

Computer Architecture Laboratory

Assignment 1

Write the following programs in the *ToyRISC* ISA:

1. Check if a given number is even or odd. If odd, write '1' to register `x10`. Else, write '-1' to register `x10`.
2. 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.
3. 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.
4. Write a program to check if a given number is a palindrome. If yes, place '1' in `x10`. If no, place '-1' in `x10`.
5. Write a program to check if a given number is prime. If yes, place '1' in `x10`. If no, place '-1' in `x10`.

Submission Format

- Submit one zipped archive named “<entry-number-1>_<entry-number-2>_assignment1.zip”.
- The archive must contain 5 files: even-odd.asm, descending.asm, fibonacci.asm, palindrome.asm, and prime.asm. Name your files *exactly* as mentioned.
- Use the template programs given. You may change the data values for your testing. But do not change the names given to the addresses, for example, `a` and `n` in descending_template.asm. Remember to remove the comment lines.
- Test each individual program using the test_each script. Make sure the test passes. You may change values in the input and expected output files for your testing purposes.
- Test your final zip archive using the test_zip script. Make sure the test passes.
- Both students must submit on Moodle.

Testing Your Programs

- Download and unzip the supporting files.
- To test your program, use the given Java application `emulator.jar`. The command is `java -jar <path-to-emulator.jar> <path-to-assembly-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 `python test_each.py <path-to-assembly-file>`. Make sure the name of your `.asm` file is as specified above.
- To test your zip archive, the command is `python test_zip.py <path-to-zip-archive>`. Make sure the name of your `.zip` file is as specified above. We will use the `test_zip` script to automatically grade your submissions. So please make sure your submission works before submitting.

Troubleshooting

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