CS2160: Computer Organization Laboratory
Lab 1

Write the following programs in the *ToyRISC* ISA:

1. Write a program to count the number of positive (>= 0), even numbers in a given array l. The size of the array is n ($n > 0$). Place the result in register `x10`.
2. Write a program to check if a given number is prime. If yes, place 1 in `x10`. If not, place -1 in `x10`.
3. Write a program to sort an array of numbers in the descending order. The sorted array should be placed in the same addresses in memory as the initial array.
4. Write a program to find the histogram of marks obtained by n students in a course. The histogram of marks should be in the `count` array (see the given code template).
5. Write a program to place the first 'n' Fibonacci numbers in the memory. The first number is placed at address $2^{16} - 1$, the second at $2^{16} - 2$, and so on.
6. Write a program to place the first n numbers of an arithmetic series in the memory ($n > 0$). The first number is to be placed at address $2^{16} - 1$, the second at $2^{16} - 2$, and so on. The first value in the series is given as a, and the difference between two consecutive elements in the series is given as d.

Troubleshooting

- Follow the program syntax given in the ISA specification document closely.
- Don't have any empty lines in your program.

Submission Format

- The respective *.asm* files in the folder *lab*1 has the code templates. You **should** modify these files to incorporate your code (or solution).
- **DO NOT** add new files into the folder.
- The evaluation will be automated, and therefore any deviation from the specifications (in terms of filenames, outputs required, etc) will cause test cases to fail. Therefore, pay attention to all the details mentioned in the question.

- You may change the data values in the template code given for testing. But do not change the names given to the addresses. Remember to remove the comment lines.
- Commit and push changes you have made to the code (or template) often to bitbucket. The instructor and the TAs will pull your latest commit at the time of the deadline and evaluate it.

Testing Your Programs

- Change the working directory to `supporting_files`.
- To test your program, use the given Java application `emulator.jar`. The command is as follows:

  ```
  java -jar <path-to-emulator.jar> <path-to-asm-file> <starting-address> <ending-address>
  ```

  This command functionally emulates the program you have written and at the end prints the contents of the register file, as well as the contents of those addresses of the memory specified by `starting-address` and `ending-address`.

  Use this to test and debug your program. An awkward crash implies your program is syntactically incorrect. A graceful completion with unexpected contents in the register file and / or the memory implies your program is logically incorrect.
- One test case for each program is given. To evaluate, the command is as follows:

  ```
  python test_each.py <path-to-asm-file>.
  ```

  Make sure the name of your `.asm` file is as specified above.