1. Goal: Accomplish the following two missions and integrate them into one single program. Deduct 5 points for unfriendly interface!

(Input) Input File
Student Pairs (undirected graph): this binary file is named as pairs#.dat and sorted in ascending order of student id. The 3 attributes are as below.

- "sid1": an array of 16 characters for the first student.

- "sid2": an array of 16 characters for the second student.
- "weight": float, in (0, 1].

(Mission One) Build adjacency lists and save them into a file
Description: Keep all the student pairs as adjacency lists in ascending order of "sid1", each "sid1" corresponds to an adjacency list. The nodes on an adjacency list are in descending order of "weight".
Notice:
1. The input file contains a variable number of pairs, so the space of your data structures MUST be dynamically allocated. A fixed-size array is NOT allowed!
2. A node on adjacency lists MUST keep two pieces of information: "sid2" and "weight".
Output: Write each adjacency list in ascending order of "sid1" into a file, including "sid2" and "weight" for each node on adjacency lists.

(Mission Two) Traverse the graph of student pairs to find connected components
Input: (Only) The adjacency lists constructed in Mission One.
Description: Start depth-first traversals at arbitrary nodes to find each connected component in the graph of student pairs.
Output: Display on the screen all the connected components in ascending order of their sizes. For a connected component, show its set of student id's in ascending order.

2. Documentation
The content must include but is not limited to the following:
(1) Describe the deail about your way of implementing adjacency lists, then introduce another way of coding, and finally point out which way is better.

(2) Try to think about how the connected components in the graph of student pairs can be used. You MUST define the meanings of pair weights and connected component sizes in the application you mentioned.

一、題目：完成下列兩項任務，並將兩者整合成單一程式，提供的操作介面若不友善先扣5分。

（輸入）讀檔
學生配對(無向圖)：此二進位檔的檔名如pairs#.dat，每筆紀錄是依照對應的學號由小到大排序，共3個欄位如下：

-【學號sid1】第一位學生的學號以16個字元陣列表示
-【學號sid2】第二位學生的學號以16個字元陣列表示
-【配對權重weight】以浮點數float儲存，介於0~1之間的正實數

（任務一）建立相鄰串列並存檔
描述：將所有的學生配對存成相鄰串列，並依【學號sid1】由小到大排序，每個【學號sid1】對應一條相鄰串列，各串列上依照【配對權重weight】由大到小排序。
注意：　(1) 輸入資料是不固定的的筆數，資料結構必須以動態配置空間，直接用固定大小的

陣列者，一律零分！

(2) 相鄰串列上的每一個節點必須存放資訊：【學號sid2】和【配對權重weight】。

輸出：由小到大輸出每位學生的【學號sid1】及其相鄰串列至檔案，依序列出串列上每位學生的【學號sid2】及【配對權重weight】。

（任務二）走訪學生配對圖找出連通成分

輸入：(僅限)使用任務一建立的相鄰串列。

內容：從任一節點開始以深度優先走訪相鄰節點，找出學生配對圖中彼此分離的每個連通成分。

輸出：依照包括的學號個數由小到大輸出所有連通成分至螢幕，各連通成分均由小到大列出所包括的學號。

二、說明文件

內容必須包含但不限於以下幾項：

1. 詳細解說自己實作相鄰串列的程式寫法，並描述另一種實作相鄰串列的程式寫法，再明確指出哪一種寫法較佳。

2. 嘗試思考學生配對圖的連通成分可以有什麼應用？須明確指出配對權重和連通成分大小在所描述應用上的意涵？