Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết chương trình nhận vào một xâu kí tự là đường dẫn đến một tệp. Thử xem có mở được tệp đó không, nếu có in ra YES ngược lại in ra NO. Mở xong nhớ đóng tệp lại.

Input:

• Một dòng duy nhất là một xâu kí tự là đường dẫn đến một tệp

Output:

- Thử xem có mở được tệp đó không, nếu có in ra YES ngược lại in ra NO.

Gợi ý:

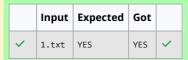
1. Dùng hàm is_open để kiểm tra xem file đã được mở thành công hay chưa.

For example:

Input	Result
1.txt	YES

Answer: (penalty regime: 0 %)

```
#include <iostream>
    #include <fstream>
    using namespace std;
4
 5 🔻
    int main() {
6
      string path;
      cin >> path;
8
      ifstream file(path);
      if (file.is_open()) {
  cout << "YES";</pre>
9 ,
10
         file.close();
11
12 🔻
      } else {
13
         cout << "NO";
14
15
    }
16
```



	Input	Expected	Got	
~	2.txt	NO	NO	~

Passed all tests! 🗸

Đúng

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết chương trình đọc các số nguyên từ một tệp tin. In ra màn hình giá trị lớn nhất, giá trị nhỏ nhất.

Input:

• Một dòng duy nhất chứa đường dẫn đến tệp tin.

Output:

- Thử mở tệp này, nếu không thành công in ra "Mission failed".
- Trong trường hợp mở được, tệp sẽ chứa các số nguyên cách nhau bởi dấu cách hoặc/và dấu xuống dòng. In ra màn hình số lớn nhất và nhỏ nhất trong tệp cách nhau bởi dấu cách.

Gợi ý:

- 1. Dùng hàm is_open để kiểm tra xem file đã được mở thành công hay chưa.
- 2. Dùng while(fin >> x) để đọc liên tiếp vào biến x từ luồng fin.

For example:

Input	Result
1.txt	23 1

Answer: (penalty regime: 0 %)

```
#include <iostream>
 2
    #include <fstream>
 3
    using namespace std;
5 v int main() {
6
     string path;
 7
      cin >> path;
      ifstream in(path);
8
9 1
      if (!in.is_open()) {
10
        cout << "Mission failed";</pre>
11
        return 0;
12
13
      int mn = 2e9;
14
      int mx = -2e9;
15
      int x;
      while (in >> x) {
16
17
        mn = min(mn, x);
18
        mx = max(mx, x);
19
      cout << mx << ' ' << mn;
20
21
```

	Input	Expected	Got	
~	1.txt	23 1	23 1	~
~	2.txt	58 1	58 1	~
~	3.txt	65 1	65 1	~
~	4.txt	Mission failed	Mission failed	~
~	5.txt	89475641 5	89475641 5	~

Passed all tests! 🗸

Đúng

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết chương trình đọc các số nguyên từ một tệp tin. In ra màn hình số lượng các số, giá trị lớn nhất, giá trị nhỏ nhất và tổng của các số đó.

Input:

• Một dòng duy nhất chứa đường dẫn đến tệp tin. Nội dung của tệp là một dãy các số nguyên cách nhau bởi dấu cách.

Output:

• In ra màn hình số lượng các số, giá trị nhỏ nhất, giá trị lớn nhất và tổng các số trên cùng một dòng, cách nhau bởi dấu cách.

Gợi ý:

1. Dùng while(fin >> x) để đọc liên tiếp vào biến x từ luồng fin.

For example:

Input	R	es	ul	t
1.txt	7	1	8	31

Answer: (penalty regime: 0 %)

```
#include <iostream>
    #include <fstream>
   using namespace std;
4
 5 v int main() {
6
     string path;
     cin >> path;
     ifstream in(path);
8
9
     int mn = 2e9;
     int mx = -2e9;
10
     int sum = 0;
11
      int count = 0;
12
13
      int x;
14
      while (in >> x) {
       mn = min(mn, x);
15
16
       mx = max(mx, x);
       sum += x;
17
18
19
      cout << count << ' ' << mn << ' ' << mx << ' ' << sum;
20
21 }
```

Passed all tests! ✓

Đúng

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết chương trình đếm số dòng trong tệp.

Input:

• Một dòng duy nhất chứa đường dẫn đến tệp tin.

