-1; ++j) {
26             int s = a[i][j] + a[i+1][j] + a[i][j+1] + a[i+1][j+1];
27             if (s == 0 || s == 4) {
28                 v.emplace_back(i, j);
29             }
30         }
31     }
32     sort(begin(v), end(v), [](pair<int, int> a, pair<int, int> b) {
33         return a.first + a.second < b.first + b.second;
34     });
35     for (auto [i, j] : v) {
36         cout << i << ' ' << j << '\n';
37     }
38 }

```

	Test	Input	Expected	Got	
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } findSubMatrix(a, n); </pre>	<pre> 6 0 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 </pre>	<pre> 1 1 1 4 3 0 4 0 4 4 </pre>	<pre> 1 1 1 4 3 0 4 0 4 4 </pre>	✓
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } findSubMatrix(a, n); </pre>	<pre> 4 1 0 0 0 1 1 1 0 0 0 1 1 0 0 1 1 </pre>	<pre> 2 0 2 2 </pre>	<pre> 2 0 2 2 </pre>	✓
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } findSubMatrix(a, n); </pre>	<pre> 5 0 0 1 1 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 1 0 1 1 0 0 </pre>	<pre> 0 0 0 2 1 0 2 2 </pre>	<pre> 0 0 0 2 1 0 2 2 </pre>	✓

	Test	Input	Expected	Got	
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } findSubMatrix(a, n); </pre>	<pre> 3 1 1 1 1 1 1 1 1 1 </pre>	<pre> 0 0 0 1 1 0 1 1 </pre>	<pre> 0 0 0 1 1 0 1 1 </pre>	✓
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } findSubMatrix(a, n); </pre>	<pre> 9 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 </pre>	<pre> 0 0 0 2 0 4 0 6 2 0 2 2 2 4 2 6 4 0 4 2 4 4 4 6 6 0 6 2 6 4 6 6 </pre>	<pre> 0 0 0 2 0 4 0 6 2 0 2 2 2 4 2 6 4 0 4 2 4 4 4 6 6 0 6 2 6 4 6 6 </pre>	✓
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } sortBySum(a, n); </pre>	<pre> 6 0 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 </pre>	<pre> 1 1 3 0 4 0 1 4 4 4 </pre>	<pre> 1 1 3 0 4 0 1 4 4 4 </pre>	✓

	Test	Input	Expected	Got	
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } sortBySum(a, n); </pre>	<pre> 4 1 0 0 0 1 1 1 0 0 0 1 1 0 0 1 1 </pre>	<pre> 2 0 2 2 </pre>	<pre> 2 0 2 2 </pre>	✓
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } sortBySum(a, n); </pre>	<pre> 5 0 0 1 1 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 1 0 1 1 0 0 </pre>	<pre> 0 0 1 0 0 2 2 2 </pre>	<pre> 0 0 1 0 0 2 2 2 </pre>	✓
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } sortBySum(a, n); </pre>	<pre> 3 1 1 1 1 1 1 1 1 1 </pre>	<pre> 0 0 0 1 1 0 1 1 </pre>	<pre> 0 0 0 1 1 0 1 1 </pre>	✓

	Test	Input	Expected	Got	
✓	<pre> int n; cin >> n; int** a = new int*[n]; for (int i = 0; i < n; ++i) { a[i] = new int[n]; } for (int i = 0; i < n; ++i) { for (int j = 0; j < n; ++j) { cin >> a[i][j]; } } sortBySum(a, n); </pre>	<pre> 9 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 2 6 4 4 6 2 4 6 6 4 6 6 </pre>	<pre> 0 0 0 2 2 0 0 4 2 2 4 0 0 6 2 4 4 2 6 0 2 6 4 4 6 2 4 6 6 4 6 6 </pre>	<pre> 0 0 0 2 2 0 0 4 2 2 4 0 0 6 2 4 4 2 6 0 2 6 4 4 6 2 4 6 6 4 6 6 </pre>	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[Back to Course](#)