## 1. Basic Requirements

(a) Only upload source codes in .cpp/.c/.h/.hpp with comments that can be successfully compiled, the file name should be "DS2ex4_*team-id_student-id1_student-id2*". Deduct 5 points immediately for any violation!

(b) Upload only one copy for each team and there must be the name and student id of each member at the first few lines in your codes. Deduct 5 points for duplicate or any missing information!

(c) Codes that are non-C/C++ or unable to be successfully executed will be treated as "Unfinished" and get no point.

## 2. Goal

***Accomplish two missions and integrate them into one. Deduct 5 points for unfriendly interface!***

### (Mission One) Transform a text file into a binary file

Input: Read a text file with the 9 fields including "sid", "sname", 6 fields of "score", and "average".

Steps: Store each of the 9 fields of "score" as unsigned char and store "average" as a float.

- "sid": an array of 10 characters.
- "sname": an array of 10 characters.
- each of the 6 fields of "score": unsigned char.
- "average": float.

Output: Store each record after the above transformation as a structure and then save as a binary file with a name extension of .bin. The file size must be consistent with the output of DEMO program.

### (Mission Two) Build a hash table by Linear probing

Hash Function: (1) hash table size = the minimum prime number that is larger than 1.2 times the number of records.

(2) hash(key) = (the product of the ASCII code of each character in sid) % hash table size. *Ref. URL: https://simple.wikipedia.org/wiki/ASCII*

Input: Read the binary file built in Mission One.

Steps: (1) Add the records one by one and use "sid" to build a hash table X by linear probing. A hash entry keeps at most one record, including all the 15 fields and an extra field "hvalue" to store the hash value.

(2) Based on the hash table, calculate the average numbers of comparisons for searching non-existed keys and existed keys, respectively.

Output: (1) Save all the records in the hash table from top to bottom as a text file with a name extension of .tab, including the fields "hvalue", "sid", "sname" and "average" for each hash entry.

(2) Output the average numbers of comparisons on screen. These numbers must be consistent with the output of DEMO program.

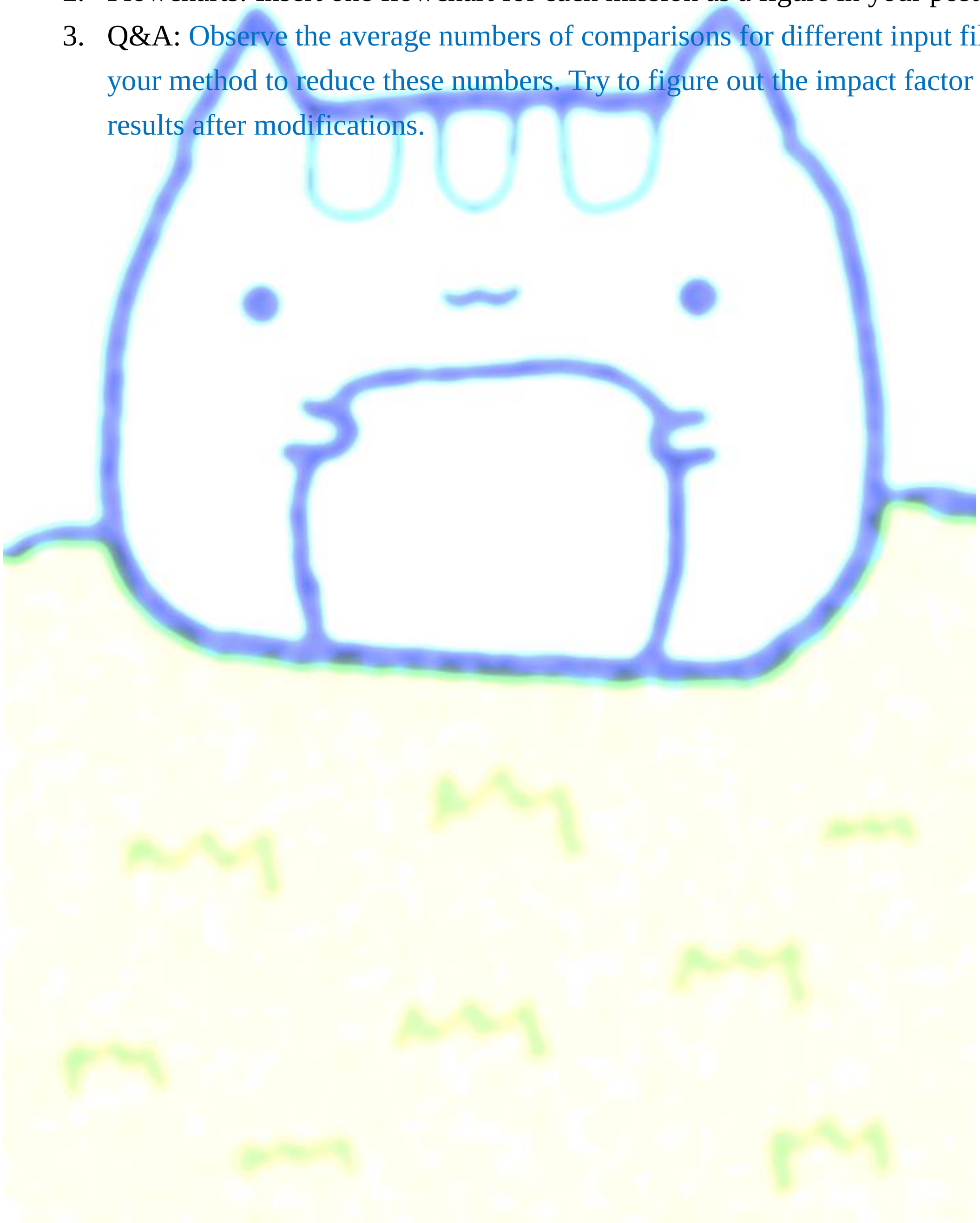
**3. Flowcharts of two missions & Documentation**

