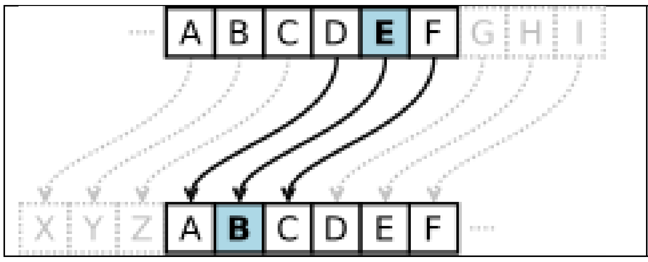
Computer Architectures **02GOLOV**	Delivery date: 3/12/2020
Laboratory **7**	Expected delivery of lab_07.zip must include: - the startup.s files for exercises 1 and 2 - this document compiled possibly in pdf format.

Starting from the *ASM_template* project, solve the following 2 exercises.

Exercise 1) In Cryptography, a **Caesar cipher** is one of the simplest encryption techniques. Using this cipher, each letter in the **plaintext** is replaced by a letter some fixed number of positions down the alphabet. The encryption **key** is the number of positions to be added to the plaintext (and subtracted from the ciphertext), in modular form (after Z, you can continue with A).
The transformation can be represented by aligning two alphabets; the cipher alphabet is the plain alphabet rotated left or right by some number of positions.



```
Plain:    ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:   XYZABCDEFGHIJKLMNOPQRSTUVW
```

Frequency analysis can be used to break classical ciphers. This analysis is based on the fact that, in any given stretch of written language, certain letters and combinations of letters occur with varying frequencies. Moreover, there is a characteristic distribution of letters that is roughly the same for almost all samples of that language. Considering the Italian language, the letters with the highest frequencies are the following:
E: 11.79%   A: 11.74%   I: 11.28%   O: 9.83%   N: 6.88%

An encrypted message can be memorized as a string of bytes terminated by NULL (or '\0') in the code section (as a part of the code itself) or in a read-only data section, as in the following example:

```
Ciphertext DCB "PBOAEOXDKXNYSVMYBCYNSKBMRSDODDEB",
           DCB "OOCSCDOWSNSOVKLYBKJSYXORYMKZSDYM",
           DCB "ROSVMSPBKBSYNSMOCKBOCSBYWZOPKMSV",
           DCB "WOXDOZOBAEOCDYWYDSFYOFSDOBYNSECK",
           DCB "BVYZOBZBYDOQQOBOSWSOSNKDSCOXCSLS",
           DCB "VS", 0
```

Considering the example above, write an assembly program that is able to identify the **most frequent letter** in the message. Assume that the letters are all uppercase, no whitespace, commas or numbers. Please also respond to questions in the following box.

Report your reasoning for both questions

Q1: Can you guess the encryption key by comparing the most frequent letter in your message and in the Italian language?

Sì, la lettera più frequente del ciphertext è la 'O'. La lettera più frequente in italiano è la 'E'. Il messaggio in chiaro si può ottenere sottraendo in modo modulare a ogni lettera del ciphertext la distanza tra 'O' ed 'E', ossia scalando ogni lettera indietro di 10 posizioni.

Q2: Can you use the same strategy to find the key if the message is composed by the first 32 characters? And with a much longer message?

Con soli 32 caratteri la lettera più frequente sarebbe la B, che è la lettera sbagliata da usare come chiave. Se si decodifica il ciphertext usando come chiave la B si ottiene un messaggio senza senso. Questo capita perchè, essendo una tecnica di analisi statistica, più è lungo il testo da analizzare e più è probabile che la frequenza delle lettere sia in linea con la frequenza delle lettere nella lingua italiana.

Exercise 2) Create a new project by starting from the previous exercise.

The extended program manipulates the message provided in exercise 1), by applying the decryption with the key that you have found in the previous exercise (specified in a constant named KEY). The resulting message has to be stored in a proper *read-write* area. Please note that, if they KEY that you have found in Q1 is correct, the resulting decrypted message will be a clear message in the Italian language.

Report the requested values in the table below.

	Execution time @12MHz	Code size	Data size
Exercise 1)	2408	104	163(const)+29(var)=192
Exercise 2)	2132	116	172(const)+163(var)=335

Respond to the following open questions.

Q3: What section have you used to store the new message?

Ho usato la sezione “VariableData”, con attributi Data e ReadWrite per potervi accedere in scrittura.

Q4: Which address range is assigned to this section and which memory of the system is used?

Lo spazio occupato dalla sezione è di 163Byte, ma per via del padding le vengono assegnati 164Byte, nel range di indirizzi da 0x10000000 a 0x100000a4. Questa porzione dello spazio di indirizzamento appartiene al 1° blocco di memoria SRAM on-chip.