Name: Avnish Sengupta

UCLA ID: 705299746

OBSTACLES OVERCOME: -

* Ensuring that a pet or owner cannot leave before they have entered. This was done by making six extra helper functions that kept a count of the number of kittens, adult cats, children, adults, puppies and adult dogs were left behind if the animal string was over. If the number dipped below 0, then the rule had been violated.
* Ensuring that cats and dogs cannot enter or leave together in the animal park string. This was done by creating a complicated if-else statement structure that did not allow d or D to be surrounded by c or C and vice versa.
* Ensuring that pets were always followed by their owners and that owners could not enter the dog park alone. This was done by making an if-else statement that returned the result as false for isValidAnimalParkString function if a ‘+’ or ‘-’ character was not preceded by a ‘p’ or ‘P’.
* Ensuring that a owner or pet cannot leave without having either. This was done by creating an if-else condition that if a ‘+’ or ‘-’ was followed by a ‘p’ or ‘P’, then the result should be false.
* Ensuring that there are no leading or ending ‘+’s or ‘-’s. This was done with an if-else statement ensuring that the animal park string should always end with a ‘p’ or a ‘P’ and should begin with a ‘c’ or ‘C’ or ‘d’ or ‘D’.
* Ensuring that no undesirable characters were typed into the string. This was done by using a Switch case by switching on b- the variable that walked over the string in the for loop. If any desirable variable was found, the result would be true and the loop would break but if there are any undesirable characters then the loop would break and the result would be false.
* Keeping a count of the number of people, cats and dogs remaining in the park after the animal park string was over. This was done by counting in three cases: -
  + Creating a loop and adding any people, dogs or cats that entered before the first ‘+’ or ‘-’ character.
  + Creating another while loop and adding any people, dogs or cats that entered after every ‘+’ character and before every ‘-’ character.
  + Creating another while loop and subtracting any people, dogs or cats that entered after every ‘-’ character and before every ‘+’ character.

