# Q1.Assembly Language Programming in MIPS

Definition: In this question, you are asked to implement an assembly application in MIPS ISA. Your application first asks a positive integer value; N, from the console. Then, it lists all positive numbers which are less than N and divisible by 7.

# Q2.Assembly Language, Programming a Recursive Method in MIPS

Definition: In this homework, you are asked to implement an application in assembly in MIPS ISA. Your application will find the maximum value in an array by implementing following two methods;

* *int maxInArray(int A[], int n) : This method finds and returns the largest value in A[0..n-1]*
* *int maxInInterval(int A[], k, n): returns the largest value in A[k..n-1]*

The pseudocode is as follows.

|  |
| --- |
| int **maxInArray**(int A[], int n)  return maxInInterval(A, 0,n) |
| int **maxInInterval**(int A[], k, n)  if k==n-1 then  return A[k]  else  temp=maxInInterval(A, k+1,n)  if temp > A[k] then  return temp  else  return A[k]  end if  end if |

Your assembly language code must implement the algorithms exactly as shown above, including using the same interfaces (number of parameters, types, and order), and using recursion in maxInInterval to find the largest value in the rest of the array.

You should use the standard MIPS calling conventions, including use of registers and the stack, for function calls.

* 1. Function maxInArray will be provided with the memory address of A[ ] in register a0, and n in register a1. Return the answer in register v0.
  2. Function maxInInterval will be provided with the memory address of A[ ] in register a0, k in register a1 and n in register a2. Return the answer in register v0.

Also in your code include the main method assigning some values to A array and calls maxInArray method.

Rules:

* Your code will be tested in QtSpim simulator. In case your code does not work in Qtspim simulator, it will not be evaluated at all.
* You must do all work individually. All submitted studies will be passed through similarity check.
* Finding an online code or using converter tools will not help you learn about computer assembly. Also, there might be other students in the class using that specific resource.

Hints:

* For understanding MIPS assembly better, please refer to the Appendix-A of course book (Computer Organization and Design, 5th edition). You can see the pdf of the book in this link: http://ac.aua.am/Arm/Public/2017-Spring-Computer-Organization/Textbooks/ComputerOrganizationAndDesign5thEdition2014.pdf
* To be able to read from the console and write to console, you need to use system calls. SPIM provides a small set of operating system–like services through the system call (syscall) instruction. To request a service, a program loads the system call code (see Figure A.9.1) into register $v0 and arguments into registers $a0–$a3 (or $f12 for floating-point values). System calls that return values put their results in register $v0 (or $f0 for floating-point results). For example, the following program prints "the answer = 5".

.data

str: .asciiz "the answer = "

.text

.globl main

main:

li $v0, 4 # system call code for print\_str

la $a0, str # address of string to print

syscall # print the string

li $v0, 1 #system call code for print\_int

li $a0, 5 # integer to print

syscall #print int

li $v0, 10 # terminate program

syscall