

CS29206 Systems Programming Laboratory

Spring 2024

Lab Assignment: 2

Date: 01–Feb–2024

The make utility

In this assignment, you design a static library for solving recurrence relations of a special kind, and write an application program (with *main*) for solving some sample recurrence relations using your library. The emphasis is on how you automate the library-building project using makefiles.

Consider the linear recurrence relation of constant order k and with constant coefficients:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + d \quad (\text{for } n \geq k).$$

A complete specification of the recurrence relation also needs k initial values a_0, a_1, \dots, a_{k-1} . Assume that $k, d, c_1, c_2, \dots, c_k, a_0, a_1, \dots, a_{k-1}$ are integers. Such a recurrence relation can be represented by a structure consisting of $k, d, c_1, c_2, \dots, c_k, a_0, a_1, \dots, a_{k-1}$. You do not have to define the structure or write the recurrence solver for a general value of k , but concentrate only on the three special cases $k = 1, 2, 3$. To that end, you define three data types *recrel1*, *recrel2*, and *recrel3*. For example, *recrel2* will consist of the five int values d, c_1, c_2, a_0, a_1 . You do not need to store k , because these three user-defined data types are designed for handling specific values of k . You also declare and implement three functions *findterm1*, *findterm2*, *findterm3* that, given the respective recurrence-relation structures and n , output a_n . You may use any algorithm (recursive or iterative) to implement these functions.

For each $k = 1, 2, 3$, you use a subdirectory for declaring *recrelk* and for declaring and implementing *findtermk*. Name the header and source files as *solnk.h* and *solnk.c*. Each of these subdirectories contains a Makefile to build the object file *solnk.o*. The top-level directory too contains a Makefile that recursively creates the three object files *solnk.o* in the respective subdirectories, and then archives these three files to a static library *librecrel.a*. Also write clean rules in each of these four Makefiles. You are required to use recursive make to recurse into the subdirectories and handle the building or cleaning rules, come back to the top-level directory, and do the remaining task if any (for example, combining the object files in the subdirectories to a single archive *librecrel.a*).

Also write an application program *examples.c* that makes *recrel* library calls to solve the following three recurrence relations.

[Tower of Hanoi]

$$a_n = 2a_{n-1} + 1, a_0 = 0.$$

[Fibonacci numbers + 5]

$$a_n = a_{n-1} + a_{n-2} - 5, a_0 = 5, a_1 = 6.$$

[Fibonacci numbers]

$$a_n = 2a_{n-2} + a_{n-3}, a_0 = 0, a_1 = 1, a_2 = 1.$$

The files you write/generate should be organized as follows.

Before make in A2/lib/	After make in A2/lib/	After make clean in A2/lib/
A2 A2/app A2/app/examples.c A2/lib A2/lib/Makefile A2/lib/order1 A2/lib/order1/Makefile A2/lib/order1/soln1.c A2/lib/order1/soln1.h A2/lib/order2 A2/lib/order2/Makefile A2/lib/order2/soln2.c A2/lib/order2/soln2.h A2/lib/order3 A2/lib/order3/Makefile A2/lib/order3/soln3.c A2/lib/order3/soln3.h	A2 A2/app A2/app/examples.c A2/lib A2/lib/Makefile A2/lib/ librecrel.a A2/lib/order1 A2/lib/order1/Makefile A2/lib/order1/soln1.c A2/lib/order1/soln1.h A2/lib/order1/ soln1.o A2/lib/order2 A2/lib/order2/Makefile A2/lib/order2/soln2.c A2/lib/order2/soln2.h A2/lib/order2/ soln2.o A2/lib/order3 A2/lib/order3/Makefile A2/lib/order3/soln3.c A2/lib/order3/soln3.h A2/lib/order3/ soln3.o	A2 A2/app A2/app/examples.c A2/lib A2/lib/Makefile A2/lib/ librecrel.a A2/lib/order1 A2/lib/order1/Makefile A2/lib/order1/soln1.c A2/lib/order1/soln1.h A2/lib/order2 A2/lib/order2/Makefile A2/lib/order2/soln2.c A2/lib/order2/soln2.h A2/lib/order3 A2/lib/order3/Makefile A2/lib/order3/soln3.c A2/lib/order3/soln3.h

You need to submit only these files.

make distclean will additionally delete **librecrel.a**.

Sample Output

```
~/SPL/A2$ cd lib/
~/SPL/A2/lib$ make
. . .
~/SPL/A2/lib$ cd ../app/
~/SPL/A2/app$ gcc ... examples.c ...
~/SPL/A2/app$ ./a.out
n = 20
ToH(20) = 1048575
Fib(20) + 5 = 6770
Fib(20) = 6765
~/SPL/A2/app$ cd ../lib/
~/SPL/A2/lib$ make distclean
. . .
~/SPL/A2/lib$ cd ../app/
~/SPL/A2/app$ rm a.out
```

Pack the entire subdirectory A2 into a single zip file A2.zip. The zip file must not contain any binary files (the object files or the library). Submit the zip file to the CSE Moodle server before the submission deadline.