

Status	Finished
Started	Sunday, 12 January 2025, 3:25 AM
Completed	Sunday, 12 January 2025, 3:25 AM
Duration	9 secs
Marks	0.00/100.00
Grade	0.00 out of 10.00 (0%)

Question 1

Not answered

Mark 0.00 out of 10.00

[MaxAndMinFromFile]

Viết chương trình nhận đầu vào là một [xâu kí tự](#) chứa đường dẫn đến một tệp. 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.

For example:

Input	Result
1.txt	23 1

Answer: (penalty regime: 0 %)

1	
---	--

## Question 2

Not answered

Mark 0.00 out of 10.00

### [Perimeter]

Viết chương trình, nhập vào đường dẫn đến một tệp. Tệp này chứa thông tin của các hình vuông, hình tròn và [hình chữ nhật](#). In ra chu vi của các hình trong tệp trên từng dòng, theo thứ tự xuất hiện trong tệp. Kết quả được làm tròn đến hai chữ số thập phân sau dấu phẩy.

Một tệp ví dụ như sau:

```
TYPE: SQUARE
SIDE: 5.3

TYPE: RECTANGLE
WIDTH: 5.0 HEIGHT: 16.0

TYPE: CIRCLE
RADIUS: 2.15
```

Một tệp có thể chứa thông tin của một hoặc nhiều hình và không theo thứ tự nhất định. Mỗi hình sẽ được định nghĩa trên hai dòng, dòng đầu tiên là loại hình (SQUARE, RECTANGLE, CIRCLE) và dòng tiếp theo là các thông số tương ứng với hình đó (SIDE, WIDTH, HEIGHT, RADIUS). Giữa thông tin của hai hình sẽ có một dòng trống.

For example:

Input	Result
1.txt	28.09 80.00 13.51

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

1	
---	--

### Question 3

Not answered

Mark 0.00 out of 10.00

#### [RocketScience]

Bạn nhận được một tệp văn bản chứa tọa độ điểm xuất phát, tọa độ mục tiêu và vận tốc của một tên lửa (giả sử tên lửa bay thẳng với vận tốc không đổi). Hãy viết một hàm tính thời gian tên lửa cần để bay tới tiêu diệt mục tiêu của mình: `double estimatedTime(const char* path)`. Hàm nhận đầu vào là [xâu kí tự](#) chứa đường dẫn tới tệp thiết lập tên lửa và trả về số giây tên lửa hoàn thành hành trình. Đơn vị đo khoảng cách là *km* và đo tốc độ là *km/s*.

Tệp thiết lập mẫu có [cấu trúc](#) như sau:

```
SRC X: 0 Y: 0
DES X: 10000 Y: 10000
VELOCITY: 6 KM/S
```

For example:

Input	Result
1.txt	2357.02

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

1	
---	--

**Question 4**

Not answered

Mark 0.00 out of 10.00

**[Grade10]**

Cho một bảng danh sách sinh viên và điểm của sinh viên đó như sau:

STT	Ho	Ten	Ngày Sinh	Dai so	THCS4	THCS1
1	Truong	Vo Ky	10/10/1999	8	10	9
2	Chu	Chi	Nhuoc	13/9/1999	10	10

Bảng đó được lưu trong tệp `.csv`:

```
STT, Ho, Ten, Ngày sinh, Dai so, THCS4, THCS1
```

```
1, Truong Vo, Ky, 10/10/1999, 8, 10, 9
```

```
2, Chu Chi, Nhuoc, 13/9/1999, 10, 10, 10
```

Hãy viết hàm `void grade10(const char* path)` nhận tham số đầu vào là một chuỗi kí tự chứa đường dẫn đến tệp `.csv` và in ra đầy đủ họ tên của các sinh viên có điểm THCS4 là 10 (in theo thứ tự trong bảng).

**For example:**

Input	Result
1.csv	Truong Vo Ky Chu Chi Nhuoc

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

1

### Question 5

Not answered

Mark 0.00 out of 10.00

#### [MoveZeroes]

Viết hàm `void moveZeroes (int * arr, char * file)` thực hiện các công việc sau:

- Đọc các số nguyên dương từ tệp \(\ file \) vào mảng \(\ arr \). Biết dòng đầu tiên của tệp lưu số nguyên \(\ n \) là số số nguyên dương, dòng tiếp theo chứa \(\ n \) số nguyên dương cần lưu vào mảng.
- Chuyển tất cả các phần tử của mảng có giá trị bằng \(\ 0 \) về cuối dãy, đồng thời, giữ nguyên vị trí tương đối của các phần tử còn lại trong mảng
- In dãy thu được ra màn hình

**Chú ý:** Bạn chỉ phải định nghĩa hàm theo yêu cầu mà **không** phải viết hàm `main()`, các thư viện cũng đã có sẵn.

For example:

Input	Result
first_array	272 881 478 93 645 14 0

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

1	
---	--

**Question 6**

Not answered

Mark 0.00 out of 10.00

**[MaxProductOfThree]**

Với một dãy số nguyên cho trước, hãy viết hàm `int maximumProduct (char * path)` tìm bộ ba số có tích lớn nhất trong tất cả các bộ số và trả về tích lớn nhất đó.

Hàm nhận đầu vào là đường dẫn đến tệp lưu mảng số nguyên cho trước. Hàm trả về tích lớn nhất có thể có.

Dòng đầu tiên của tệp chứa kích cỡ  $(n)$ , dòng kế tiếp chứa  $(n)$  số nguyên cách nhau bởi một khoảng trắng.

