Architetture dei Sistemi di Elaborazione **Laboratory 7** **REVALOR RICCARDO s339423**	Delivery date: 20th November 2024 Expected delivery of lab_07.zipmust 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 thebest 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.
The output will be, for example:

```
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. R11 = 0x07

Compute the needed bytes for the above vectors.

Vector	Size [bytes]
`Best_times_ordered`	28
`Failed_runs_ordered`	28

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

	Size [bytes]
Program Size	Code + RO + RW = 9652
Read Only data	204
Read Write data	224
Zero Initialized data	512

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

The most important directives that can change the characteristics of the program are SPACE and AREA.
The directive SPACE … (ex SPACE 4096) increases the Program Size. If I removed the two SPACE 4096 directives, overall Program Size would decrease to 992 bytes,
If I added "SPACE x" to a an Area with the attribute "READWRITE", it would increase the RW Data to 224 + x bytes.
The READWRITE attribute puts the custom defined "MyData" Area into the RW Data. Changing this attribute to READONLY would decrease RW Data to 0 bytes, increase RO Data to 756 bytes and decrease Program Size to 9428 bytes.
DCD in Areas with the attribute "READWRITE" increase the RW Data by 4 bytes every time these directives are used. If they are put in "READONLY" Areas they increase the RO Data instead.
The NOINIT attribute, if applied to an Area, increases the ZI Data and decreases the RW/RO Data because the data contained in that area will be initialized to zero at runtime.