C++ Programming Evaluation

Instructions:

You have 210 minutes to finish this evaluation. Please attempt all the problems, and try not to skip any problems completely. You can use any online resource, but cannot ask for help from any person. After 210 minutes, please submit your solution to careers@jqinvestments.com. Include all source code. Everything you write should be in English.

After you have submitted your solution, please spend 30 minutes writing a self-evaluation (in English). What did you think of the problems and how do you think you did? Send the self-evaluation to the above email as well.

If completing this evaluation in-office, use Codeblocks to program and compile. If not, you may use a programming environment of your choice.

Do not share this evaluation or your solutions with anyone

Problem 1.

Your program will answer questions based on the contents of `inputString`. We will test your program with many different values for `inputString`.

Part 1: Write a function that returns the number of unique characters in inputString. **We have already solved this part and the code is ready to compile. You do not need to do anything.**

Part 2: Write a function that returns the number of unique strings of three characters in inputString. For example, `abcbcb` returns 3.

Part 3: Write a function that uses `inputString` to print to screen a pyramid. For example,

```
asdfasdfasdfafafasdsdfsadfasdfgfafasdffasdfasdfafasasfedfafafasdfasd
```

Should output:

```
        a
       sdf
      asdfa
     sdfafaf
    asdsdfsad
   fasdfgfafas
  dffasdfasdfaf
 asasfedfafafasd
fasd
```

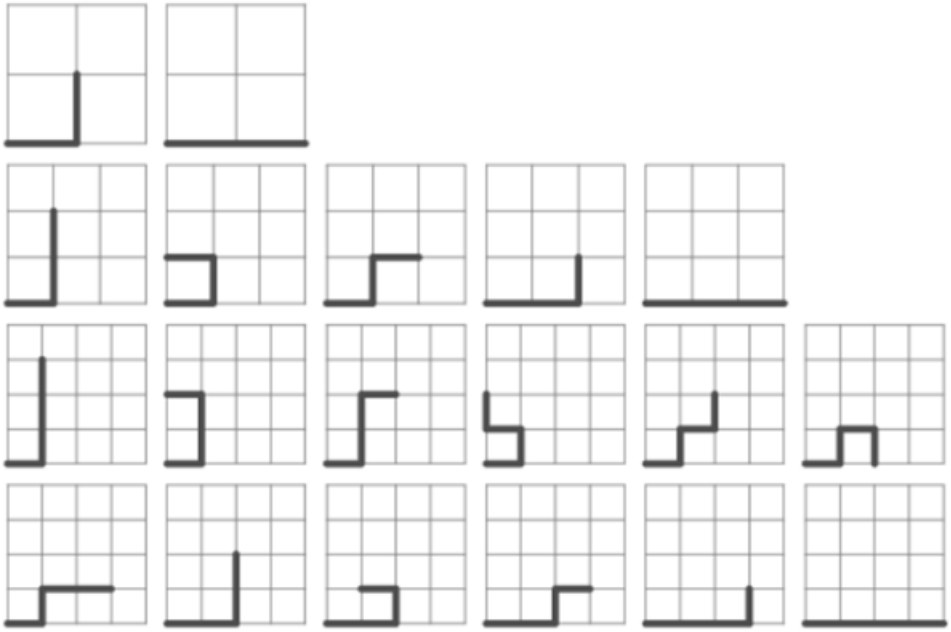
Problem 2.

On a square integer lattice of size N by N, we can move according to the following rules

- We always begin at $(0, 0)$.
- We must always move first to $(1, 0)$.
- We will move $N - 1$ additional steps. Each time, we can go from (a, b) to any of these four points: $\{(a + 1, b), (a - 1, b), (a, b + 1), (a, b - 1)\}$.
- We may not move to the same point twice.
- We may not move outside the lattice.

Your goal is to implement the `solveProblem2()` function that takes N as input, and outputs the number of unique routes that we can take through a lattice of size N.

As an example, the following is all unique routes for $N = 2, 3, 4$. The answers are f(2) = 2, f(3) = 5, f(4) = 12. Your solution does not need to generate images.



We will test your program with $N = \{5, 6, 7, ..., 24, 25\}$. The evaluation of each value of N should take less than 1 second.

Problem 3.

Run the following C (not C++) code:

```
main(x,y,z,p){while(y=++x)
for(z=0;z<y?z=z*8+y%8,y/=8,p=z==y|z/8==y,1:(x-++p||printf("%o\n",x))&&x%p;);}
```

Part 1: What does this code output? Why?

Part 2: Describe what `x, y, z, p` are used for.

Part 3: Remove the `,` (comma) operator.

Part 4: Remove the ternary operator.

Part 5: Rewrite the code to be easy to understand. Add at least 3 comments.

Problem 4.

The C++ program in the attached `problem4.cpp` has a variable `proble4Input` that is a list of unique characters in the form of a string. It generates all possible permutations of the list and a unique hash value of each permutation, and prints to screen.

The program is not working correctly. Specifically, the list of permutations is not complete and the hash values being printed are all identical. It is also too slow with the possibility of memory exceptions. Please find the 4 bugs in the program and add a comment to each change you make.