**計算機概論與程式設計 – Take-home Final**

**2023 JAN**

You have to turn in programs for each sub-question. (Total 120 points)

You have to write comments for important statements to explain how it works.

1. Text processing programs.

1-a) Write a word-count program that can read a paragraph (ending with a blank line) and compute its number of alphabet characters and number of words (ignore space and any other punctuations). (8pts)

**This is a book.**

**output:**

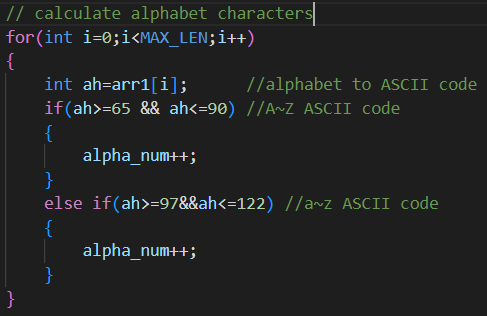
**11 4**

**Ans:**

**這一題最主要的概念就是**

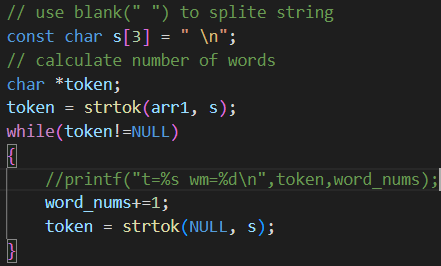
* + - 1. **用ascii碼判斷是否為英文字。**

**Ascii碼 A~Z:65~90，a~z:97~122。**



* + - 1. **用 空白 和 ”\n” 進行字串分割，然後計算word數量。**

**主要用到strtok套件進行操作。**



1-b) Write a program that reads a paragraph and performs correct capitalization for each word. (8pts)

**This is a bOOK.**

**output:**

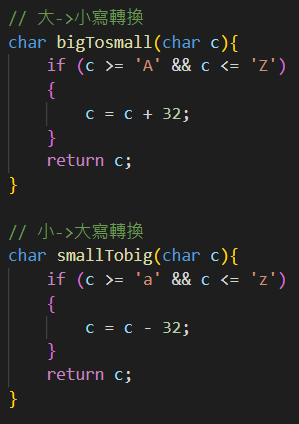
**This Is A Book.**

**Ans:**

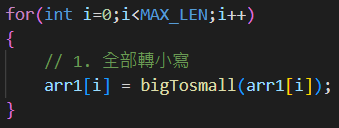
**這一題主要概念為:**

* + - 1. **先全部轉成小寫。**

**我先定義大小寫轉換**

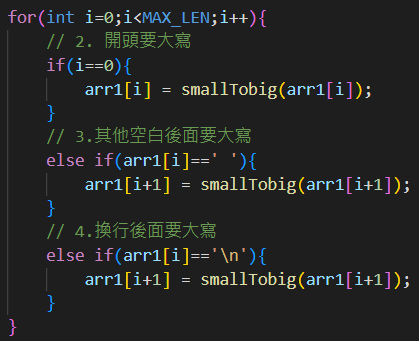


**然後全部轉小寫**



* + - 1. **如果是第一個字元，那字元一定是大寫!**
      2. **如果字元遇到空白，那下一個字元一定是大寫!**
      3. **如果字元遇到”\n”，那下一個字元一定是大寫!**

**然後根據條件2~4把小寫轉大寫!!**



1-c) word replacement: Now we allow a new operations: **replace old new** to replace all occurrences of **old** words by **new** word. (12pts)

**This book is a good book.**

**(blank)**

**replace book tiger**

**replace good bad**

**output:**

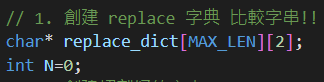
**This tiger is a bad tiger.**

**Ans:**

**這一題主要概念為:**

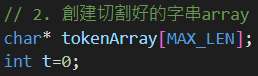
* + - 1. **要創建一個對照字典，讓字元可以對照，然後取代，所以我創一個2D的array，如下表:**

|  |  |
| --- | --- |
| **Old pattern** | **New pattern** |
| **Book** | **tiger** |
| **Good** | **bad** |