Output:

- Thử mở tệp đó, nếu không mở được thì in ra dòng "Mission failed".
- Nếu mở được thì đếm xem tệp có bao nhiêu dòng. Biết rằng tệp là một tệp văn bản, mỗi dòng có thể chứa chuỗi kí tự, số thực, số nguyên hoặc dòng trống

Gợi ý:

1. Dùng getline để đọc một dòng từ luồng dữ liệu tương tự khi dùng với cin.

For example:

Input	Result
1.txt	6

Answer: (penalty regime: 0 %)

```
#include <iostream>
    #include <fstream>
2
3
    using namespace std;
4
 5 v int main() {
      string path;
6
 7
       cin >> path;
      ifstream in(path);
8
      if (!in.is_open()) {
  cout << "Mission failed";</pre>
9 ,
10
11
         return 0;
12
13
      int count = 0;
14
       string s;
      while (getline(in, s)) {
15 🔻
16
         count++;
17
18
      cout << count + 1;</pre>
19 }
```

	Input	Expected	Got	
~	1.txt	6	6	~

	Input	Expected	Got	
~	2.txt	4	4	~
~	3.txt	5	5	~
~	4.txt	Mission failed	Mission failed	~
~	5.txt	6	6	~

Passed all tests! <

Đúng

Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết hàm string combine (const vector < string > & words) nối tất cả các từ trong words thành một xâu s. Biết các từ phải được sắp xếp giảm dần trước khi nối. Giữa các từ trong xâu s thêm một khoảng trắng.

Input:

• Hàm nhận vào một vector chứa các xâu ký tự vector<string> words

Output:

ullet Hàm trả về xâu s là kết quả phép nối.

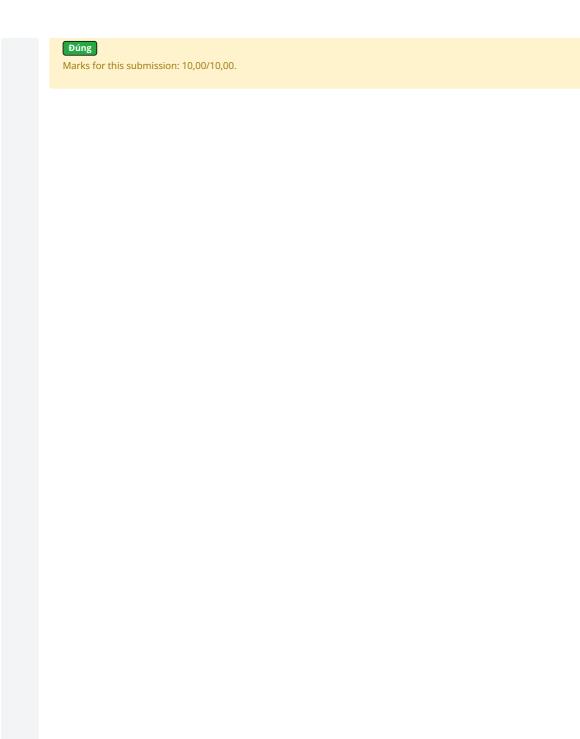
For example:

Test	Result
string s[] = {"this", "is", "a", "test"};	this test is a
<pre>vector<string> words(s, s+4);</string></pre>	true
<pre>string combined = combine(words);</pre>	
cout << combined << endl;	
<pre>cout << boolalpha << ("this test is a" == combined);</pre>	

Answer: (penalty regime: 0 %)

```
1 | string combine(const vector<string>& words) {
 2
       vector<string> v = words;
 3
       sort(rbegin(v), rend(v));
 4
       string s = v[0];
       for (int i = 1; i < int(v.size()); ++i) {
    s.push_back(' ');</pre>
 5
 6
 7
         s.append(v[i]);
 8
      }
 9
      return s;
10 }
```

Passed all tests! <



Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Viết hàm string combineWordsFromFile(const char fileName[]) nối tất cả các từ trong tệp có tên fileName thành một xâu s. Hàm cần phải sắp xếp các từ giảm dần theo thứ tự từ điển trước khi nối. Giữa các từ trong xâu s thêm một khoảng trắng.

Input:

• Hàm nhận vào một đường dẫn đến một file. Cho biết mỗi dòng của tệp này chứa 1 từ.

Output:

• Hàm trả về xâu kết quả.

For example:

