Exercise Sheet Frequency Analysis

Solve the following exercises and submit them until the communicated date.
Background: You intercepted the following encrypted message:

```
QY QWN UIG JMTV BAAA FWALSOX, QWN ZIOW NHMVW LPX UWKJMVL SXQ
NHJ BAW DBYMGWZX UQIZMK. OM AGXX LPTL GHM VXNMK MAX LPBK KBHPXJ
NHJ ATXMMQ-KKABBUIE SXIDQVSBBGVL :).

AV HJLXJ BH KQFHTBXG MZM YJMJMMGUG TFIEQABK ABYVBXQVSVMDG, MZM
YAZLL NXO AXFBXFKXK NKGU WSVBWT WWNHW'A KGJBFAHF KKMAHW NHDTHO:

Q PSA UGZG AV MZM RWIK 1632, AV MZM VABR GN RGZD, GN T YWHV
NTEQEQ, BAGCZZ VHL WY LPTL KHMVMJG, FQ NTLPXJ JXAVZ S NHJMBYVXJ
WY TZXEMG, OPH KMMLTXV NBJAM SB AMTE. ZM ZGB T YWHV MLLIMW JR
EMKUPTFLBKM, TFL EWIOAVZ GNY ZQL LZTVM EADXV IYLMKOIKV IM QWKC,
NKGU PZMGUM AW PTV UTJZBWL FQ UHLPXJ, EAGAX JMESBBGVL OMKW
VTEMW JWUAVLGV, T YWHV NTEQEQ QG LPTL KHMVMJG, TFL YJWF OPHE Q
PSA VSTEWL KGJBFAHF SKWCMRVXSZ; UMB UQ BAW CLMIE UWKJCILQHF WY
OWKVA BF MGYTTFL PW IKW VHO KTDTXV, VTQ, EX UIED WNJAXDDXK, IGV
EKABX GCK FIFW, KKMAHW, IGV AH EG VGUISVBGVL STPSGL UIEDMW EM.
```

You know that the message has been written in English.

LB-FA 00. (<u>not</u> to be submitted)

a) Discuss which simple cipher could have been used to encrypt the above message. Which clues are given by the structure of the words and the used character set?

b) Specify which methods of analysis can be used to determine the most probable key for the used cipher.

<u>Placeholder for the solution</u>:

Result of the _____ analysis:

Further analysis (_____):

The key length is _____.

LB-FA 01. (*Code::Blocks* template *Small cryptographic project*)

In order to determine the most probable key, the above message is to be split into three parts – one per key position. Write a program which accepts the path to a text file as a (command line) argument and which performs a frequency analysis of the characters `'A'...'Z'` contained in the file for each key position. The program should print these characters and their corresponding **relative** frequencies (relative **only** to the characters per key position to be analyzed) column-wise per key position to `std::cout`. Each column must be **stably** sorted in descending order. The frequencies of all characters are to be rounded to the nearest integers and to be output separated by line breaks **exactly** (i.e., including spaces and separators) as follows (the numerical values are only examples):

```
E: 13% ; F: 14% ; D: 13%
T: 9% ; U: 9% ; S: 10%
A: 8% ; B: 8% ; Z: 7%
[..]
```

Note: `[..]` is to be understood as a placeholder for the remaining lines and is **not** part of the output. The first column corresponds to the frequency distribution of letters in the English language (without encryption).
Hint: Define an appropriate data structure to store the characters and their relative frequencies. Use `std::ifstream` (requires `<fstream>`) to read the file and the `>>` operator to read single characters. Use `std::sort` (requires `<algorithm>`) to sort the array with an appropriate custom comparison function. `round` (requires `<cmath>`) can be used for rounding. The precision of the output via `std::cout` can be set by `std::fixed` and `std::setprecision` (requires `<iomanip>`). For the frequency analysis, the message should not be split into three files, but the affiliation to the key position should be determined based on the character position in the file.

LB-FA 02. (*Code::Blocks* template *Small cryptographic project*)

The Caesar cipher is defined as follows: A plaintext character m (represented as a number between 0 and 25) is encrypted by a key character k (represented as a number between 0 and 25) to a character c through shifting.

$$c = (m + k) \mod 26$$

The character `'A'` corresponds to the number 0, the character `'B'` corresponds to the number 1 etc. Analogously to encryption, the character c can be decrypted to m through reverse shifting by k.

$$m = (c - k) \mod 26$$

The Vigenère cipher corresponds to a cyclical application of the Caesar cipher per position with a key of l characters in length:

$$c_i = (m_i + k_{i \mod l}) \mod 26$$

This means that, for example, for a key length of three characters, the fourth plaintext character is encrypted in turn with the first key character.

Write a program which accepts an input file path, an output file path and a key as arguments (in exactly this order!) and that performs a decryption of the input file into the output file using the Vigenère cipher. Characters other than `'A'...'Z'` should **not** be decrypted, but transferred unmodified into the output file. Verify the correctness of your program with the above message and the key that you can determine as follows:
Use the result of the frequency analysis from exercise LB-FA 01. in order to compute the most probable key character by character. Each column corresponds to one key position. Determine the offset between the frequency distribution of the letters in the English language (see above) and the found frequency distribution. The offset corresponds to the respective key character. In the example output for LB-FA 01. above, the key would be `'ABZ'` as the offset in the first column is 0 (corresponds to `'A'`), that of the second column is 1 (corresponds to `'B'`) and that of the third column is $-1 \equiv 25 \pmod{26}$ corresponds to `'Z'`).
Hint: In addition to the file input functions utilized in LB-FA 01., use `std::ofstream` *(requires* `<fstream>`*) to output the file and the* `<<` *operator to write single characters. Before decrypting, ensure that the key only contains characters within the specified value range.*