**Question 1** (10 points)

The transpose *AT* of a matrix *A* is a new matrix obtained by swapping the rows and columns of *A*.

A bit matrix A is stored as an array of 8 bytes in a READONLY area of memory, for example:

matrix DCB 0xF8, 0x7C, 0x3E, 0x1F, 0x8F, 0xC7, 0xE3, 0xF1

Another uninitialized bit matrix *AT* is allocated as an array of 8 bytes in a READWRITE area of memory.

Write the transposition routine in ARM assembly which receives in input:

1. the address of the bit matrix *A*
2. the address of the bit matrix *AT*

The subroutine saves in the second matrix the transpose of the first matrix *A*.

Considering the matrix *A* in the example, the transpose *AT* will contain the values 0x8F, 0xC7, 0xE3, 0xF1, 0xF8, 0x7C, 0x3E, 0x1F. The binary representation of the matrices is:

You can use the following algorithm to compute the transpose of *A*:

for (*i* = 0; *i* < 8; *i* ++)

*x* = *i*-th row of *A*

for (*j* = 0; *j* < 8; *j* ++)

*b* = *j*-th bit of *x* (note: bit 0 is the most significant bit, bit 7 is the least significant bit)

store *b* to the *i*-th bit of the *j*-th row of *AT*

Important notes:

1. Write your code inside the “ARM” directory
2. The assembly subroutine must comply with the ARM Architecture Procedure Call Standard (AAPCS) standard (in terms of parameter passing, returned value, callee-saved registers).

**Question 2** (8 points)

Extend the previous exercise as follows.

Declare two uninitialized arrays of 8 bytes *A* and *B*.

Initialize TIMER2 to count to 0xFFFF with a period of 0.1 seconds. When the counter reaches 0xFFFF, the timer is reset and no interrupt is generated.

When KEY1 is pressed the first time, set the first element of the array *A* equal to the value of TIMER2. When KEY1 is pressed the second time, set the second element of the array *A* equal to the value of TIMER2, and so on until KEY1 is pressed eight times. Subsequent pressures of KEY1 are ignored.

When KEY2 is pressed the first time, set the first element of the array *B* equal to the value of TIMER2. When KEY2 is pressed the second time, set the second element of the array *B* equal to the value of TIMER2, and so on until KEY2 is pressed eight times. Subsequent pressures of KEY2 are ignored.

When INT0 is pressed, you have to verify the following equivalence: (*A* + *B*)*T* = *AT* + *BT*

In detail, you need to call the transposition routine three times (after defining new uninitialized arrays to hold the results). If the equivalence is confirmed, switch on led 1. Otherwise, switch on led 2.