SETI

Submission deadline:	**2023-11-06 11:59:59**
Late submission with malus:	**2023-12-31 23:59:59** (Late submission malus: 100.0000 %)
Evaluation:	**6.1111**
Max. assessment:	**5.0000** (Without bonus points)
Submissions:	13 / 20 Free retries + 10 Penalized retries (-10 % penalty each retry)
Advices:	2 / 2 Advices for free + 2 Advices with a penalty (-10 % penalty each advice)

The task is to develop a program to analyze messages from alien civilizations.

Radiastronomes have finally received messages from alien civilizations. Finally! The messages are broadcasted from stars that a
There are no doubts!

The problem is to decode the messages. Surprisingly, all messages are apparently coded using the same code (a variant of extr
Moreover, each alien star broadcasts the same message over and over. One question is: is it possible for all messages to synchr
an time where we receive all the messages from the beginning?

The input of the program is formed by two or more messages. Each message is passed as a single input line. The line is formec
characters, each character codes the length of a pulse: a=1, b=2, c=4, d=8, ... Next, each line contains one character |, the ch
beginning of the message. For example:

```
ea|babab
```

represent a message with pulses of length 16, 1, 2, 1, 2, 1, and 2. The length of the message is 16 + 1 + 2 + 1 + 2 + 1 + 2=
current position in the message is 16+1=17 time units before the end.

The program is given the messages, message lengths and positions may vary. The program computes when the messages sync
start receiving all messages from the beginning. For instance:

```
ea|babab
d|abaca
```

will be synchronized in 42 time units:

```
ea|   bababea|           = 42
d|    abacad|abacad|     = 42
```

The input of the program consists of the messages as described above. There are at least two messages in the input. The man
only pass two input messages, more than two messages are tested in the bonus tests.

The output of the program is the time when the input messages synchronize. If the input messages never synchronize, a differ
sample runs below).

The program must validate input data. If the input is invalid, the program must detect it, it shall output an error message (see
displayed, the error message must be displayed to the standard output (do not send it to the error output) and the error messa
by a newline (\n). The input is considered invalid, if:

- less than two messages in the input,
- invalid message format (message must not be empty, it must consist of lower case characters only, there must be exact
message).

Please strictly adhere to the format of the output. The output must exactly match the output of the reference program. The cor
machine, the machine requires an exact match. If your program provides output different from the reference, the program is co
Be very careful, the machine is sensitive event to whitespace characters (spaces, newlines, tabulators). Please note that all outp
a newline character (\n). This applies even to the last line of the output, moreover, this applies even to the error message. Dow
archive. The archive contains a set of testing inputs and the expected outputs. Read Progtest FAQ to learn how to use input/ou
to simplify testing of your programs.

Example program runs:

```
Messages:
ea|babab
d|abaca
```

```
Synchronized in: 42
```

```
Messages:
acacbbaaa|
cbbcabcbbaa|
Synchronized in: 0
```

```
Messages:
|acabbaaa
|dcbabaaaa
Synchronized in: 0
```

```
Messages:
acacbbaa|baa
bcaabaa|baaa
Synchronized in: 216
```

```
Messages:
cbbcbaa|aaaa
aadaaaabaaa|abaa
Never synchronizes.
```

```
Messages:
dbdccdccbbbbaa|aaaa
bebbaedbcabbaa|ccbaa
Synchronized in: 276
```

```
Messages:
jfddiaggfdffaagfacafcfbbbgdcddbfaedfeaacfeadcbabbaaa|hdccdfdbfadcdbbcbbaa
bhfdigdbggfadfdddacceaedbefeacbacaacbbbaa|cdcgebfichfbgcbjbhdfadgheifheceeehiceeedfbgcgaggfda
Synchronized in: 5046543
```

```
Messages:
acacbb|aaa
cbbcabcbbaa
Invalid input.
```

```
Messages:
acacbb|aaa
cbbc abcb|baa
Invalid input.
```

Advice:

