CSCI 3301 – MIPS Programming Using SPIM
Programming Assignment
Due Date: Monday, Nov 29, 2021, 11:59 PM

Instructions:

- The assignment must be done by you alone.
- Your programming code must be well commented.
- You need to write a report. You need to include your output screenshots in the report and briefly explain.
- You need to test your code for edge cases. Explain how you checked and solved the edge cases in the report.
- For the two MIPS-programs, you will generate 2 text files, named Facto.s and Sort.s. Put them in a folder.
- Put your report as a PDF file, compress the folder and submit the compressed folder.
- If you don't write the report, your assignment will not be graded.

Total Marks: 100.

Part 1 [50 points]:
The factorial function is mathematically defined as:

$$n! = \prod_{k=1}^{n} k$$

For example, factorial of 6, that is 6! = 720, which is computer as: $6 \times 5 \times 4 \times 3 \times 2 \times 1$.
Write a *recursive* MIPS assembly program using SPIM simulator (such as QtSpim), where for a given number (*n*) as an input (taken from the console), the program will recursively compute and return the corresponding factorial number. For retuning the result, use the standard console (screen) and print the output. You program must be able to take positive as well as negative integer numbers as input and must be able to respond appropriately.

Edge cases:
• 0! = 1
• Negative factorial is not defined. Return an error message.
• After 12 factorial, the output is larger than 32 bits. Therefore, return out of boundary error message.
• Also show some regular outputs, e.g. 6! = 720

Part 2 [50 points]:

Develop a MIPS program, "SORT", to sort a set of given integers (note: both positive and negative integers are allowed) by the user. Note that, you can use any sorting algorithm you want including the Bumble sort which we covered in the class.

First, your program will ask the user, how many numbers the user wants to enter, and then ask the user to provide those numbers.
Your program will read in those numbers and
will place those numbers in an array.
Sort the array elements in ascending order.
Now, print the set of numbers that the user entered saying, "You have entered: …".
Then, print the sorted list saying, "Here is the sorted list in ascending order: …".

Edge cases:
• Empty array
• All negative integers in the array
• All positive integers in the array
• Positive and negative integers together in the array