King Fahd University of Petroleum and Minerals
College of Computer Sciences and Engineering
Department of Computer Engineering

COE 301 – Computer Organization (T161)
ICS 233 – Computer Architecture & Assembly Language (T161)

Programming Assignment # 02 (*due date & time:* <mark>*Sunday 20/11/2016 noon*</mark>)
Worth twice as much as Programming Assignment # 01

Write a MIPS assembly language program that implements the following:

i. Ask the user to enter number of rows, R, and number of columns, C, of an array.

ii. Ask the user to enter an R×C array of integers in a **row-wise** fashion.

iii. A procedure, **PrintA**, that prints the content of an array of integers in a two-dimensional format **(row-wise)** leaving a space between elements. Assume that the procedure receives as parameters the address of the array in register $a0, the number of rows in register $a1, and the number of columns in register $a2.

iv. A procedure, **RSum**, that computes the sum of a given row. Assume that the procedure receives as parameters the address of the array in register $a0, the number of columns in register $a1, and the index of the row to be summed in register $a2. The procedure should return the sum of the row in register $v0.

v. A procedure, **CSum**, that computes the sum of a given column. Assume that the procedure receives as parameters the address of the array in register $a0, the number of rows in register $a1, and the index of the column to be summed in register $a2. The procedure should return the sum of the column in register $v0.

vi. A procedure, **RSort**, that sorts the array elements of a given row. Assume that the procedure receives as parameters the address of the array in register $a0, the number of columns in register $a1, and the index of the row to be sorted in register $a2. The sorted row elements should replace the existing row in the array.

vii. A procedure, **CSort**, that sorts the array elements of a given column. Assume that the procedure receives as parameters the address of the array in register $a0, the number of rows in register $a1, and the index of the column to be summed in register $a2. The sorted column elements should replace the existing column in the array.

viii. Print a menu from which the user can select one of the following options:
1. Print the Array
2. Print Sum of a Row
3. Print Sum of a Column
4. Sort and Print a Row
5. Sort and Print a Column
6. Exit the program

ix. Every time the user selects any of the first 5 options the program executes the selected option and returns back to displaying the 6 options again. Only when the user selects option 6 that the program exits.

A sample execution of the program is shown below:

```
Enter number of rows: 2
Enter number of columns: 3
Enter an array of 2×3 integers:
3
2
4
```

```
6
1
5
```

```
Select one of the following functions:
1.    Print the Array
2.    Print Sum of a Row
3.    Print Sum of a Column
4.    Sort and Print a Row
5.    Sort and Print a Column
6.    Exit the program
```

If the user selects the first option, then the following should be displayed:

```
Array of 2×3 integers is:
3 2 4
6 1 5
```

If the user selects the second option, then the following should be displayed:

```
Enter the row number: 0
Sum of row number 0 is: 9
```

If the user selects the third option, then the following should be displayed:

```
Enter the column number: 1
Sum of column number 1 is: 3
```

If the user selects the fourth option, then the following should be displayed:

```
Enter the row number: 0
Sort of row 0: 2 3 4
```

If the user selects **after that** the fifth option, then the following should be displayed:

```
Enter the column number: 1
Sort of column 1: 1 3
```

If the user selects **after that** the first option again, then the following should be displayed:

```
Array of 2×3 integers is:
2 1 4
6 3 5
```

If any of the entered row numbers or column numbers are out of range, your program should display an error message and asks the user to reenter the required information.

Submit through email a soft copy of your solution in a zip file with the subject line "COE301/ICS233-Prog02-yourID" to both marwan@kfupm.edu.sa and s201375910@kfupm.edu.sa. Your solution should be submitted in a **word file** that contains the following items:

(a) Your name and ID

(b) Assignment number

(c) Problem statement

(d) Your results along with the code

In addition, submit to my office a hard copy of the word file no later than noon on the due date.

Copying programming assignment is not allowed. This work should be done individually. Detected copies will get zero grades. This includes the one who wrote the program and the one who copied it.