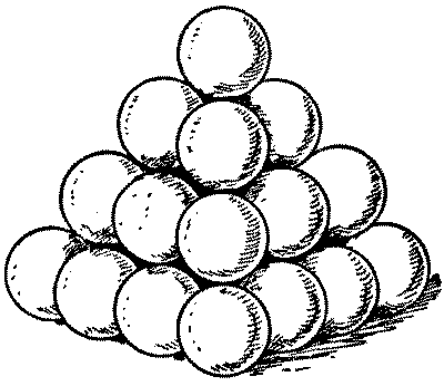
CPSC 2430 Data Structures

Fall Quarter 2024

Assignment 2 – Recursion for Mathematicians **Due: 9:00pm, Thursday Jan 18, 2024**

Task 1. Find the recursive formula for the sequence 2, 3, 6, 18, 108, 1944, … Then define a ***recursive*** function named ***mysterySequence*** that returns the n^{th} member of the sequence. For example, the output for *mysterySequence(4)* is 108.

Task 2. Spherical objects, such as tennis balls, can be stacked to form a pyramid with one tennis ball at the top, sitting on top of a square composed of four tennis balls, sitting on top of a square composed of nine tennis balls, and so forth.



Write a recursive function that takes as its argument the height of a pyramid of tennis balls and returns the number of tennis balls it contains. For example: *tennisBalls(3)* returns 14.

Task 3. Define a ***recursive*** function named ***decToBase2*** which given an integer, returns its binary representation as a string. For example, the binary equivalent of 13 may be found by repeatedly dividing 13 by 2. So, 13 in base 2 is represented by the string "1101" and the output for ***decToBase2***(13) should be “1101”.

Task 4. Define a ***recursive*** function ***isDivisibleBy7*** that returns ***true*** if the given number is divisible by 7, and ***false*** otherwise. For example: *isDivisibleBy7(1073)* returns *false*.
To find out if a number is divisible by 7, you can follow this algorithm:
Remove the last digit and double it. Subtract it from the remaining number. If the result is zero or a recognizable 2-digit multiple of 7, then the number is divisible by 7. Otherwise no or repeat if more than 2 digits are left. For example:
Is 1073 divisible by 7?

- Remove the last digit, 3, from the number and double it, which becomes 6.
- The remaining number becomes 107, so 107-6 = 101.

- 101 has 3 digits (which is more than 2), so repeating the process one more time: remove the last digit, 1, and double it, which becomes 2.
- Remaining number 10 – 2 = 8.
- As 8 is not divisible by 7, the number 1073 is not divisible by 7.

Your function *must* be recursive and implement the above algorithm. You cannot use division or modulo to check for divisibility by 7 until you are down to a 2-digit number.

Task 5. The famous Indian mathematician, Srinivasa Ramanujan, asked a question that stumbled a number of people: what is the value of

$$\sqrt{6+2\sqrt{7+3\sqrt{8+4\sqrt{9+5\sqrt{10+\cdots}}}}}$$

carried out to infinity?

Define a function, named ***ramanujan***. Which takes, as one of its arguments, the depth of a rational approximation to the above nested expression. For example, if the depth is 0, Ramanujan should return $\sqrt{6}$. If the depth is 1, Ramanujan should return the value of $\sqrt{6+2\sqrt{7}}$. If the depth is 2, the return value should be the value of $\sqrt{6+2\sqrt{7+3\sqrt{8}}}$. Your function should implement a ***recursive*** process. Your function may have more than one argument, as needed for your recursion to work.
At the end of your code, you need to give the value of the above expression when carried out to infinite.

Task 6. Write a function that tests all your other functions (Task 1-5) and neatly prints out their results. The following tests should be run for each and their results printed to screen.

1. mysterySequence with n = 0, 1, 4, 7
2. tennisBalls with height = 0, 1, 3, 10
3. decToBase2 with input = 0, 1, 13, 32, 321
4. isDivisibleBy7 with input = 1, 7, 31, 1073, 1729
5. Ramanujan with depth = 1, 3, 10

These tests are not exhaustive, and I suggest you run many more tests on your own to check the functionality of each of your functions from Task 1-5.

Your assignment should have a user interface to help test each function above and quit the program. For example:

```
Welcome to the recursion assignment. What would you like to test?
1.  mysterySequence
2.  tennisBalls
3.  decToBase2
4.  isDivisibleBy7
5.  ramanujan
6.  run tests
7.  Exit
1
Mystery Sequence. Please enter n to compute the nth number in sequence:
4
The 4th number in mystery sequence is 108.

Welcome to the recursion assignment. What would you like to test?
1.  mysterySequence
2.  tennisBalls
3.  decToBase2
4.  isDivisibleBy7
5.  ramanujan
6.  run tests
7.  Exit
2
Tennis Balls. Please enter the height of the pyramid:
3
A pyramid with 3 levels holds 14 tennis balls.

Welcome to the recursion assignment. What would you like to test?
1.  mysterySequence
2.  tennisBalls
3.  decToBase2
4.  isDivisibleBy7
5.  ramanujan
6.  run tests
7.  Exit
Ramanujan. Enter integer depth: 7
Result at depth 7: ?????
Result at infinite depth: ?????
```

Assignment 2 Submission

You need to submit the following file:

- assignment2.cpp

Your assignment2.cpp should include all the functions mentioned above. Before submission, you should ensure your program has been compiled and tested (extensively). Your assignment receives zero if your code cannot be compiled and executed.

You can submit your program multiple times before the deadline. The last submission will be used for grading. To submit your assignment, you should execute the following script (assuming your files are on cs1.seattleu.edu):

```
/home/fac/mjilani/submit/24wq2430/assignment2_submit
```

Assignment 2 Grading Breakdown

Breakdown	Points	Note
mysterySequence	10	mysterySequence returns the correct result and is implemented recursively
tennisBalls	10	tennisBalls returns the correct result and is implemented recursively
decToBase2	15	decToBase2 returns the correct result and is implemented recursively
isDivisibleBy7	20	isDivisibleBy7 returns the correct result and is implemented recursively
ramanujan	25	ramanujan returns the correct result and implemented recursively: 20 pts Give the value of the expression when carried out to infinity: 5 pts
runTests	15	Run all tests as indicated for each function and print their correct results: 3pts each
Clean code	5	Clean, well commented code, no global variables, etc.