

1. 隨身碟、手機、書包放教室前後。
2. 程式須編譯成功可執行，並瞭解自己寫的程式意義。
3. 助教詢問程式意義，依據題目定義與通過測資組數給分 - 每 1 組測試資料(題目範例測資與考試測資)都通過，得該題測資組數分數。
4. 每完成一題，務必舉手評分一次。
5. 使用全域變數不計分。
6. Output 多一個空白或多一空行不影響得分。
7. 完全輸入【Sample Input】後，程式輸出【Sample Output】，檢查得分時，測資請使用檔案，以 cmd 執行程式獲得輸出；x.exe < td.txt。
8. 題目若未註明， $1 \leq N \leq 10$ (N 為輸入個數)
9. 一、二、三題最高 55%，四、五題最多 30%，滿分最高 110%。

一、文章處理 (25%)

輸入一個英文句子 A (空白、大小寫英文字母組合)、2 個英文字(word-以大小寫英文字母組合且以一個空白分隔) P、Q、文章分析處理指令 C，若 C=4，則須輸入整數 N。

輸入說明	輸出說明
Line 1, A	C=1, 在文句 A 中 P 字串前插入 Q 字串。
Line 2, P	
Line 3, Q	C=2, 將文句 A 中 P 字串以 Q 字串取代。
Line 4, C	
Line 5, N	C=3, 將文句 A 中 P 字串刪除。
(當 C=4	C=4, 將文句中，出現頻率小於 N 的英文字刪除。
時，需輸入 N)	C=5, 統計文句 A 中英文字出現頻率前 3 低，頻率低到高排列，若頻率一樣則由字母大到小排列，大寫在小寫前，用：隔開單字與頻率，冒號前後無空白，一個單字與頻率一行。

The monkey want to eat a to make many and many because it is very like
Sample Input 4:
I like to like the like you like to like but if you like the like the likely like I would like the like you like to like like likely 4 3
Sample Output 4:
like to like the like you like to like you like the like the like like the like you like to like
Sample Input 5:
Can you can a can as a canner can can a can Can as Can a as canner Can can ban 5
Sample Output 5:
you:1 canner:2 as:3

Sample Input 1:
Do you do a do as a doer do do a do do bo 1
Sample Output 1:
Do you bo do a bo do as a doer bo do bo do a bo do
Sample Input 2:
However were however is cold or however were however is hot were b 2
Sample Output 2:
However b however is cold or however b however is hot
Sample Input 3:
The monkey want to eat a banana to make many and many banana because it is very like banana banana apple 3
Sample Output 3:

二、指標函示計算圖形的周長與面積及所有圖形的周長與面積加總 (20%)

使用 struct 定義 shape(圖形), circle(圓), rectangle (矩形), square(正方形), triangle(三角形)。

圓有半徑，矩形有長寬，正方形有邊長，三角形有三個邊。計算各圖形周長、面積，及所有圖形周長、面積加總。

注意: PI 設 4。所有長度、計算均以整數處理。

本題須使用以下 struct 定義，以及定義與實做相對的 function pointer 的 function 才計分。

<pre>#define shapeText(TYPE) char name[10]; \ int (*perimeter)(struct TYPE*); \ int (*area)(struct TYPE*); typedef struct shape_s { shapeText(shape_s); } shape_t;</pre>	
<pre>typedef struct circle_s { shapeText(circle_s); int radius; } circle_t;</pre>	<pre>typedef struct rectangle_s { shapeText(rectangle_s); int width, height; } rectangle_t;</pre>
<pre>typedef struct square_s { shapeText(square_s); int side; } square_t;</pre>	<pre>typedef struct triangle_s { shapeText(triangle_s); int s1, s2, s3; } triangle_t;</pre>

輸入說明	輸出說明
Line 1, 輸入圖形個數 N。 Line 2, 輸入圖形種類。 Line 3, 輸入圖形所需資料。 Line 4, 輸入圖形種類。 Line 5, 輸入圖形所需資料。 circle, 下一行輸入半徑。 rectangle, 下一行輸入長寬。 square, 下一行輸入邊長。 triangle, 下一行輸入三個邊。	Line 1~N: 輸出 N 個圖形的種類、周長與面積，以空白間隔。接著以周長大到小排序後輸出，若周長相同，則以面積大到小排序輸出。 Line N+1: 輸出 N 個圖形的周長總和、面積總和，以空白間隔。