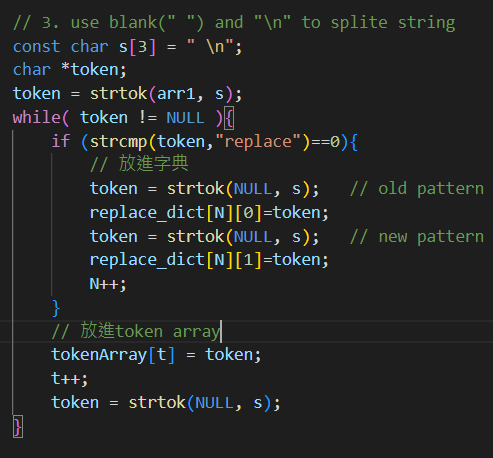


* + - 1. **因為標點符號也要留下來，我也有創一個token array放切割好的字串(有標點符號)，例如:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **This** | **book** | **is** | **a** | **good** | **Book.** |



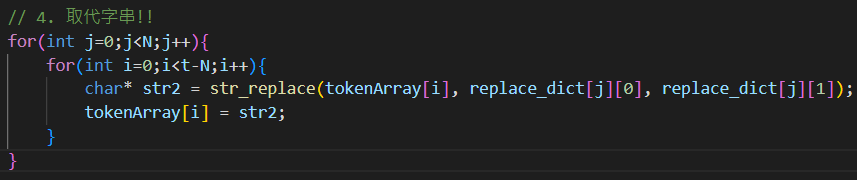
* + - 1. **然後使用 空白 切割字串，並將字串放進字典和token array。**



* + - 1. **Replace的方法主要就是利用strstr套件找到要取代的字串的起始位址index，然後將old pattern依照長度填入new pattern。詳細步驟如下圖所述。**



**然後將token array裡面的字串一個一個拿出來和字典比對並取代。**



1-d) Write a program that reads a paragraph and compute the occurrences of each word. Order the words by frequency (by the word length if the same frequency). Print out the top 5 frequent words. (You should be able to ignore the upper case or lower case.) (15pts)

**The red book is the most popular book in the market.**

**output:**

**the 3**

**book 2**

**popular 1**

**market 1**

**most 1**

**Ans:**

**概念上為**

**先創一個字典，為2D矩陣。用來第一行放字，第二行放字數。**

|  |  |
| --- | --- |
| **word** | **Word num** |
| **the** | **0** |
| **Book** | **0** |
| **Popular** | **0** |
| **most** | **0** |

**利用空白和標點符號先切割字串。**

**如果word有在字典裡，那字典相對word num就加1。**

**如果word沒有在字典裡，那就在字典填入word。**

**最後依照大小順序排列word num，印出前5多的字&字數。**

1. Write a sort program that can organize a given series of scores (ending with 0) and output them in a descending order. For each program, you also have to show the execution time to compare your program efficiency.

2-a) Implement by an array: Read the series of scores into an array, use a **largest()**to find the largest number, then use a **swap()** function to swap two integers so that the entire array is sorted (as we did in the mid-term exam). Output the numbers to stdout.

Your program should consider the count of the score series is unknown. The max size you can use for **malloc()** is 4KB each time. If not sufficient, you can double it each time. If you use **realloc** on size overflow, you should avoid dangling problem. (10pts)

But it costs lots of copy. You will get extra bonus if you use one **malloc()** as a chunk to store 1K numbers (never **free()**) and maintain another super pointer array for multiple chunks. You can implement an **index(i)** to get the exact the ith number pointer.

