Midterm Imperative Programming
Oct. 3 2022, 18:30-21:30h

- You can solve the problems in any order. Solutions must be submitted to the automated judgement system Themis. For each problem, Themis will test ten different inputs, and check whether the outputs are correct.
- Grading: you get one grade point for free. The remaining nine points are based solely on the judgment given by Themis. The first problem is worth one grade point. The remaining four problems are worth two grade points each, of which you score the full two points if you passed the complete test set of the problem (i.e. 10 test cases), or one grade point if you passed at least 5 (out of 10) test cases.
- Inefficient programs may be rejected by Themis. In such cases, the error will be 'time limit exceeded'. The time limit for each problem is two seconds.
- The number of submissions to Themis is unlimited. No points are subtracted for multiple submissions.
- There will be no assessment of programming style. However, accepted solutions are checked manually for cheating: for example, precomputed answers will not be accepted, even though Themis accepts them.
- Note the hints that Themis gives when your program fails a test.
- Needless to say: you are not allowed to work together. If plagiarism is detected, both parties (supplier of the code and the person that sends in copied code) will be excluded from any further participation in the course.
- You are not allowed to use email, phones, tablets, calculators, etc. There is a calculator available on the exam computers (see icon on the desktop). You are allowed to consult the ANSI C book and a dictionary. You are not allowed to use a printed copy of the reader or the lecture slides, however they are available digitally (as a pdf) in Themis. You are allowed to access your own submissions previously made to Themis.
- For each problem, the first three test cases (input files) are available on Themis. These input files, and the corresponding output files, are called `1.in`, `2.in`, `3.in`, `1.out`, `2.out` and `3.out`. These files can be used to test whether the output of your program matches the requested layout, so that there can be no misunderstanding about the layout and spaces in the output.
- **If you fail to pass a problem for a specific test case, then you are advised not to lose much time on debugging your program, and continue with another problem. In the last hour of the midterm, all input files will be made visible in Themis (not the output files).**

Problem 1: Time Difference

With the time notation 15:20:15 we mean 15 hours, 20 minutes and 15 seconds. Clearly, 16:21:16 is later. The time difference is 01:01:01, so one hour, one minute and one second. Note that in time notation, a number less than 10 is prefixed with a leading zero.

The input of this problem are two moments in time notation, and the output should be the difference in time notation. Note that the two moments in time are on the same day (same date), so the difference in time cannot exceed 24 hours. Moreover, the times need not be in chronological order (see examples 1 and 2).

Example 1:

input:
```
15:20:15 16:21:16
```
output:
```
01:01:01
```

Example 2:

input:
```
16:21:16 15:20:15
```
output:
```
01:01:01
```

Example 3:

input:
```
12:56:38 23:32:45
```
output:
```
10:36:07
```

Problem 2: Coalitions

The input of this problem is a series of letters (which can be 'A', 'B', or 'C') that is terminated by a newline. The series represents the votes in an election process with three parties that are (conveniently) named A, B, and C. The output should be the number of possible coalitions that have the majority of votes.

For example, if the input is ABBACBBA, then the total number of votes is 8. So, a coalition that has a majority must have at least 5 votes. The votes are distributed as follows: party A got 3 votes, party B got 4 votes, and party C got 1 vote. So, the following three coalitions have a majority: ABC (3+4+1=8 votes), AB (3+4=7 votes), BC (4+1=5 votes). Any other combination has less than 5 votes.

The output of you program must be, given the votes on the input, the number of possible coalitions that have a majority.

Example 1:

input:
```
ABBACBBA
```
output:
```
3
```

Example 2:

input:
```
BCAACBCBBB
```
output:
```
3
```

Example 3:

input:
```
ABBCBCAABCAABBC
```
output:
```
4
```

Problem 3: Perfect Product

A positive integer n is called a *perfect square* if there exists an integer m such that $n = m^2$. The first 10 perfect squares are 1, 4, 9, 16, 25, 36, 49, 64, 81, and 100.

The input for this problem is a positive `int` a. The output must be the smallest integer b such that $n = a \times b$ is a *perfect square*.

Example 1:

input:
```
42
```
output:
```
42
```

Example 2:

input:
```
97
```
output:
```
97
```

Example 3:

input:
```
12
```
output:
```
3
```

Problem 4: Cuban Primes

A *Cuban prime* is a prime number p for which there exists a positive integer q such that $p = (q+1)^3 - q^3$. For example, $p = 19$ is a prime number and using $q = 2$ it can be written as $p = (q+1)^3 - q^3 = 27 - 8 = 19$.

The input for this problem is an integer n, where $1 \leq n < 500000000 = 5 \times 10^8$. The output of your program must be `YES` if n is a Cuban prime, and `NO` otherwise.

Example 1:

input:
```
19
```
output:
```
YES
```

Example 2:

input:
```
201
```
output:
```
NO
```

Example 3:

input:
```
997
```
output:
```
NO
```

Problem 5: Curzon Numbers

A number n is called a *Curzon Number* if $2^n + 1$ is divisible by $2 \cdot n + 1$.
For example, $n = 2$ is a Curzon number because $2^2 + 1$ and $2 \cdot 2 + 1$ both evaluate to 5.
Also, $n = 6$ is a Curzon number, because $2^6 + 1 = 64 + 1 = 65$ and $2 \cdot 6 + 1 = 13$, and $5 \times 13 = 65$.

The input for this problem is an integer n (where $0 \leq n < 5000$). The output of your program must be `YES` if n is a Curzon number, and `NO` otherwise.

Example 1:

input:
```
2
```
output:
```
YES
```

Example 2:

input:
```
6
```
output:
```
YES
```

Example 3:

input:
```
7
```
output:
```
NO
```