Sample Input 1:	Sample Output 1:
3 triangle 3 4 5 square 3 rectangle 4 5	rectangle 18 20 square 12 9 triangle 12 6 42 35
Sample Input 2:	Sample Output 2:
5 triangle 8 23 24 rectangle 25 37 circle 15 square 16 square 17	rectangle 124 925 circle 120 900 square 68 289 square 64 256 triangle 55 78 431 2448

三角形面積公式(海龍公式):

$$s = \frac{s1 + s2 + s3}{2}, \Delta = \sqrt{s(s-s1)(s-s2)(s-s3)}$$

三、LinkList 操作 (25%)

針對空的 List，先進行 N 個資料節點加入，再進行加入、刪除、交換、反轉等操作。輸出操作後 Link List 資料。測試資料不會出現相同整數資料。

本題須使用以下 Link List struct 實作才計分。

```
typedef struct node_s {
    int data;
    struct node_s *next;
} node_t;
```

輸入說明
Line 1, 輸入 N 個整數，依序加入節點。 Line 2, 輸入整數 C，代表操作指令的數量。 Line 3~C+3, 輸入操作指令所須整數資料 x 或 x, y。 操作指令： 1. 從最後面加入資料 x。 2. 刪除最前面節點，若 List 無節點，則不必刪除。 3. 刪除最後面節點，若 List 已無節點，則不必刪除。 4. 刪除 List 內資料為 x 的節點；若 x 不存在則不刪除任何節點。 5. 搜尋 List 內資料為 x 的節點，並在其後加入數值 y 的新節點。若 x 不在 List 中，則不必加入 y。 6. 反轉 Link List。 7. 交換數值為 x 和 y 節點的位置，若 x 或 y 不在 List 中，則不必交換。
輸出說明
從最前面節點開始依序輸出各節點的值，每個數字中間以空白間隔；若 List 為空，則輸出 "None"

Sample Input 1:	Sample Output 1:
1 2 3 1 2	2
Sample Input 2:	Sample Output 2:
4 1 2	None
Sample Input 3:	Sample Output 3:
5 4 3 4 3 3 3 3	None
Sample Input 4:	Sample Output 4:
5 4 3 2 1 3 4 2 4 4 4 6	5 3 1
Sample Input 5:	Sample Output 5:
1 2 3 4 5 3 5 1 9	1 9 2 3 4 5 7

5 5 7 5 6 10	
Sample Input 6:	Sample Output 6:
5 4 3 2 1 3 6 6 6	1 2 3 4 5
Sample Input 7:	Sample Output 7:
8 7 6 5 4 3 2 1 4 7 8 7 7 5 6 7 2 1 7 1 2	7 8 5 6 4 3 2 1

四、LinkList 多項式相加、相減、相乘 (25%)

每筆測資皆會有三次多項式運算，請根據運算符號輸出其結果。

(1) void add(pol_t* X, pol_t* Y, pol_t* Z);

//兩個多項式 X, Y 相加，Z 是結果

(2) void sub(pol_t* X, pol_t* Y, pol_t* Z);

//兩個多項式 X, Y 相減，Z 是結果

(3) void mul(pol_t* X, pol_t* Y, pol_t* Z);

//兩個多項式 X, Y 相乘，Z 是結果

本題須使用以下 struct 與 Link List 實作才計分。

```
typedef struct node_s {
    int coef;
    int exp;
    struct node_s * next;
} node_t;
typedef node_t * nodep_t;
typedef pol_s {
    nodep_t root;
} pol_t
```

輸入說明
Line 1:多項式 X 的 N1 個係數(整數)，第 1 個是 n-1 次方的係數，...，第 n 個代表 0 次方係數。 Line 2:多項式 Y 的 N2 個係數(整數)，第 1 個是 n-1 次方的係數，...，第 n 個代表 0 次方係數。 Line 3: 加、減、乘的運算符號。 例 2 0 3 4 1 -1 3，為 $2x^6 + 3x^4 + 4x^3 + x^2 - x + 3$
輸出說明
Line 1:依照輸入的運算符號，輸出相對應的結果。 (從最高次方到 0 次方的係數與次方)

Sample Input 1:	Sample Output 1:
2 3 0 1 -1 1 0 -1 4 -3 2 + 1 0 -1 4 -3 2 1 1 + 5 4 3 2 1 5 4 3 2 1 +	1 5 2 4 2 3 4 2 -2 1 1 0 1 5 0 4 -1 3 4 2 -2 1 3 0 10 4 8 3 6 2 4 1 2 0
Sample Input 2:	Sample Output 2:

