

Software Project - Assignment 2

Programming for Engineers 2021/22 - MSc

[50% of total module marks]

Overview

This is the second assignment for Programming for Engineers MSc module. Based on what you have learnt in the seminars, we expect you to be able to finish this assignment and submit your implementation and report. In the process of designing this assignment, we also considered the time requirements, to make sure you will be able to complete all the tasks.

Problem Description

Imagine that you are given an essay that you should mark, BUT the marking criteria are tedious and frustrating to go through. As a student who is familiar with programming, you should be able to program the marking system.

We have a pointing system for our letters in the marking as follows:

Table 1: Letters with the corresponding points and types

| Letter | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|--------|----|----|---|---|----|----|---|---|---|----|----|---|---|----|---|----|----|---|----|----|----|----|----|----|----|----|
| Point | 10 | 21 | 7 | 2 | 19 | 18 | 4 | 8 | 6 | 11 | 14 | 3 | 9 | 22 | 1 | 23 | 17 | 5 | 25 | 13 | 12 | 26 | 24 | 16 | 20 | 15 |
| Type | V | C | C | C | V | C | C | C | V | C | C | C | C | C | V | C | C | C | C | C | V | C | C | C | C | C |

We also have the following **criteria** for the essay:

1. The shortest word should get between 10 and 50 points (count the points based on Table 1) (you should exclude one-letter words).
2. The longest word should get between 200 and 250 points.
3. There should be 5 or more different palindrome words in the essay (i.e. words reading the same backward as forward, e.g. "stats").
4. The length of the essay should be between 50 and 100 words.
5. The average number of letters used in a word should be between 4 and 8, and the average number of points (see Table 1) for the words used in the essay should be between 20 and 60.
6. It is not allowed to have words that have more than 4 letters of type V (find the type for each letter in Table 1).

Software Project - Assignment 2

Programming for Engineers 2021/22 - MSc

[50% of total module marks]

Also, there is a secret in the essay! The problem is that it's enciphered with a key. So, we need to follow these steps:

1. **Calculate the key:** the key for the essay is the remainder of dividing the number of points for student's name (which is exactly the 10th word of the essay) by 26, e.g. if the 10th word is John ($11 + 1 + 8 + 22 = 42$), then the key is: $42 \bmod 26 = 16$.
2. **Find the secret:** secret is the second word the number points (see Table 1) for which lies between 90 and 110. If there is no such a word, then the secret is the last word in the essay.
3. **Decipher the secret:** use the Caesar cipher and the key that you have already found to decipher the secret.

Note: Don't worry, neither the secret, nor the key (and the student's name) might make any sense in the examples (inputs/outputs) provided along with the project.

Task

Now your task is to find the followings:

1. For each of the six criteria mentioned above, print out a statement of whether it passes or fails; you do not need to explain why, just print 'pass' or 'fail' for each criterion.
2. Print out the shortest word, the longest word, the word with the lowest number of points, the word with the highest number of points, the least frequent word and the most frequent word.
3. Print out the number of times each word appears in the essay in an ascending order.
4. Print out the student's name, the key, the secret of the essay, and the deciphered version of the secret.
5. Sort all words by their points and print out the one in the middle (in case the number of words is even, concatenate the two in the middle and print them out as one word without any spaces in between)

Marking criteria: Program

Part of the marking of your program will be done using an automated marking system. Therefore it is your responsibility to make sure that your code follows the format that is explained in this document, otherwise, even if your program produces the correct output on your side, your mark may be reduced. I will check all code by hand as well, so partial credit will be given as per usual for any code which does not fully solve the given problems.

Software Project - Assignment 2

Programming for Engineers 2021/22 - MSc

[50% of total module marks]

You should implement all the classes, structs, and functions that are provided inside the code (**definitions.h** file). They will all be tested and removing them or leaving them empty will surely result in a reduced mark. Feel free to add any other functions etc. which you find useful.

We have generated a rich set of inputs/outputs that will cover all the different parts of your code and will check all the boundaries that you should consider. However, you will be given 10% of the test cases that will be used in the automated marking system, so that you can assess if your program satisfies them. Please make sure that your program satisfies all of them.

