Polar expedition

Submission deadline:	**2021-11-14 23:59:59**
Late submission with malus:	**2022-01-02 23:59:59** (Late submission malus: 100.0000 %)
Evaluation:	**3.0000**
Max. assessment:	**3.0000** (Without bonus points)
Submissions:	5 / 20 Free retries + 10 Penalized retries (-10 % penalty each retry)
Advices:	1 / 2 Advices for free + 2 Advices with a penalty (-10 % penalty each advice)

The task is to implement a function (not a whole program, just a function) which helps planning of a polar expedition.

When planning a polar expedition, the supplies are the main concern. The land is hostile, there are virtually no resources. Assume our vehicle is able to transport fuel, the capacity is `maxLoad` of fuel. Moreover, it needs 1 liter of the fuel per 1 kilometer. We have `fuelTotal` liters of fuel located at the starting point of our expedition. The problem is to compute the maximum distance we can drive and return back. Of course, we can transport some fuel to any position along the path, return back for further supplies, and use the stored fuel later (the fuel is stored in barrels, for instance).

The required interface is:

```
double twoWayDistance ( int fuelTotal, int maxLoad );
```

`fuelTotal`
: is the volume of fuel located at the expedition staring point,

`maxLoad`
: is the volume of fuel our vehicle can transport,

return value
: is the distance of the farthermost point we can travel to and back with the given parameters.

Submit a source file with the implementation of the required function `twoWayDistance`. Further, the source file must include your auxiliary functions which are called from `twoWayDistance`. The function will be called from the testing environment, thus, it is important to adhere to the required interface. Use the sample code below as a basis for your development, complete `twoWayDistance` and add your required auxiliary functions. There is an example `main` with some test in the sample below. These values will be used in the basic test. Please note the header files as well as `main` is nested in a conditional compile block (`#ifdef/#endif`). Please keep these conditional compile block in place. They are present to simplify the development. When compiling on your computer, the headers and `main` will be present as usual. On the other hand, the header and `main` will "disappear" when compiled by Progtest. Thus, your testing `main` will not interfere with the testing environment's `main`.

Your function will be executed in a limited environment. There are limits on both time and memory. The exact limits are shown in the test log of the reference. However, the implementation is very simple and the time/memory limits shall not apply.

Advice:

- Copy the attached source code and use it as a base for your development.
- The main in your program may be modified (e.g., a new test may be included). The conditional compile block must remain, however.
- There is macro `assert` used in the example `main` function. If the value passed to `assert` is nonzero (true), the macro does nothing. On the other hand, if the parameter is zero, the macro stops the execution and reports line, where the test did not match (and shall be fixed). Thus, the program ends silently when your implementation passes the tests correctly.
- Do not forget, we plan two-way expedition.

Sample data: Download

Reference

- **Evaluator: computer**
  - Program compiled
  - Test 'Zakladni test podle ukazky': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.000 s (limit: 5.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test meznich hodnot': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.003 s (limit: 5.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test nahodnymi daty': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.038 s (limit: 4.997 s)
    - Mandatory test success, evaluation: 100.00 %
  - Overall ratio: 100.00 % (= 1.00 * 1.00 * 1.00)
- Total percent: 100.00 %
- Early submission bonus: 0.30

- Total points: 1.00 * (3.00 + 0.30) = 3.30

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**2**	--	--	--
Lines of code:	**27**	**13.50 ± 5.50**	**19**	`twoWayDistance(int,int)`
Cyclomatic complexity:	**6**	**3.00 ± 2.00**	**5**	`twoWayDistance(int,int)`

5 **2021-11-12 16:19:59** **Download**

Submission status: Evaluated
Evaluation: 3.0000

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test with sample input data': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.000 s (limit: 5.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Borderline test': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.001 s (limit: 5.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Random test': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.011 s (limit: 4.999 s)
    - Mandatory test success, evaluation: 100.00 %
  - Overall ratio: 100.00 % (= 1.00 * 1.00 * 1.00)
- Total percent: 100.00 %
- Total points: 1.00 * 3.00 = 3.00

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**3**	--	--	--
Lines of code:	**42**	**14.00 ± 9.09**	**26**	`twoWayDistance`
Cyclomatic complexity:	**9**	**3.00 ± 2.83**	**7**	`twoWayDistance`

4 **2021-11-12 16:17:54** **Download**

Submission status: Evaluated
Evaluation: 0.0000

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test with sample input data': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.000 s (limit: 5.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Borderline test': Abnormal program termination (Segmentation fault/Bus error/Memory limit exceeded/Stack limit exceeded)
    - Total run time: 0.007 s (limit: 5.000 s)
    - Mandatory test failed, evaluation: 0.00 %
  - Overall ratio: 0.00 % (= 1.00 * 0.00)
- Total percent: 0.00 %
- Total points: 0.00 * 3.00 = 0.00

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**3**	--	--	--
Lines of code:	**46**	**15.33 ± 10.87**	**30**	`twoWayDistance`
Cyclomatic complexity:	**10**	**3.33 ± 3.30**	**8**	`twoWayDistance`

3 **2021-11-12 16:16:52** **Download**

Submission status: Evaluated
Evaluation: 0.0000

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test with sample input data': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.000 s (limit: 5.000 s)
    - Mandatory test success, evaluation: 100.00 %

- Test 'Borderline test': Abnormal program termination (Segmentation fault/Bus error/Memory limit exceeded/Stack limit exceeded)
  - Total run time: 0.008 s (limit: 5.000 s)
  - Mandatory test failed, evaluation: 0.00 %
- Overall ratio: 0.00 % (= 1.00 * 0.00)

- Total percent: 0.00 %
- Total points: 0.00 * 3.00 = 0.00

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**3**	--	--	--
Lines of code:	**41**	**13.67 ± 8.65**	**25**	`twoWayDistance`
Cyclomatic complexity:	**10**	**3.33 ± 3.30**	**8**	`twoWayDistance`

2	**2021-11-12 16:15:26**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test with sample input data': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.000 s (limit: 5.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Borderline test': Abnormal program termination (Segmentation fault/Bus error/Memory limit exceeded/Stack limit exceeded)
    - Total run time: 0.007 s (limit: 5.000 s)
    - Mandatory test failed, evaluation: 0.00 %
  - Overall ratio: 0.00 % (= 1.00 * 0.00)
- Total percent: 0.00 %
- Total points: 0.00 * 3.00 = 0.00

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**3**	--	--	--
Lines of code:	**41**	**13.67 ± 8.65**	**25**	`twoWayDistance`
Cyclomatic complexity:	**10**	**3.33 ± 3.30**	**8**	`twoWayDistance`

1	**2021-11-12 16:14:33**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - ☐ Compile in 'pedantic' mode failed (10 % penalty).
  - Test 'Basic test with sample input data': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.000 s (limit: 5.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Borderline test': Abnormal program termination (Segmentation fault/Bus error/Memory limit exceeded/Stack limit exceeded)
    - Total run time: 0.007 s (limit: 5.000 s)
    - Mandatory test failed, evaluation: 0.00 %
  - Overall ratio: 0.00 % (= (1.00 * 0.00) * 0.9)
- Total percent: 0.00 %
- Total points: 0.00 * 3.00 = 0.00

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**3**	--	--	--
Lines of code:	**43**	**14.33 ± 9.53**	**27**	`twoWayDistance`
Cyclomatic complexity:	**10**	**3.33 ± 3.30**	**8**	`twoWayDistance`