2 3 0 1 -1 1 0 -1 4 -3 2 - 1 0 -1 4 -3 2 1 1 - 5 4 3 2 1 5 4 3 2 1 -	-1 5 2 4 4 3 -4 2 4 1 -3 0 1 5 0 4 -1 3 4 2 -4 1 1 0 0 4 0 3 0 2 0 1 0 0
Sample Input 3:	Sample Output 3:
2 3 0 1 -1 1 0 -1 4 -3 2 * 1 0 -1 4 -3 2 1 1 * 5 4 3 2 1 5 4 3 2 1 *	2 9 3 8 -2 7 6 6 5 5 -6 4 1 1 3 -7 2 5 1 -2 0 1 6 1 5 -1 4 3 3 1 2 -1 1 2 0 25 8 40 7 46 6 44 5 35 4 20 3 10 2 4 1 1 0
Sample Input 4:	Sample Output 4:
1 2 3 4 5 -1 -2 -3 -4 -5 + 1 2 3 4 5 -1 -2 -3 -4 -5 - 1 2 3 4 5 -1 -2 -3 -4 -5 *	0 4 0 3 0 2 0 1 0 0 2 4 4 3 6 2 8 1 10 0 -1 8 -4 7 -10 6 -20 5 -35 4 -44 3 -46 2 -40 1 -25 0
Sample Input 5:	Sample Output 5:
9 -8 3 -2 11 3 -4 2 + 9 -8 3 -2 11 3 -4 2 - 9 -8 3 -2 11 3 -4 2 *	20 3 -5 2 -1 1 0 0 -2 3 -11 2 7 1 -4 0 99 6 -61 5 -27 4 37 3 -34 2 14 1 -4 0

五、LinkList 建構唯一二元樹 (20%)

給定前序或後序以及中序，建構唯一的二元樹，節點資料為大寫英文字母。輸出樹的節點資料內容，順序為由上而下，由左而右印出。

前序代號：P

中序代號：I

後序代號：O

本題須使用以下 struct 與 Link List 實作才計分。

```
typedef struct node_s {
    char data;
    struct node_s * right, * left;
} tree_t;
typedef tree_t * btree;
```

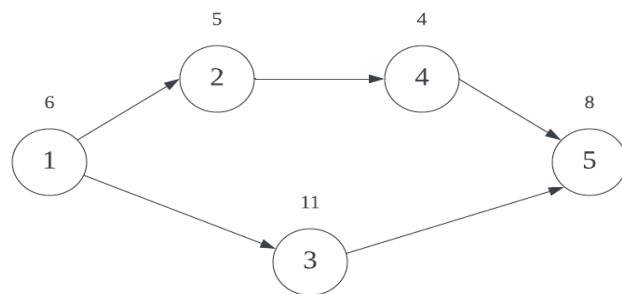
輸入說明	輸出說明
Line 1, n 表示二元樹有 n 個節點	輸出唯一二元樹的內容，由上而下，由左而右。
Line 2, 前、中、後序代號。	
Line 3, 上筆尋訪資料。	
Line 4, 前、中、後序代號。	
Line 5, 上筆尋訪資料。	

Sample Input 1:	Sample Output 1:
9 P ABCDEFGHI I BCAEDGHFI	ABDCEFGIH
Sample Input 2:	Sample Output 2:
7 P ABCDEFG I GFEDCBA	ABCDEFG
Sample Input 3:	Sample Output 3:
9 O IHGFEDCBA I ACEGIHFDB	ABCDEFGHI
Sample Input 4:	Sample Output 4:
15 P ABDHIEJKCFLMGNO I HDIBJEKALFMCNGO	ABCDEFGHIJKLMNO

六、LinkList 開發專案 (25%)

開發專案時，專案被分割為許多項目，分配給多組程式設計師開發。這些項目有順序關係，且只有當順序在前的項目完成，才能開始開發順序在後的項目。

本題使用一個有向無環圖，表示這些項目的開發順序，每個節點代表一個項目，節點內的數字為節點編號，上方所列數字代表開發這個項目所需天數；邊表示開發順序。以下圖為例，節點 2 完成後，才能開始節點 4 的開發。節點 3 與節點 4 都完成後，才能開始節點 5 的開發。節點 1 需 6 小時，節點 2 需 5 小時，節點 3 需 11 小時，節點 4 需 4 小時，節點 5 需 8 小時。完成專案最少需 25 小時，兩條開發專案的路徑為 1, 2, 4, 5 及 1, 3, 5。而花費時間最長的路徑為 1, 3, 5。



