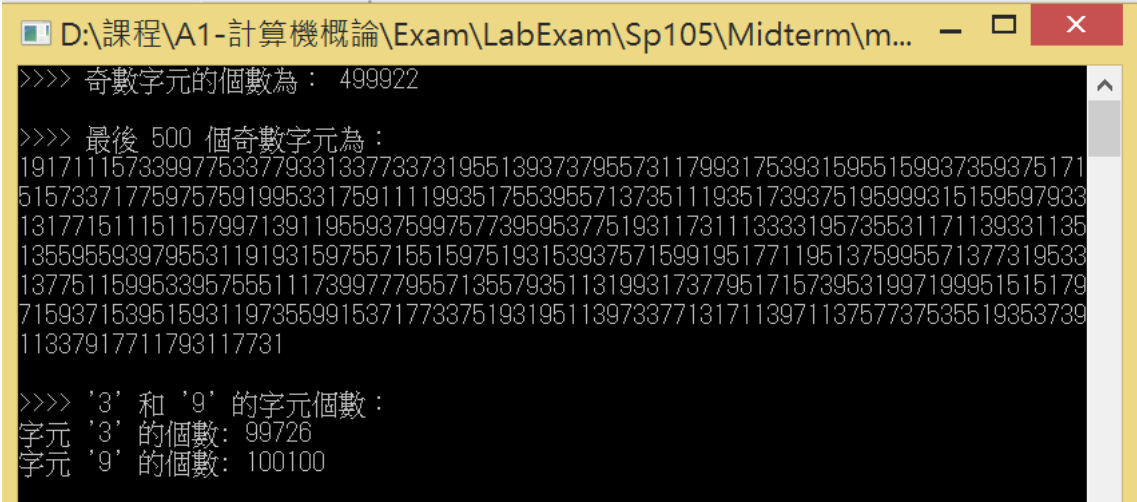


一〇四學年度 第二學期 資訊一甲
計算機概論實習(二) 期中上機考試

共三題，第 n 題請使用檔案名稱 “**mexam_dxXXXXXX_n.c**”，“**dxXXXXXX**” 為你的學號。同時，請在每個程式原始碼的第一行，以**註解方式**寫上你的學號及姓名。未使用規定之檔名或未寫上學號及姓名者，本次成績將被扣 **10%** 的分數。當你作答完畢後，請將程式的原始碼 (**source code**, 即 **mexam_dxXXXXXX_n.c** 檔) 上傳到教師機台，第三題請同時上傳影像檔 “**sunset.bmp**”。

1. (30 points) 你可使用 **mexam_skeleton_1_sp.c**, 並將此程式架構改名為 **mexam_dxXXXXXX_1.c**。寫一個 C 語言的程式，使用 0 為隨機數的種子，產生 1,000,000 個 0 到 9 的整數，產生數字之同時將其改為數字字元 (numeral character)，最後在這串字元之後加上 '\0' 字串結束 (end-of-line) 符號，使其成為一個長度為 1,000,000 的數字字串 (numeral string)。完成下列步驟的程式碼：
 1. 將偶數 (even) 的數字字元移除。
 2. 輸出處理後奇數字元的字串長度。
 3. 列印最後 500 個奇數字元的字串。
 4. 計算並輸出 3 的倍數奇數字元的個數，即是 '3' 和 '9' 的字元個數。

程式執行範例：



```
>>>> 奇數字元的個數為： 499922

>>>> 最後 500 個奇數字元為：
19171115733997753377933133773373195513937379557311799317539315955159937359375171
51573371775975759199533175911119935175539557137351119351739375195999315159597933
13177151115115799713911955937599757739595377519311731113333195735531171139331135
13559559397955311919315975571551597519315393757159919517711951375995571377319533
13775115995339575551117399777955713557935113199317377951715739531997199951515179
71593715395159311973559915371773375193195113973377131711397113757737535519353739
11337917711793117731

>>>> '3' 和 '9' 的字元個數：
字元 '3' 的個數： 99726
字元 '9' 的個數： 100100
```

(續下一頁)

2. (30 points) 你可使用 `mexam_skeleton_2_sp.c`，並將此程式架構改名為 `mexam_dxXXXXXX_2.c`。寫一個 C 語言的程式以產生字串的排序單向連結表 (sorted single-linked list of strings)，連結表的節點資料結構、節點指標型態、及連結表型態

```

struct Node {
    char data[21];
    struct Node* next;
}; // 單向連結表的節點資料結構
typedef struct Node* NodePtr; //單向連結表節點指標型態。
typedef NodePtr List; // 定義單向連結表型態。

```

此程式的運算步驟如下：

1. 輸入一個長度不超過 20 的英文或數字字串，將此字串改為全部大寫的英文字母；
2. 如果輸入的字串不是“000”時，
 - 2.1 將此字串插入**排序**單向連結表；
 - 2.2 繼續步驟 1；
3. 如果讀入的字串是“000”時，結束迴圈；從第一個節點開始，列印排序連結表所有節點的字串資料。

