Architetture dei Sistemi di Elaborazione	Delivery date: 7th December 2023
Laboratory 7	Expected delivery of lab_07.zip must include: - zipped project folder of the exercises 1 and 2 - this document compiled possibly in pdf format.

Exercise 1)

A tennis player is following a strict food diet, in which she must count the number of calories taken in from the food eaten and the sport performed. Write a program in **ARM assembly** language that counts the **number of total daily calories**, subtracting from those taken in through food, those consumed through sports.

```
Days                    DCB 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07

Calories_food           DCD 0x06, 1300, 0x03, 1700, 0x02, 1200, 0x04, 1900,
                        DCD 0x05, 1110, 0x01, 1670, 0x07, 1000

Calories_sport          DCD 0x02, 500, 0x05, 800, 0x06, 400

Num_days                DCB 7
Num_days_sport          DCB 3
```

`Days` is a table where each entry consists of a day of the week (e.g., 0x01 is Monday, 0x02 Tuesday, ..)
`Calories_food` is a table where each entry consists of two integer values: the ID of the day (4 bytes) and the quantity of calories assumed with food (4 bytes).
`Calories_sport` is a table where each entry consists of two integer values: the ID of the day (4 bytes) and the quantity of calories consumed with sport activities (4 bytes). Notice that not all days she plays sport.
`Num_days` is a 1-byte constant and indicates the number of days in a week.
`Num_days_sport` is a 1-byte constant and indicates the number of days she plays tennis.

Compute the **total number of days** she takes in less than 500 calories per day and store it in register R11.

Note: The constant data section must be defined in the code section, with a 2byte alignment and 4096 boundary zero bytes.
Example:

```
…
// ALIGNMENT
// BOUNDARY (SPACE ….)
MY DATA
// BOUNDARY (SPACE ….)
..
```

Exercise 2)

Save in two separate vectors `Calories_food_ordered` and `Calories_sport_ordered`, the ID of the days in descending order by calories assumed or consumed, respectively.
The output will be, for example:

```
Calories_food_ordered          DCD   0x04,0x03,0x01,0x06,0x02,0x05, 0x07
Calories_sport_ordered         DCD   0x05,0x02,0x06
```

Then, save in R11 the ID of the least "caloric" day.

Compute the needed bytes for the above vectors.

Vector	Size [bytes]
`Calories_food_ordered`	28
`Calories_sport_ordered`	12

Report the following program characteristics (Hint: See the build output window in Keil).

	Size [bytes]
Program Size	564
Read Only data	204
Read Write data	68
Zero Initialized data	516

And provide a brief explanation about which directives can influence the previous program characteristics.

La direttiva AREA dati, DATA, READWRITE, ALIGN=2 nel codice assembly Arm definisce una sezione di memoria chiamata "dati" destinata a contenere dati modificabili, consentendo sia operazioni di lettura che di scrittura. Inoltre, l'inizio di questa sezione è allineato su un limite di memoria di 2 byte, ottimizzando la disposizione in memoria dei dati.
In assembly Arm, i valori costanti inizializzati vengono collocati in una sezione separata chiamata "literal pool". Questa pratica contribuisce a ottimizzare la disposizione dei letterali nel codice sorgente, migliorando l'organizzazione della memoria e semplificando la gestione di tali costanti.
Quando si scrive l'istruzione "LDR", lo spazio riservato al campo Offset, che serve a risalire alla costante, è di 12 bit. Pertanto, il " literal pool " non può trovarsi a una distanza superiore a 12 bit di indirizzo dall'istruzione che richiama la costante. In tal caso, utilizziamo la direttiva "LTORG" per spostare il " literal pool " nella posizione in cui viene scritta la direttiva, rendendolo più vicino al codice. Questo consente alle istruzioni di accedere alle costanti in modo più efficiente.