- The example runs above list both the output of your program (boldface font) and user input (regular font). The bold/re included here, in the problem statement page, to increase readability of the listing. Your program must output the text markup.
- Do not forget the newline (\n) after the last output line.
- The program can be developed without additional functions (i.e., in one big `main`). However, if divided into functions, t and easier to debug.
- The problem does not require any arrays. When used, the arrays just make the problem more difficult. The messages r aggregated on-the-fly.
- Use `scanf` or `fgetc` functions to read the input (scanf format `%c`).
- Basic naive solution does not pass all tests. There are bonus tests that require an advanced algorithm or an efficient im
    - the first bonus will be awarded to solutions that correctly handle more than two input messages,
    - the second bonus will be awarded to solutions that are time optimal - can efficiently process long messages,
    - the third bonus will be awarded to solutions that are time optimal - can efficiently process very long messages.
- All computations may be done in the standard `int` data type. The only exceptions are the second and third bonus test. messages, thus the program needs to handle big numbers. We recommend using `long int` or `long long int` data program that is intended to pass the second and third bonus. Some algorithms may need to multiply that big numbers, temporary variables with more than 64 bits. Your implementation may need to use non-standard type `__int128_t`. Th gcc, there is no standard input/output conversion for this data type, however, the type may be used in common arithme

Sample data:	
Submit:	Browse... **No file selected.**

☑ Reference

- **Evaluator: computer**
  - Program compiled
  - Test 'Základní test s parametry podle ukázky': success
    - result: 100.00 %, required: 100.00 %
    - Max. run time: 0.007 s (limit: 1.000 s)
    - Total run time: 0.059 s
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test mezních hodnot': success
    - result: 100.00 %, required: 20.00 %
    - Max. run time: 0.006 s (limit: 1.000 s)
    - Total run time: 0.055 s
    - Optional test success, evaluation: 100.00 %
  - Test 'Test ošetření vstupních dat': success
    - result: 100.00 %, required: 20.00 %
    - Max. run time: 0.007 s (limit: 1.000 s)
    - Total run time: 0.106 s
    - Optional test success, evaluation: 100.00 %
  - Test 'Test náhodnými daty': success
    - result: 100.00 %, required: 20.00 %
    - Max. run time: 0.006 s (limit: 1.000 s)
    - Total run time: 0.123 s
    - Optional test success, evaluation: 100.00 %
  - Test 'Bonus 1: více vstupních zpráv': success
    - result: 100.00 %, required: 100.00 %
    - Max. run time: 0.009 s (limit: 1.000 s)
    - Total run time: 0.137 s
    - Bonus test - success, evaluation: 115.00 %
  - Test 'Bonus 2: efektivita': success
    - result: 100.00 %, required: 100.00 %
    - Max. run time: 0.006 s (limit: 0.500 s)
    - Total run time: 0.062 s
    - Bonus test - success, evaluation: 120.00 %
  - Test 'Bonus 3: efektivita': success
    - result: 100.00 %, required: 100.00 %
    - Max. run time: 0.006 s (limit: 0.250 s)
    - Total run time: 0.063 s
    - Bonus test - success, evaluation: 125.00 %
  - Overall ratio: 172.50 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.15 * 1.20 * 1.25)
- Total percent: 172.50 %
- Early submission bonus: 0.50
- Total points: 1.73 * (5.00 + 0.50) = 9.49

SW metrics:

	Total	Average	Maximum	Function name
Functions:	**6**	--	--	--
Lines of code:	**145**	**24.17 ± 10.43**	**38**	`main`
Cyclomatic complexity:	**30**	**5.00 ± 3.27**	**10**	`processLine`

13	**2023-10-24 20:26:13**	**Download**
Submission status:	Evaluated	
Evaluation:	6.1111	

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test with sample input data': success