(a) It consists of two stages: two flowcharts during on-machine exercise, and a report for codes before the DEMO.

(b) Before the discussion board is closed, each team MUST have had a post in order to be arranged for the DEMO.

(c) The content must include but not limited to the following:

1. Introduction: Brief by text the main goal, assumptions, difficulties you encountered and the solutions. Do NOT copy anything directly from here.
2. Flowcharts: Insert one flowchart for each mission as a figure in your post.
3. Q&A: Observe the average numbers of comparisons for different input files and modify your method to reduce these numbers. Try to figure out the impact factor and the execution results after modifications.



## 一、基本需求

(a) 只上傳可成功編譯的原始碼(.cpp/.c/.h/.hpp)含註解、檔名請用「DS2ex4_*分組編號_學號1_學號2*」，違反任何一項先扣 5 分！

(b) 以組為單位只上傳一份，程式碼開頭幾行註解必須要有整組每位同學的中文姓名和學號，多傳一份或資訊不完整就扣 5 分！

(c) 非 C/C++程式 或 無法成功執行 一律視為「未完成」並以零分計！

## 二、題目

***完成兩項任務，將二者整合在一個簡易選單下，未整合或介面無法連續執行先扣 5 分。***

### （任務一）文字檔轉存二進位檔

輸入：讀入一個.txt 文字檔，其欄位包括：【學號 sid】、【姓名 sname】、(6 個)【分數 score】、【平均分數 average】，共 9 個字串欄位。

步驟：將 6 個分數欄位都改以整數(unsigned char)型態儲存，平均分數改以浮點數(float)儲存。

- 【學號 sid】以 10 個字元大小的陣列儲存。
- 【姓名 sname】以 10 個字元大小的陣列儲存。
- (6 個)【分數 score】各自以整數 unsigned char 型態儲存。
- 【平均分數 average】以浮點數 float 儲存。

輸出：以 struct 結構儲存上述轉換後的每一筆學生資料，改以二進位格式儲存成一個同名但改以.bin 為延伸檔名的新檔，檔案大小必須和範例程式的輸出一致。

### （任務二）以線性探測 Linear probing 建立雜湊表

雜湊函數：(1)雜湊表大小 ＝ 大於 1.2 倍資料總筆數的最小質數。

(2)只限用函數：hash(key) =（學號每個字母對應的 ASCII 編碼相乘） 除以 雜湊表大小 取餘數。 *參考網址*：*https://simple.wikipedia.org/wiki/ASCII*

輸入：任務一建立的二進位檔。

步驟：(1)依序逐筆讀取檔案後，採用線性探測以【學號 sid】建立雜湊表，每個雜湊位址只放一筆資料，儲存每筆資料的所有欄位及額外的【雜湊值 hvalue】。

(2)基於雜湊表，計算搜尋不存在值（除以雜湊表大小）及搜尋現存值（除以現存資料筆數）的平均比較次數。

輸出：(1)依序逐筆輸出雜湊表內的所有資料至同名但改以.tab 為延伸檔名的文字檔，顯示每個雜湊位址內資料的【雜湊值 hvalue】、【學號 sid】、【姓名 sname】及【平均分數 average】。

(2)輸出搜尋不存在值及搜尋現存值的平均比較次數至螢幕，這些次數必須和範例程式的輸出一致。

## 三、流程圖和程式說明文件

(a) 分為兩個階段，上機練習時要繳交兩張流程圖，機測前要繳交程式說明文件。

(b) 各組必須在看板關閉期限以前完成貼文，才會被排入機測。

(c) 貼文內容必須包含但不限於以下幾項：

1. 簡介：以文字簡述程式主旨，假設，遇到的困難和解法，勿直接剪貼題目字句！
2. 流程圖：每項任務各一張流程圖，以插圖放入貼文之中！
3. 答問：觀察不同輸入檔的平均比較次數，並修改方法以減少平均比較次數，試推測影響因素及修改後的執行結果。