



職種一程式設計

Problem 1 (堆數分配 13%)

桌上有 1000 元、500 元、200 元、100 元的鈔票若干張，現在想要將這些鈔票平均分成若干堆，分堆的原則：每一堆所包含 1000 元、500 元、200 元及 100 元的張數要相同，且堆數必須為最多，請寫一個程式完成分堆工作。(輸出結果的堆數若非最多，則本題算錯，零分計算。)

輸入說明：輸入之第一列到第四列分別為 1000、500、200、100 元的張數，張數的值介於 1 至 100。

輸入範例：輸入 1000 元的鈔票 10 張，500 元的鈔票 5 張，200 元的鈔票 20 張，100 元的鈔票 30 張。實際輸入之檔案內容如下：

實際輸入：(test1.txt)

```
10
5
20
30
```

輸出說明：第一列為堆數，第二列起為各堆中 1000 元、500 元、200 元、100 元的張數。

輸出範例：上例中的鈔票可分成 5 堆。每堆中包含有：1000 元的鈔票 2 張，500 元的鈔票 1 張，200 元的鈔票 4 張，100 元的鈔票 6 張。實際輸出之檔案內容如下：

實際輸出：(result1.txt)

```
5
1000 2
500 1
200 4
100 6
```

Problem 2 (支票兌換 10%)

目前台幣的鈔票有 2000 元、1000 元、500 元、200 元及 100 元等五種，今周小姐拿了一張支票來銀行兌現，你是銀行櫃台人員，請設計一個計算兌換張數的程式，兌換的原則：兌換鈔票的張數必須為最少。(輸出結果的張數若非最少，則本題算錯，零分計算。)

輸入說明：輸入兌換的金額(金額介於 100 元至 15000 元)。

輸入範例：假定周小姐拿的支票面額為 13500 元。實際輸入之檔案內容如下：

實際輸入：(test2.txt)

```
13500
```

輸出說明：輸出結果為最少的總兌換張數，而 2000 元、1000 元、500 元、200 元及 100 元個別兌換張數，必須列出且順序不得改變。

輸出範例：上例中周小姐的支票面額為 13500 元，則兌換的最少張數為 8 張。實際輸出之檔案內容如下：

實際輸出：(result2.txt)

```
8
2000 6
1000 1
500 1
200 0
100 0
```

Problem 3 (分數相加 14%)

請寫一個程式，可計算多個分數相加的結果，若結果為假分數，必須將其轉成帶分數，且須將它化簡。例如：

$$\frac{5}{6} + \frac{2}{3} + \frac{3}{4} + \frac{1}{12} = 2\frac{1}{3}$$

輸入說明：輸入的第一列為分數的個數 N。(1<n<10)

輸入的第二列到第 N+1 列分別為每一個分數之分子及分母值。

輸入範例：假定輸入 4 個分數，分別為 $\frac{5}{6}$ ， $\frac{2}{3}$ ， $\frac{3}{4}$ ， $\frac{1}{12}$ 。實際輸入之檔案內容如下：

實際輸入：(test3.txt)

```
4
5 6
2 3
3 4
1 12
```

輸出說明：輸出分數相加的結果，共有 3 個值分別為整數值、分子及分母。

輸出範例：上例中 $\frac{5}{6} + \frac{2}{3} + \frac{3}{4} + \frac{1}{12}$ 的值為 $2\frac{1}{3}$ ，故其實際輸出之檔案內容為 2、1、3。

實際輸出：2 1 3 (result3.txt)

請注意背面尚有試題共 2 張 4 頁



Problem 4 (字串數字算式及總和, 12 分)

輸入一組字串(Strings)，每個字串以空白區隔。每個字串中最多僅包含一個數字，且這個數字最小為 1、最大為 999。如果字串中無任何數字，則令此字串包含的數字為 0。請寫一個程式，輸出數字的加法算式及總和。

輸入說明：一組不包含換行符號的字串。每個字串的长度介於 1 到 10 之間，字串個數最多 10 個。

輸入範例：假定有一組字串，共包含 5 個字串，分別為：asdf j213kas kfjas 93wk s82kwv。實際輸入之檔案內容如下：

實際輸入：(test4.txt)

asdf j213kas kfjas 93wk s82kwv

輸出說明：輸出數字的算式及總和。

輸出範例：上例中的一組字串為：asdf j213kas kfjas 93wk s82kwv，共包含 5 個字串，字串所包含的數字分別為：0、213、0、93、82，其數字的算式及總和為：0+213+0+93+82=388。實際輸出之檔案內容如下：

輸出格式：(result4.txt)

213 + 93 + 82 = 388

Problem 5 (最小離差, 18 分)

請將 n 個不同重量的蘋果分成甲、乙兩堆。分堆的原則是：每一個蘋果重量與其所屬堆的平均重量之差距的總和必須為最小。設計一程式，可輸出甲、乙兩堆各個蘋果的重量，及每一蘋果與該堆平均重量的差距總和。(輸出結果的重量差距若非為最小，則本題算錯，零分計算。)

