| CS 211 | Computers and Programming | Handout # |
|---|---|---|
| | Fall 1999 | |
| **Assignment 5: Recursion and stuff** | | October 28, 1999 |

## Due date: November 4, 1999

You have ONE WEEK to do this assignment.

1. The binomial coefficients C(N,k) are defined as follows:

```
C(N,0) = 1
C(N,N) = 1
C(N,k) = C(N-1,k) + C(N-1,k-1) (for 0 < k < N)
```

   Write a recursive method to compute the value of C(N,k).

2. A fully parenthesized arithmetic expression (fpe) is an ordinary arithmetic expression in which parenthesese are put around all arithmetic operations so that the order of computation is explicit. For example,

```
 (2 + 3)
 ((2 + 3) - (3 - 4))
 ((2 * 3) * (3 - 4))
```

   are all fpe's. We will assume that an integer by itself is also an fpe. The following grammar describes fpe's.

```
E -> integer
  -> (E + E)
  -> (E - E)
  -> (E * E)
```

   It is easy to write a program that reads in an fpe from a file and evaluates it (the program would be similar to the boolean expression reader that you wrote in the last assignment). In this problem, however, we want you to write a program that reads in an fpe from a file and returns a SaM program for computing the expression. When this program is executed, it should leave the answer on the stack. Here are some fpe's and the SaM programs for evaluating them:

```
  fpe                  SaM program
  ---                  -----------

   2                   PUSHIMM 2
                       STOP

 (2 + 4)               PUSHIMM 2
                       PUSHIMM 4
                       ADD
                       STOP


((2 + 4) * (7 - 3))    PUSHIMM 2
                       PUSHIMM 4
```

```
ADD
PUSHIMM 7
PUSHIMM 3
SUB
TIMES
STOP
```

Remember that SaM arithmetic operations pop their operands from the stack, perform the computation, and push the result on the stack. So the code for computing $(E_1 + E_2)$ must

(a) compute $E_1$, leaving the result on the stack,

(b) compute $E_2$, leaving the result on the stack, and finally

(c) execute an ADD operation, which pops the two values computed earlier, adds them up, and pushes the result on the stack.

*Implementation details:*

- You may assume that a single legal fpe is given to you in the file. The file is passed to you as a command-line parameter. You should write the generated SaM code to a file of your choice. It is a good idea to write the output to a file with a .SaM extension — for example, if the input file is named exp.txt, the output file can be named exp.SaM but we will not insist on this.

  You can download the SaM interpreter from the web-page and run your code on this interpreter if you want to, but you do not have to show us the result of execution.

- The SaM program can be represented in your program as a String.

  Hint: If intIn is an integer variable containing the value 45, the expression

  ```
  "PUSHIMM " + intIn + "\n" + "STOP\n"
  ```

  will evaluate to a String which will be printed as follows:

  ```
  PUSHIMM 45
  STOP
  ```

  Note that `\n` is the new line character.

- The structure of the reader/code generator should be almost identical to the structure of the boolean expression evaluator from the last assignment. Study the solution to the boolean expression evaluator carefully and make sure you understand it before attempting to do this problem.

- This problem can be done in about 25 lines of Java code, so you start writing hundreds of lines of code, you are doing something wrong. Talk to your TA.

3. This problem does not require any programming. Answer each question, explaining your answers briefly.

   (a) A certain species of bacteria (Arborus Arborus) reproduces every hour by *binary fission*, a process that can be described as follows. Suppose you start with one bacterium at time t = 0. At time t = 1 hour, this bacterium divides itself into two bacteria. At time t = 2 hours, each of these two bacteria divides itself into two, so you have a total of 4 bacteria, and so on. Answer the following question about Arborus Arborus, *assuming no bacterium ever dies before reproducing.*

      i. How many bacteria exist at time t = $n$ hours? This will be some function of $n$, where $n \geq 0$.

      ii. If there is a single Arborus Arborus at t = 0, what is the first point in time when the number of existing bacteria will be larger than $10^{80}$? (The number $10^{80}$ is believed to be an upper bound on the number of atoms in the universe.)

iii. What is the total number of bacteria that have ever existed at or before time t $= n$? This will be some function of $n$, where $n \geq 0$. Assume that a bacterium is distinct from the two bacteria that result when it undergoes fission.

(b) A more realistic scenario is that some bacteria die before reproducing. Answer the following questions.

i. What is the smallest number of bacteria that can exist at time t $= n$ hours?

ii. What is the largest number of bacteria that can exist at time t $= n$ hours?

iii. At time t $= n$ $(n > 0)$, a biologist finds that at least one bacterium is alive. What is the smallest number of bacteria that could have existed at or before time $t = n$?