**2 14 5 90 2 40 0**

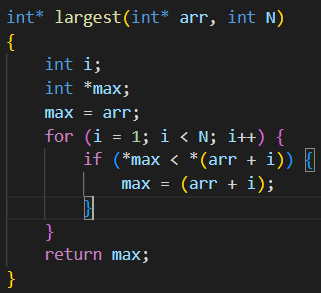
**output:**

**90 40 14 5 2 2**

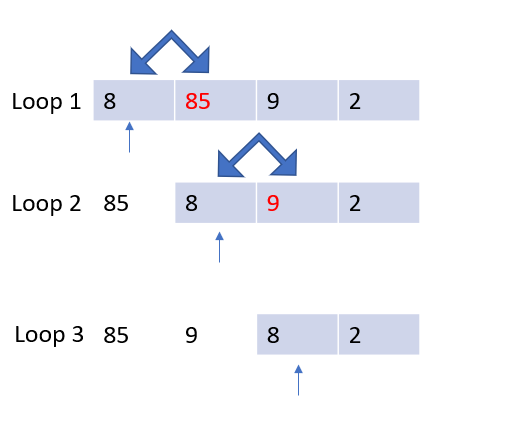
**Ans:**

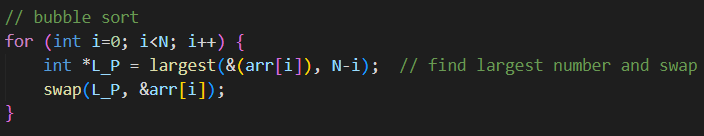
**主要概念為:**

* + - 1. **使用之前的期中考試的largest，回傳最大值的pointer。**

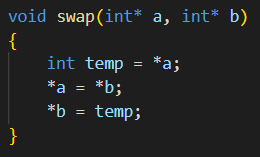


* + - 1. **然後模擬bubble sort直接一個for 迴圈每次都將最大的值，往前交換。步驟如下圖所述，注意:丟進去找最大值的array會越來越小(N-i)!**





* + - 1. **交換(swap)也是用pointer直接交換。**



2-b) Sort numbers in an array: You have to implement **add\_num()** and **del\_num()** functions to deal with the number array always in a descending order. Now when the program reads each number initially, you perform many **add\_num()** operations until finding 0. In the input, we also allow two operations: **add *nn nn*** … and **del *nn*** ***nn*** … to insert and delete any number from the series. One operation appears in a line only. Again if you implement a chunk mechanism, it can save more execution time. (12pts)

**2 14 5 90 2 40 0**

**add 86 50 1**

**del 40 5**

**add 55**

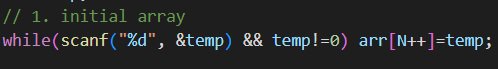
**output:**

**90 86 55 50 14 2 2 1**

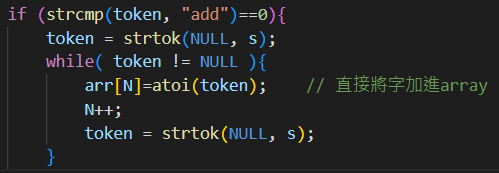
**Ans:**

**主要概念為:**

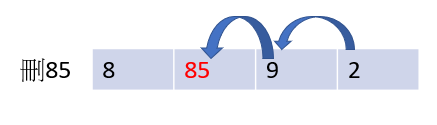
* + - 1. **創一個array放initial 的數據值。**

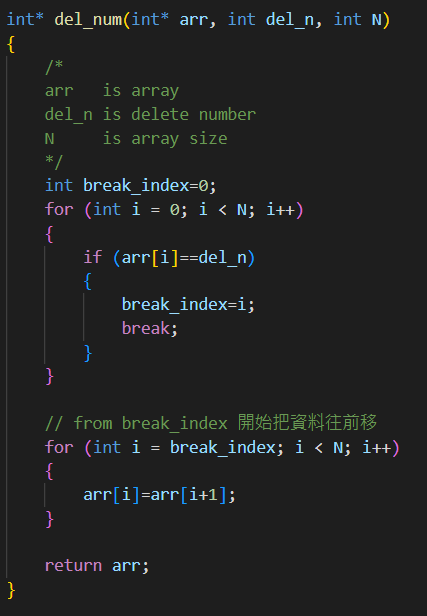


* + - 1. **然後切割字串時，先判斷是否為”add”，如果是就將add後的數字，放進array裡面。所以我就沒有特別寫add\_num的function了。**