Your final mark for your implementation would be based on the following scores:

- Your program works and produces the correct output, **and you have explained your algorithms in your report**: 50%
 - Task 1: 12%
 - Task 2: 15%
 - Task 3: 10%
 - Task 4: 8%
 - Task 5: 5%
- Functions *criterion1* to *criterion6* make 4.5% each (totaling 27%)
- Caesar decipher function: 2%
- isPalindrome function: 2%
- calPoints function: 3%
- createTable function: 3%
- Class Word: 5%
- struct letter: 2%
- readInput function: 3%
- writeOutput function: 3%

There is also a minimal version of the marking system provided for you that will help you to figure out if your code is able to pass each of the tasks. Its use will be explained and demonstrated in a Panopto recording on the module's Canvas site.

Software Project - Assignment 2

Programming for Engineers 2021/22 - MSc

[50% of total module marks]

When you run the test program, the result will be a file called 'finalMark.txt'. This file will provide you with an indication of the degree to which the output of your program is correct. The percentages for a full mark break down as follows:

- Task 1: 24%
- Task 2: 30%
- Task 3: 20%
- Task 4: 16%
- Task 5: 10%

Marking criteria: Report

Your report is a very important element of the assessment, and is where you will need to demonstrate understanding of the code which you have submitted. There is a nominal word limit of 1500 words (excluding the Appendix), but you can feel free to write more or less than this - the main thing to focus on is how well you describe your algorithms. There is no need to submit any code from standard libraries. If any non-standard external libraries were used, please provide their files inside your project and import them locally from that file, otherwise using them will not be acceptable.

Your report should contain the following parts:

1. Title page - Your candidate number and the title of your report / project.
2. Introduction - A brief explanation of how the complete program works in principle.
3. Program description - An explanation of how the various elements (e.g. functions, structs, classes) of your program work. As indicated above, your marks for Tasks 1 to 5 will be affected by how well you explain your implementation. Where appropriate, you should use pseudocode and/or flowcharts to describe your algorithms. **No program code should appear in your report except in Appendices.**
4. Conclusion: A brief reflection on how successful your implementation is, as well as how you might add to or improve it if you had more time.
5. Appendix (source code as text, if and only if you needed it for referencing from section 3, otherwise you do not need to have the code in the report)

Software Project - Assignment 2

Programming for Engineers 2021/22 - MSc

[50% of total module marks]

Program format

You are provided with a Dev-C++ project, with two files that you should write all of your code in. You can use external libraries only if you provide them alongside your submission files. Please note that changing the signature of any of the predefined functions might result in them not being recognised by the marking system, which will reduce your mark.

Main files should follow these criteria:

- **main.cpp** *must only contain the main function*. You cannot put any auxiliary code (i.e. code outside of the main function) inside this file
 - Make sure that you do not change the position of the separators (in C, every expression is separated using white space character/s, statements are separated using semicolons)
 - Make sure that you add ‘,’ after every part of the output (each part of the task). Take a look at the provided test cases to have a better understanding of how you should print the outputs
 - For instance, you need to put a ‘,’ after producing the result of each criterion (see Page 1 of this assignment), BUT not after criterion 6
- **definitions.h** is where you should implement *all* of the code that you want to have outside of the main function. Inside this file, we have the following parts that you should complete:
 - A class called ‘Word’ which contains the word, its point (based on Table 1) and the number of times that it appeared in the essay
 - A struct called ‘letter’ which contains the letter, its point and its type
 - Two functions to work with files called ‘readInput’ and ‘writeOutput’ for reading from and writing into files respectively
 - A function called ‘calPoints’ to calculate the number of points for a word
 - A function called ‘isPalindrome’ to check if a string is palindrome
 - A function called ‘createTable’ to fill in the array of letters (from ‘struct letter’) with the data from Table 1
 - A function called ‘caesarDecipher’ to decipher the text from Caesar cipher
 - 6 different functions called ‘criterion1’, ‘criterion2’, ‘criterion3’, ‘criterion4’, ‘criterion5’ and ‘criterion6’ which check if the input essay passes each of these criteria

Software Project - Assignment 2
Programming for Engineers 2021/22 - MSc
[50% of total module marks]

Submission Requirements

Your submission should have the following parts:

- **Program**
 - **main.cpp** file which only contains a main function that runs the program, gets the name of input file and output file, reads the input and prints out the output
 - **definitions.h** which is the auxiliary file for you to implement all the functions, classes, structs, etc. that you need
- **Report**
 - A single pdf file containing your report

You should submit a single zip file containing all these files on Canvas.

Please keep in mind that this is an individual assessment; the related University rules on plagiarism apply, and **here in particular with respect to collusion with fellow students and plagiarising code from elsewhere** (e.g. the web, software repositories, etc.).

Submission Deadline

The assignment is due on **Thursday, 13th January, at 4pm. Please check Sussex Direct to confirm your exact deadline.**