**Introduction to Computer Organisation and Architecture**

**Tutorial 8**

1. Convert the following code into ARM assembly code. Assume that x has been loaded into R0 and y into R1. You need to write only the CODE section.

|  |
| --- |
| if (y is larger than x)  swap the content of x and y  else  reset both x and y to 0 |

1. In Given a key value ***key****,* we can reference the lookup table **INDEX** to get an ***index***. The ***index****, in turn*, are used to retrieve an item in the table **DATA**. Assume that all tables are zero-indexed. For example, given ***key*** = 1, ***index***= **INDEX**[***key***] = 2, **DATA**[***index***] = 0x13. Store the result into the variable ***result***. Write an assembly program which performs this lookup function.

Write an assembly which retrieves a data from the table:

|  |  |
| --- | --- |
| ***result*** | (Expecting output = 0x13) |

|  |  |
| --- | --- |
| ***key*** | 1 |

|  |  |
| --- | --- |
| **DATA** | 0x11 |
|  | 0x12 |
|  | 0x13 |
|  | 0x14 |
|  | 0x15 |
|  | 0x16 |
|  | 0x17 |
|  | 0x18 |
|  | 0x19 |

|  |  |
| --- | --- |
| **INDEX** | 0 |
|  | 2 |
|  | 3 |
|  | 7 |

1. Assume the following 3x3matrix **A**has been loaded into the memory.

1 1 1

A = 2 2 2

3 3 3

Note that two-dimensional arrays are saved into the memory as single-dimensional array by concatenating the rows. To save memory transfer time, use **LDM** and **STM** instruction. Write a program to perform the following operations:Sum of all rows. Put the result into the array **RSUM**.

1. To enable detection of error in a 4-bit data, write an assembly code to append 1 parity bit to the data. Assume that *even* parity is used.

Input:

[A] = 0x05 (only the four lower bits are used)

Output:

[B] = 0x000000015 (Only the five lower bits are used)

1. Compare the following two strings to determine if they are equal. Assume that the length of the string is 3.

Input:

[STR1] = “CAT”

[STR2] = “CBT”

Output:

[S] = 0 (false)