輸入說明：第一列為蘋果個數 n (n 介於 2~50)。

第二列至第 $n+1$ 列為各個蘋果的重量。(蘋果重量介於 1~50)

輸入範例：假定有 5 個蘋果，其重量分別為：2、8、4、3、10。則實際輸入之檔案內容如下：

實際輸入：(test5.txt)

5
2
8
4
3
10

輸出說明：第一列為甲堆各個蘋果的重量。

第二列為乙堆各個蘋果的重量。

第三列為差距總和。

輸出範例：上例中 5 個蘋果重量分別為 2、8、4、3、10(單位)，可分成甲堆為 2、3、4，乙堆為 8、10。甲堆平均重量為 3，乙堆的平均重量為 9，每一蘋果與該堆平均重量的差距總和為 $|2-3| + |3-3| + |4-3| + |8-9| + |10-9| = 4$ 。故實際輸出之檔案內容如下：

實際輸出：(result5.txt)

2 3 4
8 10
4

Problem 6 (迴歸方程式, 16 分)

已知廣告費(X)與銷售量(Y)有因果關係，利用廣告費(X)可預測銷售量(Y)。預測時可使用迴歸方程式(Regression Function)：

$Y = a + bX$ 。要建立迴歸方程式需利用 n 筆的(X, Y)求得 a 及 b 兩係數。 a 及 b 的公式如下，

$$b = \frac{n \sum_{i=1}^n X_i Y_i - (\sum_{i=1}^n X_i)(\sum_{i=1}^n Y_i)}{n \sum_{i=1}^n X_i^2 - (\sum_{i=1}^n X_i)^2}$$

$$a = \bar{Y} - b \bar{X}$$

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i, \quad \bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$$

請寫一個程式輸入 n 筆的(X, Y)，求得 a 及 b ，並輸出其迴歸方程式。

例如：現有 1 月至 6 月的廣告費及銷售量的資料如下：



職種—程式設計

n	X_i (廣告費)	Y_i (銷售量)
1	10	30
2	12	35
3	14	40
4	17	45
5	20	56
6	24	70
Σ	97	276

輸入說明：第一列為資料的個數 n ，(n 介於 5~10)。

第二列至第 $n+1$ 列為廣告費及銷售量的資料。(數值最多為兩位數)

輸入範例：假定有 6 組廣告費及銷售量的資料，分別為：(10,30)、(12,35)、(14,40)、(17,45)、(20,56)、(24,70)。實際輸入之檔案內容如下：

實際輸入：(test6.txt)

```
6
10 30
12 35
14 40
17 45
20 56
24 70
```

輸出說明：輸出迴歸方程式： $Y = a + bX$ ，(a 、 b 取至小數第三位，即第四位四捨五入)。

輸出範例：上例中 6 組廣告費及銷售量的迴歸方程式為： $Y = 0.749 + 2.799X$ 。

實際輸出：(result6.txt)

$Y = 0.749 + 2.799X$



職種—程式設計

Problem 7 (後序運算式, 17 分)

有一種四則運算式稱為後序運算式, 它的運算子置於運算元的後面, 例如: 532^{*+} 為 $5+3*2$ 的後序運算式; 而 $53+2^{*}$ 為 $(5+3)*2$ 的後序運算式。後序運算式的求解過程如下圖示。本題所有測試數值皆為一位數, 所有運算子皆需要兩個運算元(即沒有正負號)美依個測試資料長度皆不超過 20 個字元。

本題實際的輸入及輸出範例如下:

實際輸入: (test7.txt)

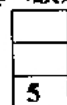
532*+

實際輸出: (result7.txt)

11

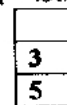
範例 1: 532^{*+}

Step1: 讀入 5, 置入堆疊



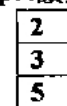
堆疊

Step2: 讀入 3, 置入堆疊



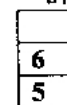
堆疊

Step3: 讀入 2, 置入堆疊



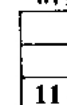
堆疊

Step4: 讀入 *, 由堆疊取出 3 2, 計算結果, 並置入堆疊



堆疊

Step5: 讀入 +, 由堆疊取出 5 6, 計算結果, 並置入堆疊



堆疊

Step6: 輸出 11

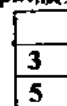
範例 2: $53+2^{*}$

Step1: 讀入 5, 置入堆疊



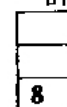
堆疊

Step2: 讀入 3, 置入堆疊



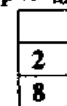
堆疊

Step3: 讀入 +, 由堆疊取出 5 3, 計算結果, 並置入堆疊



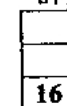
堆疊

Step4: 讀入 2, 置入堆疊



堆疊

Step5: 讀入 *, 由堆疊取出 8 2, 計算結果, 並置入堆疊



堆疊

Step6: 輸出 16