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

(Input) File & Function
Input File: Read a binary file with the 15 fields.
- "sid": an array of 16 characters.

- "sname": an array of 16 characters.

- each of the 12 fields of "score": unsigned char.
- "average": float.

Hash Function: (1) hash table size = the minimum prime number that is larger than the number of records.
(2) hash(key) = (the product of the ASCII code of each digit in sid) % hash table size.
Ref. URL: https://simple.wikipedia.org/wiki/ASCII
Step Function: (1) step(key) = maximum step - ((the product of the ASCII code of each digit in sid) % maximum step).
(2) maximum step = the minimum prime number that is larger than (the number of records divided by 3).

(Mission One) Build a hash table X by linear probing, output the result to the screen and a file
Description: (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 X, calculate the average numbers of comparisons for searching unexisted keys and existed keys, respectively.
Ouput: (1) Output the records in the hash table from top to bottom as a text file, including the fields "hvalue", "sid", "sname" and "average" for each hash entry.
(2) Output the average numbers of comparisons on screen.

(Mission Two) RBuild a hash table Y by double hashing, output the result to the screen and a file
Description: (1) Add the records one by one and use "sid" to build a hash table Y by double hashing. 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 Y, calculate the average numbers of comparisons for searching unexisted keys and existed keys, respectively.
Ouput: the same as Mission One...

2. Documentation
The content must include but is not limited to the following:
(1) Use at least five sample data to get the average numbers of comparisons and then write down the results and your comments in the post.

(2) Modify the step function by yourself. If the average number of comparisons is reduced, attach at the end of your post the results and your own step function.

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）讀檔及函數
讀檔：讀取前一個練習的二進位檔，共15個欄位。

-【學號sid】和【姓名sname】分別以16個字元char陣列儲存。

- (12個)【分數score】各自以整數unsigned char儲存。

-【平均分數average】以浮點數float儲存。

雜湊函數：(1)雜湊表大小 = 大於資料總筆數的最小質數。

(2)只限用函數：hash(key) =（學號對應的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

步階函數：(1)只限用函數：step(key) = 最高步階 -（（學號對應的ASCII編碼相乘）除以 最高步階 取餘數）。

(2)最高步階 = 大於(資料總筆數/3)的最小質數。

（任務一）以線性探測建立雜湊表X，輸出結果至螢幕及寫檔

描述：　(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X，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X，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。

（任務二）以雙重雜湊建立雜湊表Y，輸出結果至螢幕及寫檔

描述：　(1)依序逐筆讀取檔案後，採用雙重雜湊以【學號sid】建立雜湊表Q，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Q，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：同任務一。

二、說明文件

內容必須包含但不限於以下幾項：

1. 採用至少五種不同筆數的資料求出各自的平均比較次數，在貼文中附上兩個任務的比較結果及評論。
2. 嘗試自行修改步階函數，若有效減少平均比較次數，可在貼文結尾附上數據和自行修改的步階函數。