

Compilers [Fall 2017]

Programming Assignment I

Objectives

1. To understand the definitions of DJ and DISM, which will serve as source and target languages for a compiler built in future assignments.
2. To implement small DJ and DISM programs.
3. To become familiar with and able to use a DISM simulator.

Due Date: Sunday, September 3, 2017 (at 11:59pm).

Machine Details

Complete this assignment by yourself on the following CSEE network computers: c4lab01, c4lab02, ..., c4lab20. These machines are physically located in ENB 220. You can connect to the C4 machines from home using SSH. (Example: Host name: *c4lab01.csee.usf.edu* Login ID and Password: <your NetID username and password>) You are responsible for ensuring that your programs compile and execute properly on these machines.

Assignment Description

For this assignment, you will acquaint yourself with the DJ and DISM languages by implementing one small program in each language. You will write a DJ program and a DISM program for storing, retrieving, and dividing numbers.

The desired functionality is for your programs to input natural numbers as they're entered by the user, until the user enters a zero. Think of these numbers as getting stored into an array A , with the first number entered stored in $A[0]$, the second number stored in $A[1]$, etc. After the user enters the first 0, your programs will input two more natural numbers, i and j , and then output the quotient and remainder when $A[i]$ is divided by $A[j]$. If the user enters an i or j value beyond the bounds of A (for example, if A 's last element is stored at $A[5]$ but the user enters 6 for the value of i), your program needs to print 0 for both the quotient and remainder. You may assume that the user never stores more than 10,000 numbers into A .

Write your DJ program in a file called *qr.dj*, and your DISM program in a file called *qr.dism*.

Examples of Desired Behavior:

```
Enter a natural number: 7
Enter a natural number: 2
Enter a natural number: 0
Enter a natural number: 0
Enter a natural number: 1
3
1
```

(Here $A[0]=7$ and $A[1]=2$, so $A[0]/A[1]$ has quotient 3 and remainder 1)

```
Enter a natural number: 0
Enter a natural number: 0
Enter a natural number: 0
0
0
```

(Here A has no elements, so the $i=0$ and $j=0$ values are out of bounds and the program prints 0 for both the quotient and remainder)

Hints

Array A never contains the value 0 because the program stops populating A as soon as the first 0 is input. Hence, you don't need to worry about division by (or of) zero.

Whenever a DJ or DISM program attempts to read a natural number, the prompts of "Enter a natural number: " get printed automatically. Hence, you also don't need to worry about outputting those prompts. DJ and DISM programs can only input and output natural numbers (using the *readNat* and *printNat* calls in DJ, and the *rdn* and *ptn* instructions in DISM).

My *qr.dj* is 51 lines of code (not counting whitespace/comments), and my *qr.dism* is 19 lines of code.

Testing Your DISM Program

Please use the DISM simulator, *sim-dism*, to test your DISM program. When your DISM program halts, it may halt with any code.

Testing Your DJ Program

Because you are writing a program in a new language for which no compiler yet exists, you can't test your program by executing it! This situation is unpleasant but realistic. You'll have to ensure by hand that your DJ program is valid and would behave correctly if executed. You could, however, modify your DJ program into a valid Java program (using Java's Scanner class to mimic *readNat*), and then test that Java program.

Formatting, Grading, and Submission Notes

- To make it easier for our TA to read and grade code files, **use spaces rather than tabs** in your code and **avoid long lines of code** (I try to limit lines to 80 characters).
- Your programs will be graded on both correctness and style, so include good comments, well-chosen variable names, etc. For full credit, your code must not be significantly more complicated than necessary.
- The TA will test submissions on inputs not shown in the example above.
- Type the following pledge as an initial comment in every file you submit for this course: "I pledge my Honor that I have not cheated, and will not cheat, on this assignment." Type your name after the pledge. Not including this pledge in a submitted file will lower your assignment grade 50%.
- Upload and submit both of your files for Assignment 1 in Canvas.
- You may submit your assignment in Canvas as many times as you like; we will grade your latest submission.
- For every day that your assignment is late (up to 3 days), your grade reduces 10%.