程式碼中的字串運算，必須使用 `<string.h>` 函式庫的副程式。至少測試下列程式執行範例的字串：

```

輸入一個長度小於 20 的英文字串 ("000" 即停止) : Department
輸入一個長度小於 20 的英文字串 ("000" 即停止) : of
輸入一個長度小於 20 的英文字串 ("000" 即停止) : Information
輸入一個長度小於 20 的英文字串 ("000" 即停止) : Engineering
輸入一個長度小於 20 的英文字串 ("000" 即停止) : and
輸入一個長度小於 20 的英文字串 ("000" 即停止) : Computer
輸入一個長度小於 20 的英文字串 ("000" 即停止) : Science
輸入一個長度小於 20 的英文字串 ("000" 即停止) : C
輸入一個長度小於 20 的英文字串 ("000" 即停止) : Programming
輸入一個長度小於 20 的英文字串 ("000" 即停止) : Language
輸入一個長度小於 20 的英文字串 ("000" 即停止) : 000

列印排序連結表所有節點的字串資料 :
AND, C, COMPUTER, DEPARTMENT, ENGINEERING, INFORMATION, LANGUAGE, OF, PROGRAMMING, SCIENCE,

```

(續下一頁)

3. (40 points) 你可使用 `mexam_skeleton_3_sp.c`, 並將此程式架構改名為 `mexam_dxxxxxx_3.c`。這個程式作業是有關影像處理。一個點陣圖 (bitmap image) 的檔案格式可分為檔頭 (file header) 和影像資料 (image information) 兩部份, 其資料結構如 `mexam_skeleton_4.c` 中 `typedef struct BitmapHeader` 的定義所示。

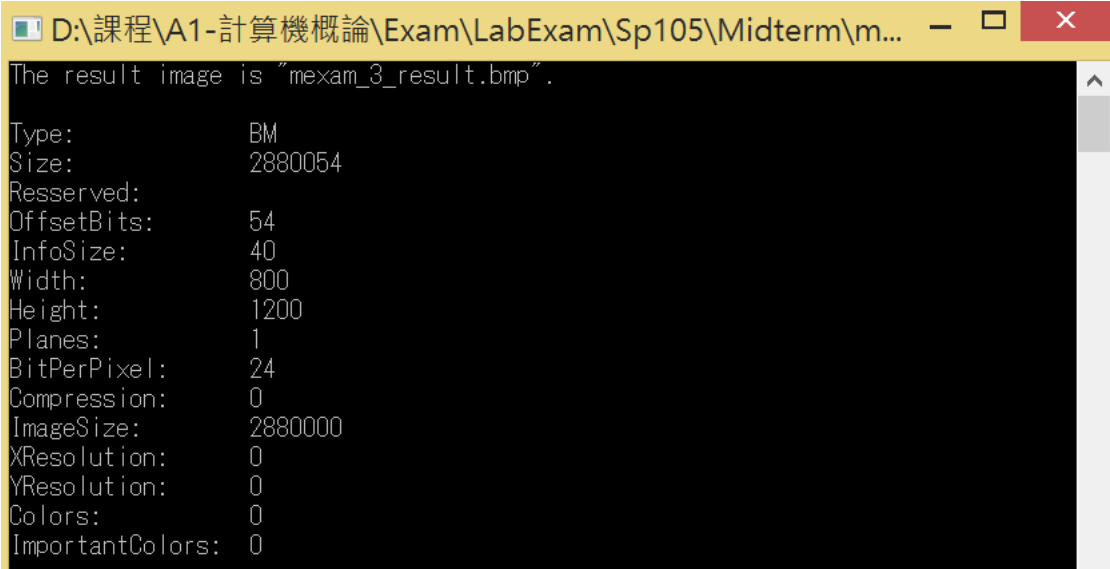
檔頭及影像資料結束後, 為一個色盤區域 (palette), 其大小為 `OffsetBits-InfoSize-14` 個位元組; 其後即為像素區域, 大小佔用 `ImageSize` 個位元組, 此二區域的資料型態可宣告為 `unsigned char` 的指標。像素區域的儲存方式如下:

1. 像素的儲存是以列為主, 自左而右, 自下而上。即最後一列儲存在像素區域的最前面。
2. 每個像素佔用三個位元組代表紅、綠、藍 (RGB) 各佔一個位元組, 其順序為藍色 (B)、綠色 (G)、然後紅色 (R)。
3. 每一列的位元組數必須是 4 的倍數。如果每列實際像素的總位元組不是 4 的倍數, 則在結尾填入 0 的資料。

寫一個 C 語言的程式以執行下列步驟:

1. 讀入一個彩色影像點陣圖檔案, 檔名為 “`sunset.bmp`”
2. 將這張影像產生依其水平軸的中心做上下鏡射轉換, 再將鏡射轉換的影像黏貼在原影像之下, 使其上下高度擴充成兩倍的影像。
3. 將處理後影像的點陣圖寫到硬碟, 檔名為 “`mexam_3_result.bmp`”: 且將影像檔案檔頭和資料部份的內容到顯示器上。

完成 “`mexam_skeleton_3_sp.c`” 中第 86 行至第 93 行未完成的程式碼。請同時上傳影像檔 “`sunset.bmp`” 程式執行範例:



```
The result image is "mexam_3_result.bmp".
Type:          BM
Size:          2880054
Reserved:      0
OffsetBits:    54
InfoSize:      40
Width:         800
Height:        1200
Planes:        1
BitPerPixel:   24
Compression:   0
ImageSize:     2880000
XResolution:   0
YResolution:   0
Colors:         0
ImportantColors: 0
```

輸入的影像圖 “`sunset.bmp`” (40% 縮小圖):



輸出的轉換後之影像 “mexam_3_result.bmp” (40% 縮小圖):

