Architetture dei Sistemi di Elaborazione	Delivery date: **By 2:00 AM on 20th November 2024**
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 videogame speedrunner is tracking their daily attempts at speedrunning a game, recording both their best times and their total attempts per day. Write a program in **ARM assembly** language that analyzes their **speedrunning performance data**.

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

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

Failed_runs         DCD 0x02, 50, 0x05, 30, 0x06, 100, 0x01, 58,
                     DCD 0x03, 40, 0x04, 90, 0x07, 25

Num_days           DCB 7
```

`Days` is a table where each entry consists of a day of the week (e.g., 0x01 is Monday, 0x02 Tuesday, ..)
`Best_times` is a table where each entry consists of two integer values: the ID of the day (4 bytes) and the best time (in seconds) achieved that day by the speedrunner (4 bytes).
`Failed_runs` is a table where each entry consists of two integer values: the ID of the day (4 bytes) and the number of times the player had to reset the game (4 bytes). Notice that not all days he plays videogames.
`Num_days` is a 1-byte constant and indicates the number of days in a week.

Compute the **total number of days** the speedrunner best time was better or equal to 1300 and store it in register R11. Then for each day this time was better or equal to 1300 sum the number of Failed_runs and store it in register R10.

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 `Best_times_ordered` and `Failed_runs_ordered`, the ID of the days in descending order by best times and failed runs, respectively.
For example at the end the vectors would be ordered as follows:

```
Best_times_ordered       DCD   0x04,0x03,0x01,0x06,0x02,0x05, 0x07
Failed_runs_ordered            DCD   0x06,0x04,0x01, 0x02, 0x03, 0x05, 0x07
```

Then, save in R11 the ID of the worst "best_time" day.

Compute the needed bytes for the above vectors.

Vector	Size [bytes]
`Best_times_ordered`	7 * 2 = 14
`Failed_runs_ordered`	7 * 2 = 14

Student notes: i used 2 memory areas for each ordering to keep track not only of the ordered days but also for the corresponding values

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

	Size [bytes]
Program Size	108 in a 564 block
Read Only data	8520
Read Write data	32
Zero Initialized data	512

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

Program Size: This is not the actual size of your code but how memory is actually allocated to keep your code, in this case 564 bytes are enough to allocate our code (I guess the code is organized in pages of 564 bytes). So to increase the number of bytes(pages) needed by our program we need to use more than 564 bytes
Read Only Data: To increase this value we need to allocate more bytes in the ROM memory (READONLY area)
Read Write Data:To increase this value we need to allocate more bytes in the ROM memory (READWRITE area)
Zero Initialized Data: These are the bytes initialized to 0 by the areas with the directive NOINIT