Akmal Arifov
Elijah Amador

Problem:

Using recursion; take two numbers in from the user (a human) and add them together then separate the least significant digit and add the remaining digits and so on until you have a single digit answer.

EX: 87345 => 8734+5= 8739 => 873+9 = 882 => 88 + 2 = 90 => 9+0 =9

The goal of this should be to prepare for implementation—we don't have to get there, but we should build a roadmap to do so.

What is the goal? What is the outcome we are looking to resolve?

Recursion: Loop over the number set until we reach the destination number (in the example, 9)

Goal: Create a system that allows users to input two positive integers and calculate the total summation into a single digit by "breaking off" the integer's least significant digit and adding it back in until complete.

Requirements:

Functional:

User Input:

- The solution MUST prompt the user to input **two** POSITIVE integers
- The solution MUST respectively store the two integers (ex: int1, int2)

Input Validation:

- Should NOT take in alphabetical characters (Aa-Zz);
- Should NOT take in decimals or floats;
- Integer CANNOT be lower than 0 or greater than half of C# integer
  MaxValue = 2147483647 rounded down.

  Reason: Half of MaxValue was decided as our upper limit to avoid overflow from calculations/conversions.

Mathematical Requirements:

- The solution MUST calculate the sum of the two entered integers

- The solution MUST summate the digits of the calculated results until a single digit integer is returned
- At each summation, the solution MUST separate the least significant digit from the result to add back in

User Interactions:

- The solution MUST allow the user to retry after invalid input is entered.
- The solution MUST allow the user to exit upon request.
- The solution MUST display the final calculation to the user.

Non-Functional:

The above input validation requirements MUST be implemented into the system via try/catch exception handling

- In other words, the solution should NOT crash when receiving invalid input

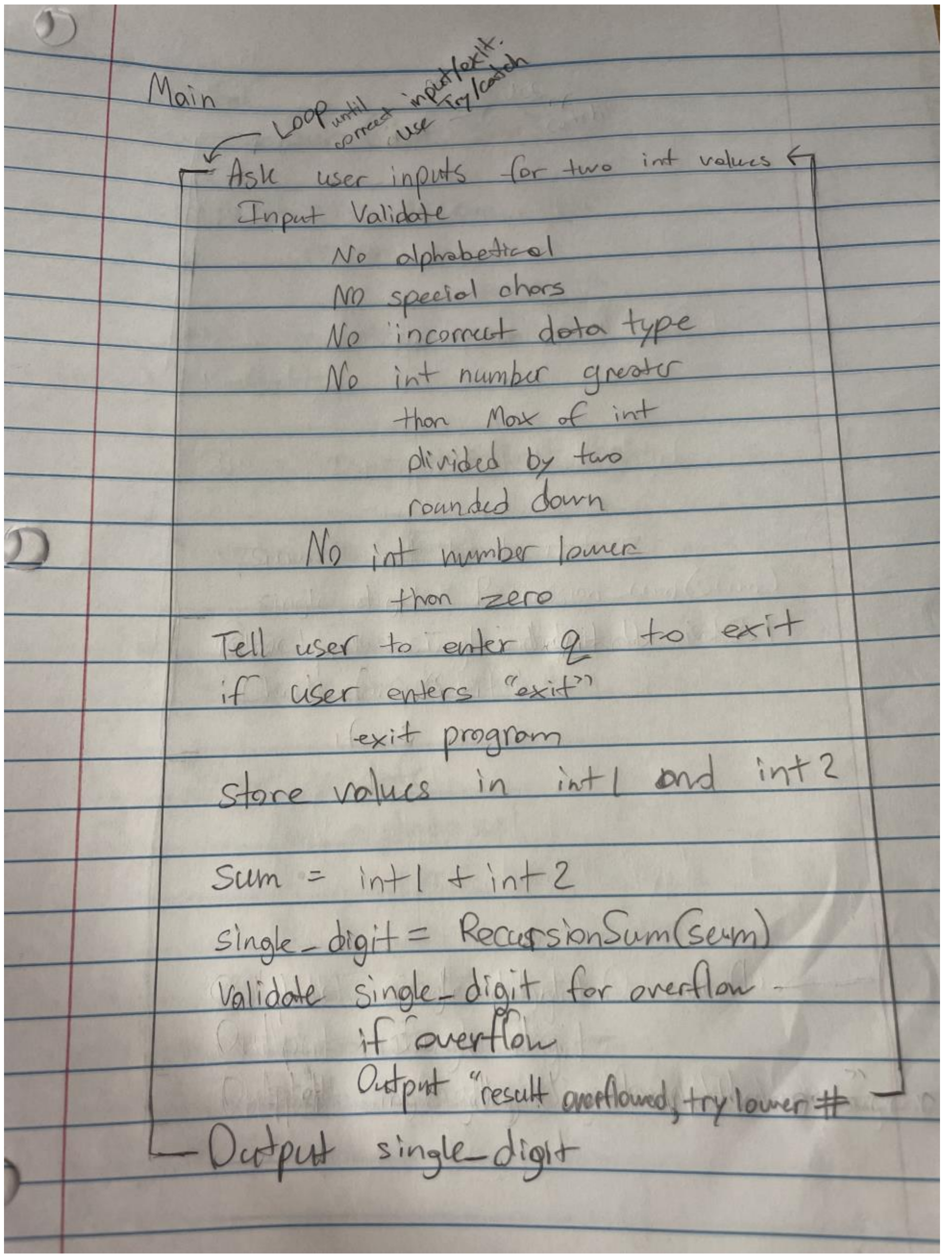
The solution MUST use recursion to implement the summation – non-functional constraint