PSEUDOCODE: -

1. Declaring all the functions so that they can be called in any order in other function definitions.
2. Defining the isValidAnimalParkString function which takes string inputs.
   1. Using a for loop to walk across the string.
      1. Ensuring that no undesirable characteracters are present in the string using a Switch case.
      2. If the result were true: -
         1. Ensuring that ‘+’s or ‘-’s were only preceded by owners.
         2. Ensuring that pets are always followed by their owners.
         3. Ensuring that an owner cannot enter or leave without a pet.
         4. Ensuring that cats and dogs cannot enter or leave together.
      3. If any of these cases returned false, then break out of the loop.
   2. Ensuring that a pet or owner cannot leave before they have entered by calling declared functions (kittensLeft, CATSLeft, puppiesLeft, DOGSLeft, childrenLeft, adultsLeft) that have been defined later.
   3. Ensuring that the string cannot just be one character long.
   4. Ensuring that the string always begins with pets and ends with owners only.
3. Defining the catsLeft function: -
   1. Ensuring the program only runs if isValidAnimalParkString(animalparkString) returns true. Otherwise, it should return -1.
   2. Two helper functions are defined before this function-kittensLeft and CATSLeft: -
      1. kittensLeft
         1. Count the kittens that enter in the first segment in a loop placed inside a while loop in the animal park string before the first ‘+’ or ‘-’ string.
         2. Count the number of kittens that enter in the same variable from the first loop after every ‘+’ character till the next ‘-’ character.
         3. Count the number of kittens that leave in another variable in a while loop after every ‘-’ till the next ‘+’ character.
         4. Calculate the number of kittens left in a variable by subtracting the number of kittens leaving from all the kittens entering.
      2. CATSLeft
         1. Count the adult cats that enter in the first segment in a loop placed inside a while loop in the animal park string before the first ‘+’ or ‘-’ string.
         2. Count the number of adult cats that enter in the same variable from the first loop after every ‘+’ character till the next ‘-’ character.
         3. Count the number of adult cats that leave in another variable in a while loop after every ‘-’ till the next ‘+’ character.
         4. Calculate the number of adult cats left in a variable by subtracting the number of adult cats leaving from all the adult cats entering.
   3. The final return value is the sum of the return values of the two helper functions.
4. Defining the dogsLeft function: -
   1. Ensuring the program only runs if isValidAnimalParkString(animalparkString) returns true. Otherwise, it should return -1.
   2. Two helper functions are defined before this function-puppiesLeft and DOGSLeft: -
      1. puppiesLeft
         1. Count the puppies that enter in the first segment in a loop placed inside a while loop in the animal park string before the first ‘+’ or ‘-’ string.
         2. Count the number of puppies that enter in the same variable from the first loop after every ‘+’ character till the next ‘-’ character.
         3. Count the number of puppies that leave in another variable in a while loop after every ‘-’ till the next ‘+’ character.
         4. Calculate the number of puppies left in a variable by subtracting the number of puppies leaving from all the puppies entering.
      2. DOGSLeft
         1. Count the adult dogs that enter in the first segment in a loop placed inside a while loop in the animal park string before the first ‘+’ or ‘-’ string.
         2. Count the number of adult dogs that enter in the same variable from the first loop after every ‘+’ character till the next ‘-’ character.
         3. Count the number of adult dogs that leave in another variable in a while loop after every ‘-’ till the next ‘+’ character.
         4. Calculate the number of adult dogs left in a variable by subtracting the number of adult dogs leaving from all the adult dogs entering.
   3. The final return value is the sum of the return values of the two helper functions.
5. Defining the peopleLeft function: -
   1. Ensuring the program only runs if isValidAnimalParkString(animalparkString) returns true. Otherwise, it should return -1.
   2. Two helper functions are defined before this function-childrenLeft and adultsLeft: -
      1. childrenLeft
         1. Count the children that enter in the first segment in a loop placed inside a while loop in the animal park string before the first ‘+’ or ‘-’ string.
         2. Count the children when the string has a segment that begins with ‘-’ and ends with ‘+’, as the for-loop increments the index variable by 1 causing the program to miss the next ‘+’ character.
         3. Count the number of children that enter in the same variable from the first loop after every ‘+’ character till the next ‘-’ character.
         4. Count the number of children that leave in another variable in a while loop after every ‘-’ till the next ‘+’ character.
         5. Calculate the number of children left in a variable by subtracting the number of children leaving from all the children entering.
      2. adultsLeft
         1. Count the adults that enter in the first segment in a loop placed inside a while loop in the animal park string before the first ‘+’ or ‘-’ string.
         2. Count the adults when the string has a segment that begins with ‘-’ and ends with ‘+’, as the for-loop increments the index variable by 1 causing the program to miss the next ‘+’ character.
         3. Count the number of adults that enter in the same variable from the first loop after every ‘+’ character till the next ‘-’ character.
         4. Count the number of adults that leave in another variable in a while loop after every ‘-’ till the next ‘+’ character.
         5. Calculate the number of adults left in a variable by subtracting the number of adults leaving from all the adults entering.
   3. The final return value is the sum of the return values of the two helper functions.
6. The int main function is defined where: -
   1. isValidAnimalParkString function is run
      1. if it returns true: -
         1. catsLeft function is invoked.
         2. dogsLeft function is invoked.
         3. peopleLeft function is invoked.
      2. If it returns false: -
         1. Return -1.

TEST DATA: -

* DdDPPP+CcCpP-CcCPPP-DdDpP
  + To check if isValidAnimalParkString returns true for an animal park string that follows all the guidelines given.
* DdDPPP+CcCeP-CcCPPP-DdDP
  + To check if isValidAnimalParkString returns false for a string with undesirable characters hidden in between the string.
* asdf1ABC000:2-55
  + To check if isValidAnimalParkString returns false for a string with undesirable characters.
* +dp+cp
  + To check if isValidAnimalParkString returns false for a string which has a leading ‘+’ character.
* d p + c p
  + To check if isValidAnimalParkString returns false for a string with many empty spaces in between characters.
* -dp+dp
  + To check if isValidAnimalParkString returns false for a string which has a leading ‘-’.
* dp-CP
  + To check if isValidAnimalParkString returns false for a string that allows pets or owners to leave before they have entered.
* cCcDP
  + To check if isValidAnimalParkString returns false for a string which allows cats and dogs to enter or leave together.
* cP+dP-cdPP
  + To check if isValidAnimalParkString returns false for a string that has multiple segments but allows cats and dogs to enter or leave the park together.
* cpP-P
  + To check if isValidAnimalParkString returns false for a string that allows an owner to leave the park without a pet.
* cp+P-cpP
  + To check if isValidAnimalParkString returns false for a string that allows an owner to leave the park without a pet.
* dddddPcccccP
  + To check if isValidAnimalParkString returns false for a string that allows cats and dogs to enter the park together and for segments to be unseparated by ‘+’ or ‘-’ characters.
* ddDP+ccCP-
  + To check if isValidAnimalParkString returns false for a string that ends with a ‘-’ character.
* ddDP+cccP+
  + To check if isValidAnimalParkString returns false for a string that ends with a ‘+’ character.