Janelle Boyd, Deshaun Crawford, Robert Morris

Operating Systems

October 16, 2016

**Game Algorithm**

The way our algorithm works is by using enumeration. We ask the players for both of their names, and then set a flag that sets the current player to False so we can alternate between players. Additionally, we have a variable named CHOICE that will keep track of each player’s choice when they pick one or two options. While the stack has been created and displayed for the user to see, we created a flag for the variable CHOICE. If the user picks 0, then they will be prompted to pick another number. If the user picks 2 or a number above two (Error checking) then 2 numbers will be removed from the stack; however, if the user chooses 1 or a number less than 1 then 1 number will be removed from the stack. The algorithm is constantly checking for when the stack has either 1 or no elements in it. Once the stack as 1 element, the last element is automatically assigned to the current player. Thus, the sums are calculated and the winner is selected!

**Extra:**

In order to always “win” the only thing the player should be concerned with is whether or not to go first. This can be decided by using the variable X which represents the turn such that none of the subsets of available numbers add up to X. The objective for the player would be to for their turn to end on round X. So if X is odd the player using the algorithm should play first otherwise they would let the opponent play first.

To determine the value for X the player using the algorithm should use Sieve. The way Sieve works is the player takes account of all the numbers from 0 to the sum of all the numbers on the table, and crosses off zero, one at a time, without replacement. When each number is taken off every number that is the same is taken off plus the previously crossed off number. When this is done for every number on the table, the smallest un-crossed off number is X