

Programming Assignment 4: Dominoes

CECS 328

1 Deadline

4/15/2022 at 5 PM

2 Introduction

A domino is a game piece that has 2 numbers on it, one number on top and another underneath. Standard domino pieces have a number from 0 to 6, but these dominos are specially made and can have any nonnegative number in either the top or bottom position. Domino numbers are not required to be unique.

A family places a set of dominos on a table, one after the other, in a row (see picture) in such a way that the number on bottom of the domino in front is equal to the number on top of the domino behind. At night, the family goes to sleep and leaves the dominos on the table, and the family cat jumps on the table and scatters the dominos all over the floor.

In the morning, the kids wake up to find the dominos scattered all over the floor and look to you to put them back the way they were. Your goal in this assignment will be to arrange all the dominos in any configuration (the kids won't know the difference if it's not exactly the same) such that the number on bottom of the domino in front is equal to the number on top of the domino behind.

HINT: Consider a walk along the string of dominoes as being a change in state from one number to another. Think about a “state” as being which number you happen to currently be. Questions to ask yourself: Is it possible to reassemble the dominoes if there is exactly 1 number that occurs an odd number of times? Is it possible to reassemble the dominoes if there are 3 or more numbers that occur an odd number of times?

3 Your code

When considering the dominos as pairs, the first element of the pair is the top and the second is the bottom.



If you are writing the file in Java: StudentSolver.java should have a function with the header `public static ArrayList<int[]> solve(ArrayList<int[]> dominos)`

If you are writing the file in Python: studentsolver.py should have a function with the header `def solve(dominos):`

If you are writing the file in C++: StudentSolver.h should have a line with the header
`static std::vector<std::pair<int,int>> solve(std::vector<std::pair<int,int>>);`

4 Example

Consider the set of dominos: (1,1),(3,1),(2,1),(1,3),(2,2),(1,2)

One possible solution is: (1,3),(3,1),(1,1),(1,2),(2,2),(2,1)