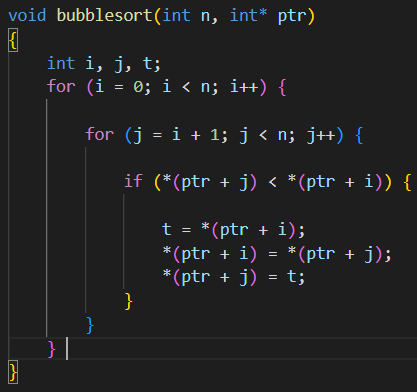


* + - 1. **如果是”del”，就去array裡面一個一個找，如果找出符合，就記錄下欲刪除的值的index，並把它後面的值都往前移一格。**





* + - 1. **排序部分其實就是使用bubble sort，但是這次需要兩個for loop迴圈處理!!**



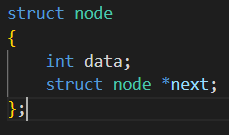
2-c) Implement by linked list: the same sort functionality as **1-b** program. We also support two operations: **add *nnn*** and **del *nnn*** to insert and delete any number from the series. (Hint: You have to use a double-pointer mechanism for **add\_num()** and **del\_num()** functions.)

**In your final documentation, compare the execution time of 1-a, 1-b and 1-c (or chunk mechanism). Give your explanation.** (15pts)

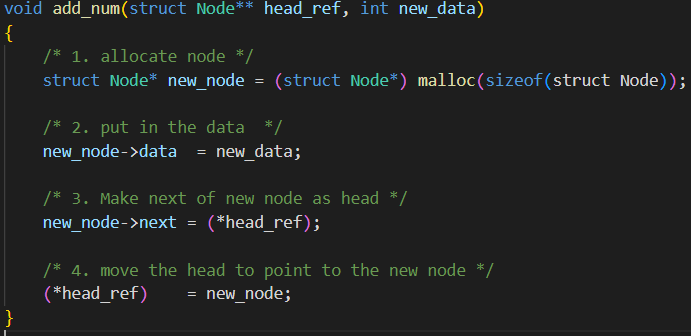
**Ans:**

**主要概念如下:**

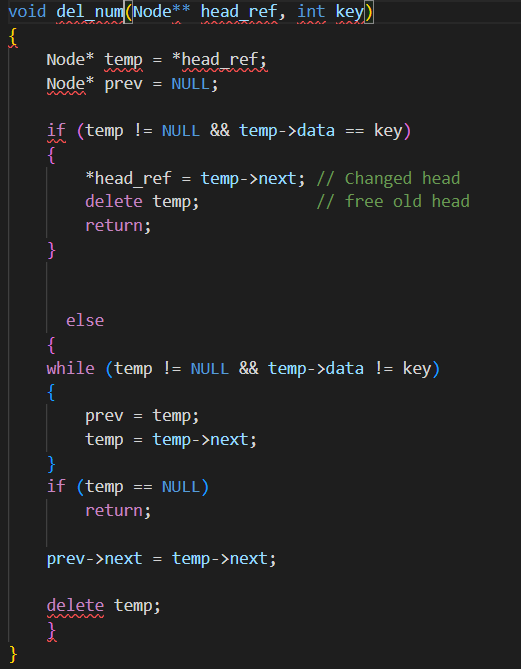
* + - 1. **一樣創一個linked list的struct放initial 的數據值。**

****

* + - 1. **一樣切割字串，如果遇到”add”就將值放進link list尾部。**



* + - 1. **如果遇到”del”就去link list裡面一個一個搜尋相對應值，並刪除。**



1. Misc and bonus programs.

3-a) Write a program that can read a date format yyyy/mm/dd or ROC yyy/mm/dd (中華民國紀元). Then print it in a form of day name in the week, month name, date and year (AD). The input range is only between 2021/1/1~2023/12/31. Just ignore the lunar year. (10pts)