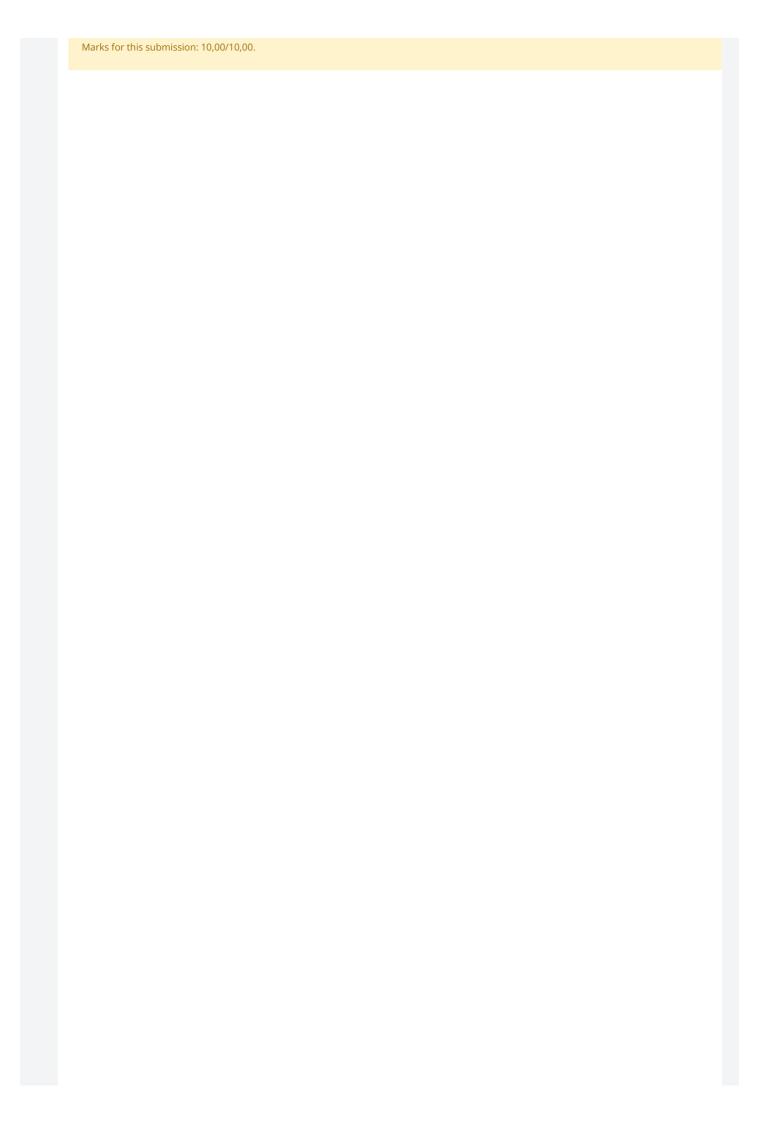
Test	Result
<pre>char testFileName[] = "words1.txt";</pre>	this test is a
<pre>string combined = combineWordsFromFile(testFileName);</pre>	true
cout << combined << endl;	
<pre>cout << boolalpha << ("this test is a" == combined);</pre>	

Answer: (penalty regime: 0 %)

```
1 | string combineWordsFromFile(const char fileName[]) {
2
      ifstream in(fileName);
3
      vector<string> v;
4
      string s;
 5
      while (in >> s) {
6
        v.emplace_back(s);
 7
8
      sort(rbegin(v), rend(v));
9
      s = v[0];
      for (int i = 1; i < int(v.size()); ++i) {
    s.push_back(' ');</pre>
10 🔻
11
12
         s.append(v[i]);
13
      }
14
      return s;
15 }
```

	Test	Expected	Got	
~	<pre>char testFileName[] = "words1.txt";</pre>	this test is a	this test is a	~
	<pre>string combined = combineWordsFromFile(testFileName);</pre>	true	true	
	cout << combined << endl;			
	<pre>cout << boolalpha << ("this test is a" == combined);</pre>			

Passed all tests! ✓



Đúng

Đạt điểm 10,00 trên 10,00

Yêu cầu:

Số trung vị của một dãy là số chia dãy đó thành 2 nửa, một nửa có giá trị nhỏ hơn hoặc bằng số đó, một nửa có giá trị lớn hơn hoặc bằng số đó. Viết chương trình đọc từ một tệp chứa dãy các số nguyên và in ra màn hình số trung vị của dãy đó.

Input:

• Một dòng duy nhất chứa đường dẫn đến tệp. Bên trong tệp, dòng đầu tiên chứa số n là số lượng các số trong dãy, dòng tiếp theo chứa n số là các giá trị của dãy.

Output:

• In ra màn hinh số trung vị của dãy số trong tệp.

Gợi ý:

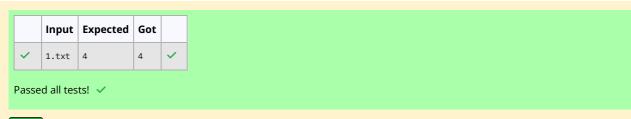
- 1. Đọc vào dãy số và sắp xếp dãy số.
- 2. Trả về số đứng giữa trong dãy đã sắp xếp.

For example:

Input	Result
1.txt	4

Answer: (penalty regime: 0 %)

```
1 #include <iostream>
    #include <fstream>
3
    #include <vector>
 4
    #include <algorithm>
   using namespace std;
7 v int main() {
8
     string path;
9
      cin >> path;
10
      ifstream in(path);
11
      int n;
12
      in >> n;
      vector<int> a(n);
13
      for (int &x : a) {
14
15
        in \gg x;
16
      }
17
      sort(begin(a), end(a));
18
19
      cout << a[n/2];
20 }
```



Đúng

Trở lại Khoá học