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

1	
---	--

**Question 7**

Not answered

Mark 0.00 out of 10.00

**[NeedForSpeed]**

Frank Martin sau khi thôi không làm người vận chuyển nữa thì anh chuyển sang làm tài xế thử xe cho hãng xe. Công việc hàng ngày của anh là chạy những chiếc xe mới của hãng vài vòng và tính giờ. Kết quả được ghi trong một bảng biểu và được lưu trữ trong một tệp \(.csv\). Cột đầu tiên của bảng là số thứ tự vòng đua (lap) và cột tiếp theo là thời gian Frank hoàn thành vòng đua đó, ví dụ một bảng và [cấu trúc](#) tệp tương ứng như sau:

**Lap Time**

1 143.23

2 134.12

Lap, Time

1,143.23

2,134.12

Cuối cùng, Frank cần phải tính thời gian trung bình anh hoàn thành vòng đua. Hãy giúp Frank bằng cách viết hàm **double** `averageTime(const char* path)`, nhận tham số là [xâu kí tự](#) lưu đường dẫn tới tệp \(.csv\) và trả về thời gian trung bình hoàn thành vòng đua. Lưu ý, tùy tâm trạng Frank sẽ chạy thử xe nhiều vòng hay ít vòng, nên số vòng đua trong các tệp \(.csv\) là không cố định.

**For example:**

Input	Result
1.csv	138.68

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

1





Question 8

Not answered

Mark 0.00 out of 10.00

[NotBoringMovies]

Thành phố X mở một rạp chiếu phim mới, rất nhiều người muốn đến xem thử và hưởng các khuyến mại trong ngày khai trương. Để đưa cho khách hàng các lựa chọn, rạp chiếu phim treo các poster về phim cùng với điểm đánh giá IMDB (rating) và mô tả của khách hàng về bộ phim đó.

Hãy viết hàm `void findMovies (char * path)` giúp khán giả tìm ra phim với điểm đánh giá lớn hơn 8.0 và có mô tả không "nhàm chán" ("*boring*").

Hàm nhận đầu vào là đường dẫn đến tệp chứa các poster phim. Hàm in ra tên những phim thỏa mãn điều kiện trên. Nếu không tìm thấy phim nào phù hợp, in ra thông báo "*All are boring!*".

Tệp có [cấu trúc](#) như sau:

```
id, movie, description, rating
1, Thor, great 3D, 8.0
2, The Rings, Fantasy, 7.5
3, Irish, boring, 6.2
4, Ice song, Fantasy, 8.6
5, Home Alone, Interesting, 6.5
```

For example:

Test	Input	Result
findMovies(path)	1.txt	Thor Ice song

Answer: (penalty regime: 0 %)

1

**Question 9**

Not answered

Mark 0.00 out of 10.00

**[ProductExceptSelf]**

Với một dãy gồm  $n$  số nguyên với  $n > 1$ , giá trị của từng phần tử tại vị trí  $i$  được tính lại bằng tích của tất cả các phần tử còn lại trong dãy số nguyên trừ chính bản thân nó - \(\text{Product of Array Except Self}\)

Viết hàm `void productExceptSelf (char * path)` tính và in ra màn hình dãy số nguyên sau khi được tính lại theo qui tắc trên.

Hàm nhận đầu vào là đường dẫn \(\text{path}\) đến tệp chứa dãy số nguyên.

**Chú ý:** Bạn chỉ phải viết hàm mà không cần viết `main()`, các thư viện cũng đã được khai báo trước.

Tệp có [cấu trúc](#) như sau: Dòng đầu tiên chứa kích thước \(\text{n}\). Dòng tiếp theo chứa \(\text{n}\) số nguyên được ngăn cách bởi dấu cách.

```
5
4 6 2 5 2
```

**For example:**

Test	Input	Result
<code>productExceptSelf(path)</code>	<code>1.txt</code>	<code>-192 -240 320 240 240</code>

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

1	
---	--

**Question 10**

Not answered

Mark 0.00 out of 10.00

**[HouseRobber]**

Bạn là một kẻ trộm chuyên nghiệp và đang lên kế hoạch trộm những ngôi nhà dọc theo một con phố.

Mỗi ngôi nhà có một số tiền nhất định, sự ràng buộc duy nhất ngăn cản bạn lấy trộm là hai nhà liền kề có hệ thống an ninh kết nối và nó sẽ tự động liên lạc với cảnh sát nếu hai nhà liền kề đã bị đột nhập vào cùng một đêm.

Cho danh sách các số nguyên dương biểu diễn cho số tiền của mỗi ngôi nhà, xác định số tiền tối đa bạn có thể trộm tối nay mà không bị báo cảnh sát.

Viết hàm `int getMoney (int * houses, char * path)` để nhập số tiền chứa trong mỗi ngôi nhà từ tệp đầu vào được lưu theo đường dẫn `\( path \)` tương ứng với tên phố bạn định trộm. Số tiền của mỗi nhà được lưu cách nhau bởi một khoảng trắng và không biết trước số lượng nhà trong khu phố đó. Hàm trả về số lượng nhà sau khi bạn đã khởi tạo mảng `\( houses \)`.

Viết hàm `int rob(int * houses, int numOfHouses)` để tính và trả về số tiền lớn nhất mà bạn có thể trộm được từ con phố đó, biết số tiền trong mỗi ngôi nhà được lưu trong mảng `\( houses \)`

**For example:**

Input	Result
Montgomery_Street	23611

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

1	
---	--

[Back to Course](#)