本題須使用以下 struct 與 Link List 實作 task 才計分

```
typedef struct node_s {
    int id;
    int time;
    struct node_s * nexts[30];
    int nextCount;
    .....
} task_t;
typedef task_t * pTask;
```

輸入說明
Line 1: 正整數 M，代表輸出的類型。 M=1 代表輸出專案花費的最少時間。 M=2 代表輸出專案開發時花費時間最長的路徑。 M=3 代表輸出專案開發時的兩條路徑。 Line 2: 正整數 N，代表專案共有 N 個 task。 Line 3~N+1: 從第 1 個 task 開始依序往後： 每一行輸入正整數 T, K, t1~tk； T 是完成此 task 時間。 K 表示此 task 完成，後面 K 個 task 才能開始做。 t1~tk 表示等待此 task 完成的 task 編號。
輸出說明
依照正整數 M，輸出對應的結果。

Sample Input 1:	Sample Output 1:
1 5 6 2 2 3 5 1 4 11 1 5 4 1 5 8 0	25
Sample Input 2:	Sample Output 2:
1 8 2 2 2 3 3 1 4 6 1 5 11 2 6 7 9 1 7 10 0 7 1 8 11 0	35
Sample Input 3:	Sample Output 3:
2 5 6 2 2 3 5 1 4 11 1 5 4 1 5 8 0	1 3 5

Sample Input 4:	Sample Output 4:
2 8 2 2 2 3 3 1 4 6 1 5 11 2 6 7 9 1 7 10 0 7 1 8 11 0	1 3 5 7 8
Sample Input 5:	Sample Output 5:
3 10 1 2 2 3 10 1 5 15 1 4 20 1 7 5 1 6 16 1 9 14 1 8 27 1 10 9 1 10 12 0	1 2 5 6 9 10 1 3 4 7 8 10

七、工作排程 (25%)

有 M 個工作要在 N 台機器上加工，每個工作 i 包含若干個工序 O_{ij} ，這些工序須依序加工，也就是前一道工序 $O_{i(j-1)}$ 完成後才可開始下一道工序 O_{ij} 。

每道工序 O_{ij} 可用一個有序對 (K_{ij}, T_{ij}) 表示它需在機器 K_{ij} 上面花費 T_{ij} 小時完成，而每台機器一次只能處理一道工序。

所謂一道工序 O_{ij} 的「最早完成時間的 C_{ij} 」是指考慮目前排程中機器 K_{ij} 之可用性以及前一道工序 $O_{i(j-1)}$ (若該工序存在) 之完成時間後可得的最早完成時間。

所有工序的排程規則如下：

針對每一個工作的第一個尚未排程的工序，計算出此工序的「最早完成時間」，然後挑選出最早完成時間最小的工序納入排程中，如果有多個完成時間都是最小，則挑選其中最小工作編號之工序，一個工序一旦納入排程就不會再更改，重複以上步驟直到所有工序皆納入排程。

輸入說明
Line 1: 正整數 S ，代表輸出的類型。 輸入 1 代表輸出每台機器處理的工序屬於哪個工作。 輸入 2 代表輸出每個工作完成時間的總和。
Line 2: 正整數 N 與 M ， N 代表有多少台機器， M 代表有多少個工作。
接下來會有 M 個工作資訊，輸入順序即為工作編號順序，每個工作資訊有兩行，第一行為整數 P ，代表到工序數量。
Line 3: 有 $2 * P$ 個整數，依序每兩個一組代表一道工序的機器編號與需求時間，每個整數間以空白符號相隔開。
輸出說明
依照正整數 S ，輸出對應的結果。

2 2 4 1 1 3 0 2 2 2 0 1 1 0 7	1 2
Sample Input 2:	Sample Output 2:
1 2 3 1 0 4 1 1 5 1 1 3	1 3 2
Sample Input 3:	Sample Output 3:
1 4 4 3 0 4 1 8 3 5 4 0 2 0 3 2 4 3 1 1 0 9 2 2 3 3 3	2 2 1 3 1 4 2 4 2 1
Sample Input 4:	Sample Output 4:
2 5 5 2 0 1 1 2 2 2 3 3 4 2 0 3 4 5 2 0 1 1 1 2 1 1 2 2	29
Sample Input 5:	Sample Output 5:
2 1 5 3 0 2 0 4 0 6 10 0 2 0 5 0 7 0 9 0 10 0 9 0 7 0 5 0 12 0 7 1 0 30 2 0 14 0 36 5 0 17 0 23 0 7 0 3 0 5	662

Sample Input 1:	Sample Output 1:
1 3 3	2 2 3 1