        - result: 100.00 %, required: 100.00 %
        - Max. run time: 0.006 s (limit: 1.000 s)
        - Total run time: 0.055 s
        - Mandatory test success, evaluation: 100.00 %
    - Test 'Borderline test': success
        - result: 88.89 %, required: 20.00 %
        - Max. run time: 0.006 s (limit: 1.000 s)
        - Total run time: 0.055 s
        - Optional test success, evaluation: 88.89 %
        - Failed (invalid output) **[Unlock advice (64 B)]**
    - Test 'Invalid input test': success
        - result: 100.00 %, required: 20.00 %
        - Max. run time: 0.011 s (limit: 1.000 s)
        - Total run time: 0.111 s
        - Optional test success, evaluation: 100.00 %
    - Test 'Random test': success
        - result: 100.00 %, required: 20.00 %
        - Max. run time: 0.007 s (limit: 1.000 s)
        - Total run time: 0.125 s
        - Optional test success, evaluation: 100.00 %
    - Test 'Bonus 1: more input messages': failed
        - result: 10.00 %, required: 100.00 %
        - Max. run time: 0.007 s (limit: 1.000 s)
        - Total run time: 0.126 s
        - Bonus test - failed, evaluation: No bonus awarded
        - Failed (invalid output) **[Unlock advice (120 B)]**
        - Failed (invalid output) **[Unlock advice (111 B)]**
        - Failed (invalid output) **[Unlock advice (178 B)]**
        - Failed (invalid output) **[Unlock advice (118 B)]**
        - Failed (invalid output) **[Unlock advice (129 B)]**
        - Failed (invalid output) **[Unlock advice (109 B)]**
        - Failed (invalid output) **[Unlock advice (155 B)]**
        - Failed (invalid output) **[Unlock advice (150 B)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
        - Failed (invalid output) **[Unlock advice (2.06 KiB, incomplete data)]**
    - Test 'Bonus 2: speed test': failed
        - result: 0.00 %, required: 100.00 %
        - Max. run time: 0.006 s (limit: 0.500 s)
        - Total run time: 0.063 s
        - Bonus test - failed, evaluation: No bonus awarded
        - Failed (invalid output) **[Unlock advice (841 B)]**
        - Failed (invalid output) **[Unlock advice (841 B)]**
        - Failed (invalid output) **[Unlock advice (968 B)]**
        - Failed (invalid output) **[Unlock advice (747 B)]**
        - Failed (invalid output) **[Unlock advice (949 B)]**
        - Failed (invalid output) **[Unlock advice (795 B)]**
        - Failed (invalid output) **[Unlock advice (904 B)]**
        - Failed (invalid output) **[Unlock advice (945 B)]**
        - Failed (invalid output) **[Unlock advice (751 B)]**
        - Failed (invalid output) **[Unlock advice (811 B)]**
    - Test 'Bonus 3: speed test': success
        - result: 100.00 %, required: 100.00 %
        - Max. run time: 0.007 s (limit: 0.250 s)
        - Total run time: 0.063 s
        - Bonus test - success, evaluation: 125.00 %
    - Overall ratio: 111.11 % (= 1.00 * 0.89 * 1.00 * 1.00 * 1.25)
- Advices used: 2
- Penalty due to advices: None (2 <= 2 limit)

- Total percent: 111.11 %
- Early submission bonus: 0.50
- Total points: 1.11 * (5.00 + 0.50) = 6.11

		Total	Average	Maximum	Function name
SW metrics:	Functions:	**4**	--	--	--
	Lines of code:	**106**	**26.50 ± 29.24**	**77**	`main`
	Cyclomatic complexity:	**31**	**7.75 ± 9.96**	**25**	`main`

12	**2023-10-24 20:24:33**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test with sample input data': failed
    - result: 22.22 %, required: 100.00 %
    - Max. run time: 0.007 s (limit: 1.000 s)
    - Total run time: 0.058 s
    - Mandatory test failed, evaluation: 0.00 %
    - Failed (invalid output) **[Unlock advice (72 B)]**
    - Failed (invalid output) **[Unlock advice (78 B)]**
    - Failed (invalid output) **[Unlock advice (75 B)]**
    - Failed (invalid output) **[Unlock advice (82 B)]**
    - Failed (invalid output) **[Unlock advice (85 B)]**
    - Failed (invalid output) **[Unlock advice (97 B)]**
    - Failed (invalid output) **[Unlock advice (247 B)]**
  - Overall ratio: 0.00 %
- Advices used: 2
- Penalty due to advices: None (2 <= 2 limit)
- Total percent: 0.00 %
- Early submission bonus: 0.50
- Total points: 0.00 * (5.00 + 0.50) = 0.00

		Total	Average	Maximum	Function name
SW metrics:	Functions:	**4**	--	--	--
	Lines of code:	**106**	**26.50 ± 29.24**	**77**	`main`
	Cyclomatic complexity:	**32**	**8.00 ± 10.39**	**26**	`main`