**January 9, 2023 (Monday)**

**January 8, 2023 (Sunday)**

**December 31, 2022 (Saturday)**

**output**

**2023/1/9**

**ROC 112/1/8**

**ROC 111/12/31**

**input**

**Ans:**

**主要概念如下:**

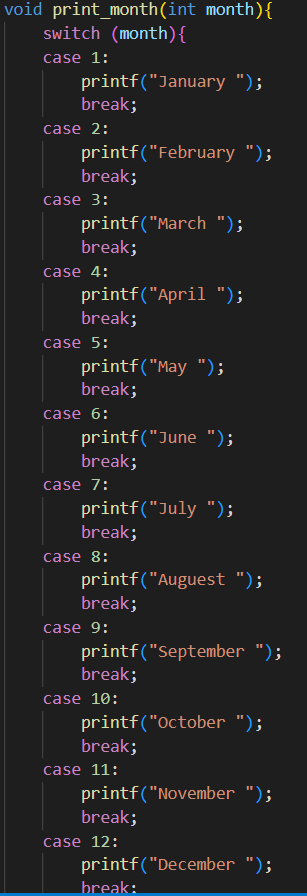
* + - 1. **如果開頭是’R’，就是ROC yyyy/mm/dd格式，用ROC專門的token。**

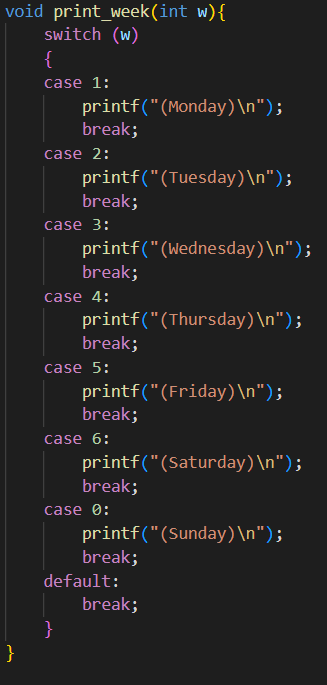


* + - 1. **如果是普通的就直接token。**

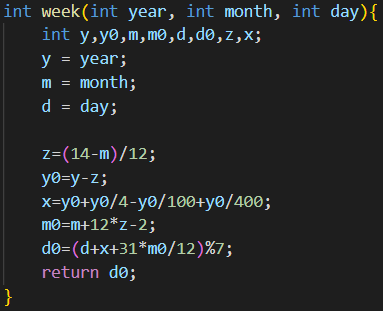


* + - 1. **月、星期幾就用case switch去做替換。**





* + - 1. **重點是week的計算是參考基姆拉爾森計算公式。可以用year/month/day找出week。**



3-b) A valid password has the following requirements. For given multiple lines that contain one word. Print out the validation of each password. (10pts)

1. Password must contain at least one upper case;
2. Password must contain at least two lower case letter letters and all have to be **strictly inside** (i.e. not as the first or the last character);
3. Password must contain at least one digit [0-9] and all digits have to be **strictly inside**;
4. Must contain at least one special character from the set { '@', '!', '%', '&', '\*' };
5. Password must be at least 8 characters in length, but it can be longer.

**yes**

**no**

**no**

**output**

**BjdaA00C!**

**U\_R\_kill@final0109**

**Ha!ha!ha!**

**input**

**Ans:**

主**要概念如下:**

**a) 其實主要就是利用前面ASCII碼A~Z:65~90，判斷大寫出現次數。**

**b) 其實主要就是利用前面ASCII碼a~z:97~122，判斷小寫出現次數。只是要注意從index:1~N-2,頭尾不算。**

**c) 其實主要判斷是否為int。只是要注意從index:1~N-2,頭尾不算。**

**d) 這其實也只是ASCII碼判斷，'@'=64, '!'=33, '%'=37, '&'=38, '\*'=42。**

**e) 這其實只是算字元數，字元數要>8。**

3-c) Write a program that can read a digit string [0~9] and also a few following patterns. Your job is to determine if the pattern is a substring of the given input string and also give the string position (starting from 0). (5pts)

**2023010912345654321**

**12345**

**20320109**

**3210**

**input**

**Yes 8**

**No x**

**No x**

**output**

**Ans:**

**主要概念如下:**

* + - 1. **主要為比較子字串，那就可以參考q2字串取代的部分，使用套件strstr直接判斷是否有相同，然後output index。**



3-d) The same function as 3-c. The decision becomes checking “**anagram of secret**” (a new number is created by rearranging every single digit found in the given patterns), the same as we did in past mid-term exam.(5pts)

**Yes 8 14**

**Yes 0**

**Yes 2**

**output**

**2023010912345654321**

**12345**

**20320109**

**3210**

**input**