

Logic Programming Part 2

1. For the prime sieve in lecture, we wrote a numbers predicate that relates a number N to an ordered list of numbers from 2 to N . However, the implementation in lecture takes quadratic time, since it uses the append predicate. Write another version of numbers that has a "tail-recursive" structure so that it only takes linear time.
2. Use the CLP(FD) Prolog library to write a relation between a natural number N and its factorial F .

3. Say you are creating a project that has only one source file, `main.cpp` and you have the start of a Makefile with the following variables defined:

```
CCX = g++
PROJECT_FILE = main.cpp
PROJ_EXE = main.exe
TEST_INPUT_FILES = $(wildcard *.in)
CORRECT_TEST_OUTPUT = $(wildcard *.outc)
OUTPUT_FILES = $(patsubst %.in, %.out, $(TEST_INPUT_FILES))
```

Complete the makefile with minimal repeated code such that the following commands have the specified behavior. You may not need to use all the variables defined.

- `Make`: Builds the project
- `Make project`: Builds the project
- `Make tests`: Runs and diffs all tests
- `Make clean`: Removes all executables and output files

Hint: You may find it helpful to have an implicit makefile rule with the target `%.out`. Also, `$<` can be used to access the first dependency for a target