|  |  |
| --- | --- |
| **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 | 28 |
| Failed\_runs\_ordered | 28 |

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

|  |  |
| --- | --- |
|  | Size [bytes] |
| Program Size | 8588 |
| Read Only data | 764 |
| Read Write data | 0 |
| Zero Initialized data | 512 |

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

|  |
| --- |
| Il program size influenza in maniera significativa le prestazioni e la memoria occupata e .text contiene tutte le istruzioni eseguibili.  Il read only data include dati immutabili come costanti e vettori e vengono modificati solo al termine dell’esecuzione.  Il read write data include tutti i dati che possono essere modificati durante l’esecuzione del programma come ad esempio Best\_times\_sort e Failed\_run\_sort.  Il zero initialized data contiene tutti i dati inizializzati implicitamente a 0, come tutte le zone di memoria allocate con SPACE. |