

Assignment: Writing a Makefile for a Multi-Binary C Project

MultiBinary

```
|-----include
|       |----- dataop.h
|       |----- mathop.h
|       |----- stringop.h
|-----src
|       |-----dataop
|       |       |----- dataop.c
|       |       |-----main_data.c
|       |-----main.c
|       |-----mathop
|       |       |----- main_math.c
|       |       |----- mathop.c
|       |-----stringop
|       |       |----- main_string.c
|       |       |----- stringop.c
```

You are given a codebase with multiple C and header files, structured as shown above, for which you need to write a Makefile. The project consists of multiple C source files and header files, and is intended to generate four separate executable files:

1. Math_ops – Performs basic arithmetic operations.
2. String_ops – Performs basic string manipulations.
3. Data_ops – Handles simple data operations.
4. All_ops – Performs all the operations.

Requirements for the Makefile:

1. Build four executables:

- a. math_ops: Uses main_math.c, mathop.c, and mathop.h
- b. string_ops: Uses main_string.c, stringop.c, and stringop.h

- c. data_ops: Uses main_data.c, dataop.c, and dataop.h
 - d. all_ops: Uses main.c, mathop.c, mathop.h, stringop.c, stringop.h, dataop.c, and dataop.h
- 2. Track header dependencies:** Ensure that modifying a header (.h file) triggers recompilation of related .c files.
- 3. Include Debug Mode (make DEBUG=1):**
- a. When DEBUG=1, compile with -g -DDEBUG (debugging symbols).
 - b. Otherwise, optimize with -O2.
- 4. Provide standard Makefile targets:**
- a. all → Compile all binaries.
 - b. clean → Remove all compiled files.
 - c. run_math → Run the math_ops binary.
 - d. run_string → Run the string_ops binary.
 - e. run_data → Run the data_ops binary.
 - f. run_all_ops → Run the all_ops binary.