The solution MUST implement a Console Application in C#

PSEUDOCODE & DESIGN ON PGS. 3 & 4

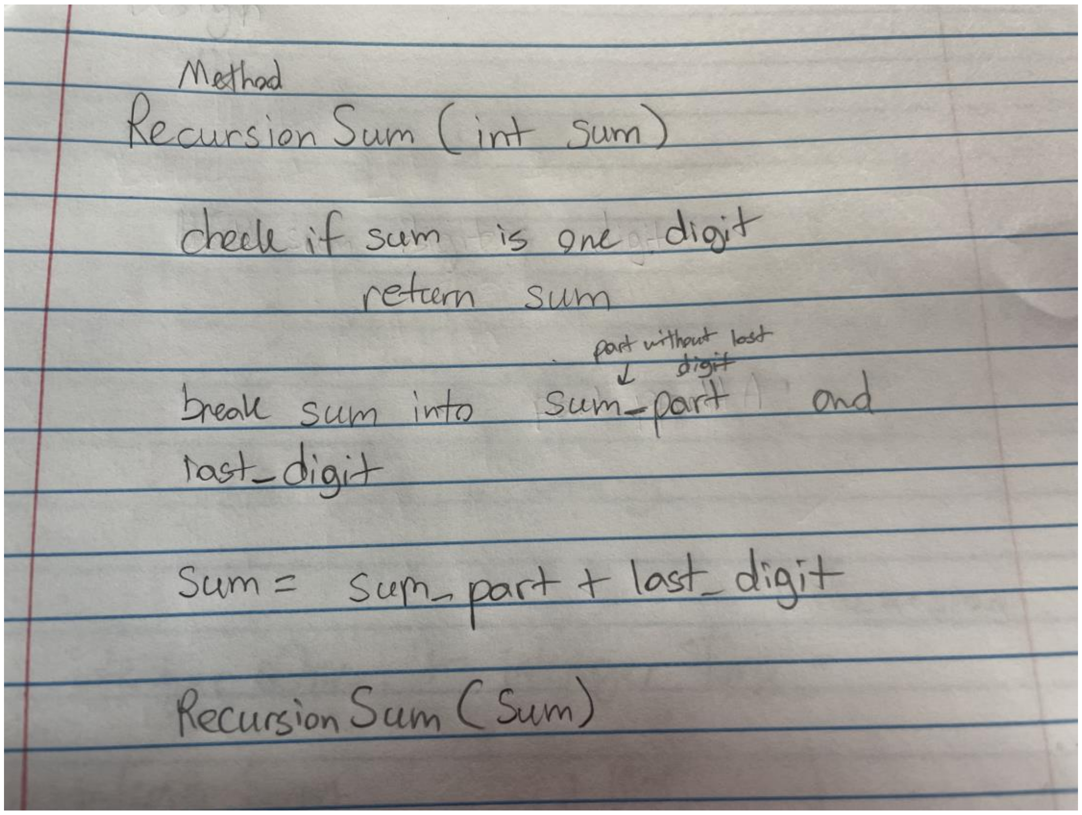
Akmal Arifov
Elijah Amador

Pseudocode & Design:



```
Main
  Loop until correct input/exit. Use Try/catch
  Ask user inputs for two int values
  Input Validate
          No alphabetical
          No special chars
          No incorrect data type
          No int number greater
              than Max of int
              divided by two
              rounded down
        No int number lower
              than zero
  Tell user to enter q to exit
  if user enters "exit"
          exit program
  store values in int1 and int2

  Sum = int1 + int2
  single_digit = RecursionSum(sum)
  Validate single_digit for overflow
          if overflow
          Output "result overflowed, try lower #"
  Output single_digit
```

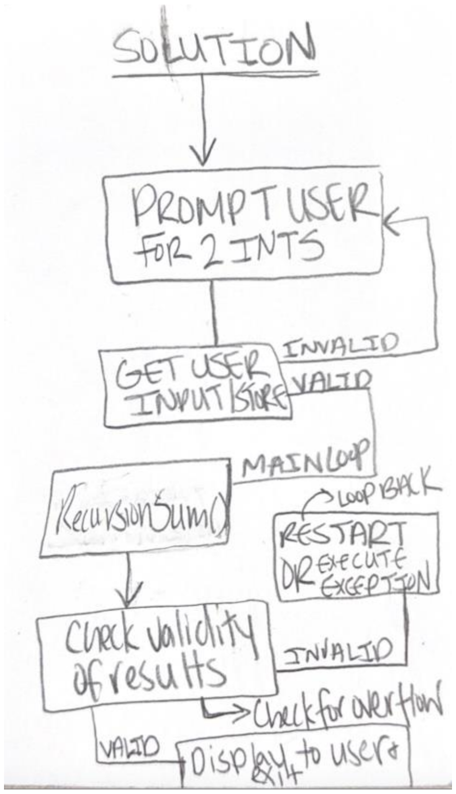
Akmal Arifov
Elijah Amador



Method

```
Recursion Sum (int sum)

    check if sum is one digit
        return sum

    break sum into sum_part and
    last_digit
                     part without last digit

    Sum = sum_part + last_digit

    Recursion Sum (Sum)
```

Design Flow Chart:


SOLUTION
PROMPT USER FOR 2 INTS
INVALID
GET USER INPUT|STORE
VALID
MAINLOOP
RecursionSum()
LOOP BACK
RESTART OR EXECUTE EXCEPTION
Check validity of results
INVALID
Check for overflow
VALID
Display to user & exit