A1 - RISC-V Assembly Programming Assignment

Use the RARS simulator (https://github.com/TheThirdOne/rars) for this assignment. Your task is to write RISC-V code in RV64I instructions only and execute on the simulator.
Download RARS from here: https://github.com/TheThirdOne/rars/releases/tag/continuous. If you already have java, RARS will start right away.
Ubuntu command line to start RARS: $ java -jar {RARS jar file}
RARS Video explanation is here:
https://passlab.github.io/ITSC3181/resources/UsingRARS_ITSC3181.mp4

Aliter: You are free to go ahead and try and use any of the other RISC-V simulators to write and execute these programs. Please note that the ecall conventions (system call conventions) will be different for each simulator. List of RISC-V Simulators - https://riscv.org/exchange/software/.
Submission Notes:

1. Create 1 directory per Q. In each directory put code, screenshots, any other. Include a README containing the group members and the code contributions of each. Pack all the directories and the README in a Zip file and upload before the deadline.
2. A team has 2 members. Both members have to submit the same ZIP file.

Assignment Questions. Write a RISC-V Assembly program to

1. return your favorite number. Verify.
2. Print Hello World.
3. a. N_Sum: Calculate the sum of first N positive integers. Initiate an arbitrary N in the program.
   b. Calculate the sum of first N positive integers. N is an input to the program.
4. Print the first N Fibonacci numbers.
5. print_int: Write a program to convert a 2 digit integer to its equivalent ASCII string. Create a print_int procedure which takes the 2 digit decimal int as the input, and outputs the address of the ASCII string.
6. Use the print_int procedure to print the sum from the N_sum program.
7. Find the GCD of two numbers. Both numbers are initiated by the programmer in the code.
8. multiply_proc: multiply two numbers using only RV64I instructions.
9. Use multiply_proc to print the factorial of a number.
10. add two arrays. All the data can be init in the program.
11. sort an array (use your favorite sort algo). Array can be init in the program.
12. Print the first N Fibonacci numbers using recursion.
13. binary search.
14. convert an infix expression to prefix and postfix.

******