

Objectives:

Use arrays/vectors, random number generators, command line arguments.

This card game requires a certain number of cards, n , where n is a triangular number. A triangular number is a number that represent the sum of a sequence: 1, 2, 3, 4, etc. For example, $21 = 1 + 2 + 3 + 4 + 5 + 6$. The first few triangular numbers are 1, 3, 6, 15, 21, 28, 36, 45, etc.

The Game:

You will start the game with an n value (default 45) for the number of cards. Divide the cards into a random number of piles.

Playing the game (Assuming $n = 45$):

Each round, take one card from each pile and create a new pile. Repeat the process until you have 9 piles of sizes 1, 2, 3, 4, 5, 6, 7, 8, and 9.

For example, if you start with 4 piles of 20, 15, 6, and 4, they would be transformed into 19, 14, 5, 3, and 4, etc.

Simulator:

Write a C++ program that simulates the game. Your program must obtain the number of cards (n) from the command line (see below). Assume the number of cards will never exceed 500. If the number is not provided, then use 45 as the default value of n . If the user, enters more than one value, print an error message and quit the program (return 0).

Start by producing a random number of piles. Make sure the total is n . Apply the steps outlined above repeatedly until the final configuration is reached. At each step, print the number of cards in each of the piles. The final configuration is reached when the number of cards in the piles are in sequence: 1, 2, 3, 4, 5, 6, 7, etc. Display the number of steps it took to reach the final configuration.

Hints:

- Store the number of cards in each pile into an array/vector of integers.
- Write a function (`isTriangular`) that returns `true` if a number `n` is a triangular number, `false` otherwise.
- To generate a random number use the function `rand()` in `<cstdlib>`
 - First seed the random number generator with the current time. Do this only once in the program (beginning of `main`)
 - `srand(time(NULL));`
 - To generate a random integer between 0 and 2147483647
 - `randomInteger = rand();`
- Write a function to print all the elements of the array
- Write a function that checks if the game is done by confirming the last sequence. You may want to sort the values to make it easier to check.
- To convert a string to an integer, use `stoi()` or `atoi()` function. For example,
`number = stoi(st);`

Parsing command line arguments

In C++ you can input data (strings) into your program on the command line (command line arguments). You can capture these data by adding two parameters to your main program as follows:

```
int main (int argc, char *argv[])
```

`argc`: (argument count) number of arguments on the line, including the name of the program

`argv`: (argument vector) list of c-style strings that represent all the arguments including the name of the program (two dimensional array of characters).

For example, executing the command:

```
./a.out 45
```

Assigns:

```
argc = 2
```

```
argv[0] will be "./a.out"
```

```
argv[1] will be "45"
```

Sample output (n = 45):

```
./a.out 5
Error: the number 5 is not a triangular number
```

```
./a.out 5 7
Error: too many parameters.
```

```
./a.out 10
number of cards: 10
8      2
7      2      1
6      3      1
5      3      2
4      3      2      1
Number of rounds: 5
```

```
./a.out
34      7      3      1
33      6      4      2
32      5      4      3      1
31      5      4      3      2
...
...
11      8      7      6      5      4      3      1
10      8      7      6      5      4      3      2
9      8      7      6      5      4      3      2      1
-----
Number of rounds: 28
```

Grading:

Programs that contain syntax errors will earn zero points.

Programs that do not include functions will also earn zero points.

Programs that use global variables other than constants will earn zero points.

Your grade will be determined using the following criteria:

- Correctness (35 points)
 - (7 points) Parsing the command line arguments and printing error messages
 - (7 points) Verifying the number of cards is triangular
 - (7 points) The initial number of cards in each pile is selected randomly
 - (3 points) The number of cards in each pile is printed repeatedly
 - (4 points) The number of rounds is printed
 - (7 points) Clarity and format of the output
- Style & Documentation (5 points)

Follow the coding style outline on GitHub:

<https://github.com/nasseef/cs2400/blob/master/docs/coding-style.md>