Project 1
C# People Parser

Difficulty: easy
Estimated time: 2 - 4 hours

Task:

1. As input, take in 3 parameters: A .csv file, number of rows, target userid
2. As output,
   A. If the userid exists:
      1. Print the name associated to the userid to the terminal, and
      2. Save the following information to a .txt file: the random_string.

   B. If there is an error, and or if the userid does not exist,
      1. Print an error to the terminal describing what the error is

Example Outputs:

Proper example success run:
`myProgram input1.csv 4 IanAR1`

Terminal	Output File
`Name: Ian`	Hi There!

Proper Example Failed run, userid not found:
`myProgram input9.csv 55 billy`

Terminal	Output File (does not exist)
`userid 'billyb' not found, exiting`	

Proper Example Failed run, input arguments invalid:
`myProgram input99.csv -1.3 long-boi`

Terminal	Output File (does not exist)
`Input arguments invalid` `Format: p.exe <filename.csv> <N> <userid>`	

Requirements (Definition of Done):

1. Code should compile
2. Output should match above and work for any test case
   - Error logging and printing to console for any input and output; all errors should be caught.
3. Code should have a main function that calls helper functions.
4. No global variables or code snippets besides functions and imports.

Stretch Goals:

1. Dynamically allocate all data and data structures for optimal memory usage
2. No memory loss (free all allocated memory)

Input Details:
CSV FILE:

- The .csv file is a spreadsheet of name, userid, and a random string.
- The first line is the header, which is unimportant; it exists to tell the developer what the columns are named.
- Every line after the header are users with exactly three columns of information.
- The length of the file, including the header is specific as the number of rows in the input command line arguments.
- The .csv file will be a form of file called 'inputX.csv', where X is a single digit.
- The following input file names are valid:
  input2.csv, input0.csv, input9.csv
- The following input file names are invalid:
  Input2.csv, input .csv, input10.csv, input.csv, myfile.csv, input1.txt, input 3.csv
- The file name must not be hardcoded, but rather taken in on the command line as a variable

N:

- N can be any positive integer.
- The following formats are valid:
  0, 10000, 45, 2
- The following formats are invalid:
  -1, .5, -.0009

USERID DETAILS:

- userid can be any string
- The following formats are valid:
  'jw2004', 'billyB', 'test_1'
- The following formats are invalid:
  'my_password', 'Howdy there partner', 'small', '123'

CSV File Structure:

name	userid (char[6])	random_string
Ian	IanAR1	Hi There!
Jarod	jw2004	I love whiteboards
Michelle	nobody	I am no-one important
…	…	…

CSV Column details:

- Valid characters: alphanumeric (case sensitive), whitespace, symbols (_,-.!? … etc.)
- Invalid characters: non alphanumeric (no utf-8)
- IMPORTANT: userid contains extra rules:
  - userid can only be alphanumeric char[6] and contain no whitespace or symbols