By: Renato Chomen Ramos

Embedded Software Engineer Quiz

Please submit answers by creating a public repo on Github or Bitbucket and sharing the url of the repo. Please do not submit answers in any other format. There are two questions in this quiz, please see page 2 for the second question.

Q1) Consider that you have a rectangular piece of paper of arbitrary dimensions N by M (where N and M are positive integers). You also have a pair of scissors which can cut perfectly straight with no loss of paper. You wish to reduce the original piece of paper into a series of perfect squares of paper, making the largest possible squares, and using all of the paper provided. Write a function in C that takes the inputs N, and M, and returns the series of squares that can be made out of that piece of paper. No fractional squares, i.e. no square should be less than 1 in length and width.

Some examples:
Input: N =6 , M = 5
Output: 5x5, 1x1, 1x1, 1x1, 1x1, 1x1

Input: N = 1, M = 1
Output: 1x1

Input: N = 9, M = 9
Output: 9x9

Answer for Question 1)

```
// Quiz Question 1
// By: Renato Chomen Ramos
#include <stdio.h>

int main() {
    int M = 11;
    int N = 4;
    int Current_number;
    int Largest_dimension;
    int Smallest_dimension;
    int Series_of_squares[200];
    int swap_variables;
    int i = 0;

    printf("Inputs: M = %d, N = %d", M, N);
    printf("\n");
    // Current_number will start as the smallest dimension
    if( M > N ){
        Current_number = N;
        Largest_dimension = M ;
        Smallest_dimension = N ;
    }
    else{
        Current_number = M;
        Largest_dimension = N ;
        Smallest_dimension = M ;
    }

    // while I can cut the papaer
    while(Largest_dimension > 0 && Smallest_dimension > 0){

        // See if I can cut the paper with the Current_number
        if( (Current_number <= Largest_dimension) && (Current_number <=
Smallest_dimension)){

            // cutt the paper
             Series_of_squares[i] = Current_number;
             i = i + 1;

             // reduce dimmension
             Largest_dimension = Largest_dimension - Current_number;

             // if no longer largest dimenssion swap
              if( Smallest_dimension > Largest_dimension ){
                    swap_variables = Largest_dimension;
                    Largest_dimension = Smallest_dimension;
                    Smallest_dimension = swap_variables;
              }
        }
        else{
            Current_number = Current_number - 1;
        }
    }

     printf("Output = ");

    for(int j = 0; j<i; j++){
       printf("%dX%d, ", Series_of_squares[j], Series_of_squares[j]);
    }
    return 0;
}
```

Q2) Below is a link to a spreadsheet with two columns A and B such that:

- A is the input
- B is the output

Based on the spreadsheet, there exist a function such as f that relates A to B which is:

$$B_i = f(A_i)$$

Where i is the row number of the spreadsheet.
For example:

For row i = 1:	f(15840) = cGp
For row i = 2:	f(16465) = cmW
For row i = 3:	f(17941) = cX3

Q2 a) First task is to find function $f(A_i)$ using these sets of points in the spreadsheet.

Q2 b) Once the $f(A_i)$ is found, what would be the output for the following inputs?

- ★ $f(30001) = ?$
- ★ $f(55555) = ?$
- ★ $f(77788) = ?$

Q2 c) What is the upper limit or maximum range of this function before there will be collisions or overflow?

Q2 d) The first three parts are mostly mathematical and once you find the solution, you realize you need your programming skills to solve these questions completely. Therefore, please share any code that you write in the process of solving the above problems.

Hint 1: One must have a good idea about the domain and the range of the function f.
Hint 2: There are patterns within the spreadsheet that can help you find the function f.

Answer for Question 2)

After hours of work, I was not able to figure out the function f(Ai). I first tried to relate the output to the ASCII table by transforming each letter into its corresponding ASCII value. From there I tried to find the equation by making each digit of the input correspond to a constant and this would be equal to the concatenation of the outputs. So, if the input was 40958 and the output was AQq the equation would look like this: x*4 +y*0+ z*9+ a*5+ b*8 = 658176. After many iterations aiming to determine the constants, no clear solutions were found. The exploration of other changes in the equation returned no valid results as well. Then, I explored different graph patterns with no valid results. As an example, Figure 1 shows the results after separating each letter and transforming it into it excel number representation, and summing the numbers together. Several attempts involving variations around these strategies found no coherent patterns as well.

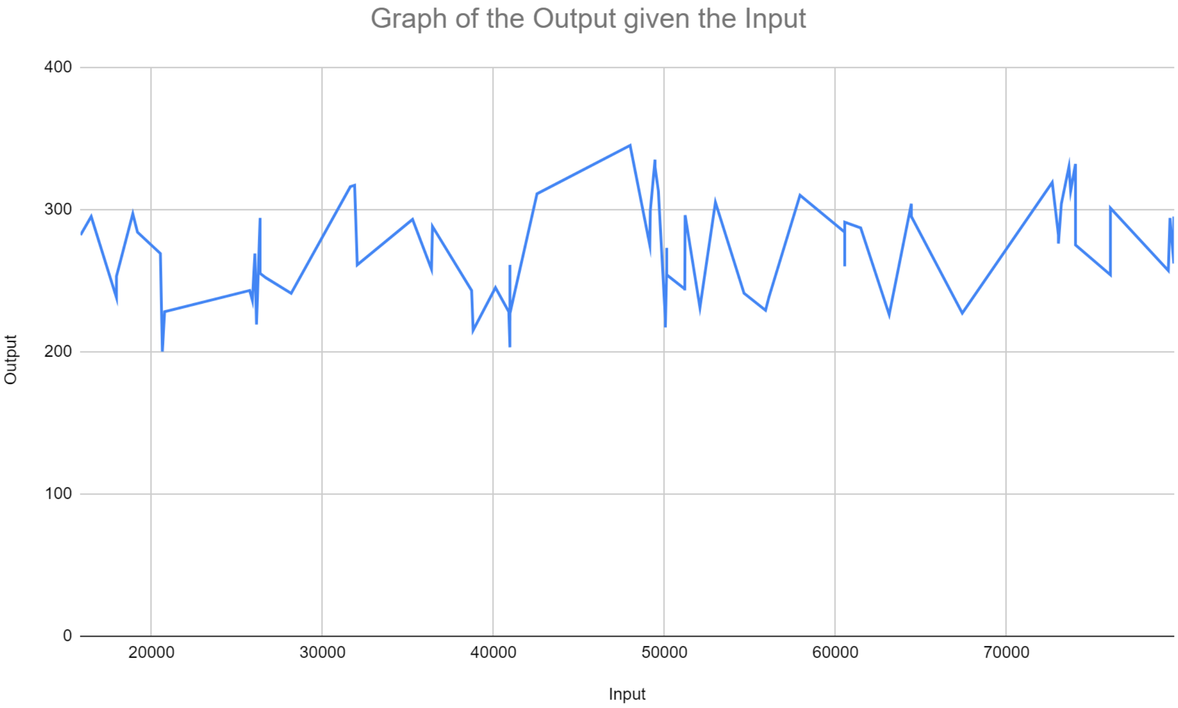

Figure 1: Graph of the output given the input.