**Kolmogorov Complexity**

**Background**

In this project, you need to deal with a problem related to Kolmogorov complexity. Kolmogorov is a great mathematician who defined the algorithmic descriptive complexity of an object to be the length of the shortest binary computer program that describes that object. Kolmogorov complexity KU(x) of a string x with respect to a universal computer U is defined as

,

the minimum length over all computer programs that print x and halt. Thus, KU(x) is the shortest description length of x over all descriptions interpreted by U.

**Description**

Kolmogorov also made a crucial observation that the description length of the object is essentially independent to the computer. However, It is proved that Kolmogorov complexity is not computable. In other words, there is not a computer program that takes a string x as an input and outputs KU(x). Anyway, it is still possible to write short programs to estimate the Kolmogorov complexity, which shows an upper bound of KU(x). In the task, you should write C++ programs that output certain strings. There is a string that only contains ‘0’ and ‘1’ in each input file. The length of your programs should be as short as possible.

**Instructions**

You must follow several instructions when you write the programs.

1. You should write one program for each input string.
2. You should use C++ programming language on Windows platforms.
3. Your program is compiled by “mingw32-g++.exe –o prog prog.cpp” from [MinGW](https://sourceforge.net/projects/mingw-w64/).
4. Your program should output the string given in the input file.
5. Your program should halt within 5 seconds.
6. The length of your program should be as short as possible.
7. You may use auxiliary files but the length of these files is also taken into account.

**Performance Evaluation**

The length of your programs (including auxiliary files used) is evaluated in bytes. For each test case, you can get at most 10 points, i.e., the student writes the shortest program will get 10 points. Other students’ grades are given as

where lm is the length of the shortest program. For example, if there are 40 students in the class and the best program’s length is 100 bytes, the grades of programs of different length are shown in the following table.

Table I

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Example 1 | | | Example 2 | | |
| Program’s length | Student’s rank | Student’s grade | Program’s length | Student’s rank | Student’s grade |
| 100 Bytes | 1 | 10 | 100 Bytes | 1 | 10 |
| 210 Bytes | 2 | 9.5 | 110 Bytes | 2 | 9.5 |
| 250 Bytes | 3 | 9 | 120 Bytes | 3 | 9 |
| 300 Bytes | 4 | 8.5 | 125 Bytes | 4 | 8.5 |
| …… | | | …… | | |
| 500 Bytes | 17 | 2 | 170 Bytes | 10 | 5.5 |
| 550 Bytes | 18 | 1.5 | 180 Bytes | 11 | 5 |
| 800 Bytes | 20 | 1 | 190 Bytes | 12 | 5 |
| 900 Bytes | 21 | 1 | 200 Bytes | 13 | 5 |
| …… | | | …… | | |

**Project Schedule**

You are also asked to introduce your analyses and methods to TAs and then write a report, and some students writing effective programs are also invited to do a presentation in the class. The final grade of the project is evaluated according to all three parts, including the interview (presentation), the report and the performance of your programs.

Table II

|  |  |  |
| --- | --- | --- |
| Project schedule | | |
| Item | | Date |
| Project introduction | | 2017-10-18 |
| Releasing test cases | | Before 2017-10-19 |
| Presentation | | 2017-12-27, 2018-01-03 |
| Interview  (AFTER submission) | | 2017-12-28, 2017-12-29  2018-01-04, 2018-01-05 |
| Writing a report | | Before 2018-01-15 |
| Program submission | Test case 01-05 | Before 2017-12-25 |
| Test case 06-10 | Before 2018-01-01 |