

台北市 107 學年度高級中等學校學生（高工組）電腦軟體設計競賽 決賽試題
工作桌編號 _____ 選手姓名 _____ 代表學校 _____ 總分 _____

試卷說明：1. 請將寫好之程式原始檔依題號命名存檔，第一題取：選手姓名_Q1，第二題取：選手姓名_Q2，依序命名存檔，並存於 C 碟之選手姓名_Contest 目錄。2. 競賽時間 4 小時。（若同分者依完成時間(<4 小時)先後排序，餘者依演算法優劣排序）

試題 1：排列計算和列印

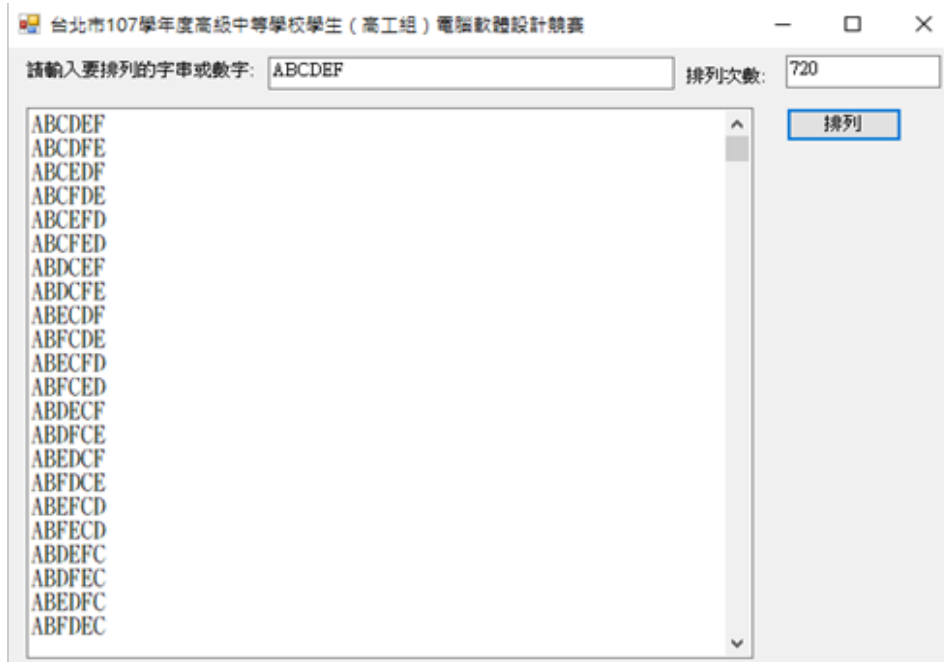
說明：排列是集合中，所有元素的可能組合，例如，給一個集合有三個整數： $\{0, 1, 2\}$ ，此三個整數有 6 種可能的排列： $\{012, 021, 102, 120, 201, 210\}$ 。當給另一個集合有四個字元： $\{t, e, s, t\}$ ，此四個字元有 24 種可能的排列： $\{test, tets, tset, ttes, tste, ttse, etst, etts, stet, ttes, stte, ttse, estt, etts, sett, tets, stte, tste, estt, etst, sett, test, stet, tset\}$ 。

請您寫一程式，能讓使用者輸入要排列的字串或數字，字串長度 $3 \leq N \leq 8$ ，使用者再按排列按鈕，畫面會顯示所有可能的排列，同時輸出排列次數。

評分：

1. 可以讓使用者輸入要排列的字串或數字(2.5 分)。
2. 可以讓使用者按一個排列按鈕，執行排列的程式(2.5 分)。
3. 承上，可以顯示所有可能的排列和排列次數(2.5 分)。
4. 可以顯示 3 個字串或數字所有可能的排列和排列次數(2.5 分)。
5. 可以顯示 4 個字串或數字所有可能的排列和排列次數(2.5 分)。
6. 可以顯示 5 個字串或數字所有可能的排列和排列次數(2.5 分)。
7. 可以顯示 6 個字串或數字所有可能的排列和排列次數(2.5 分)。
8. 可以顯示 7 個字串或數字所有可能的排列和排列次數(2.5 分)。
9. 可以顯示 8 個字串或數字所有可能的排列和排列次數(3.5 分)。
10. 友善的介面(1.5 分)

範例 1：輸入 6 個字元的排列和排列次數



範例 2: 輸入 7 個數字的排列和排列次數

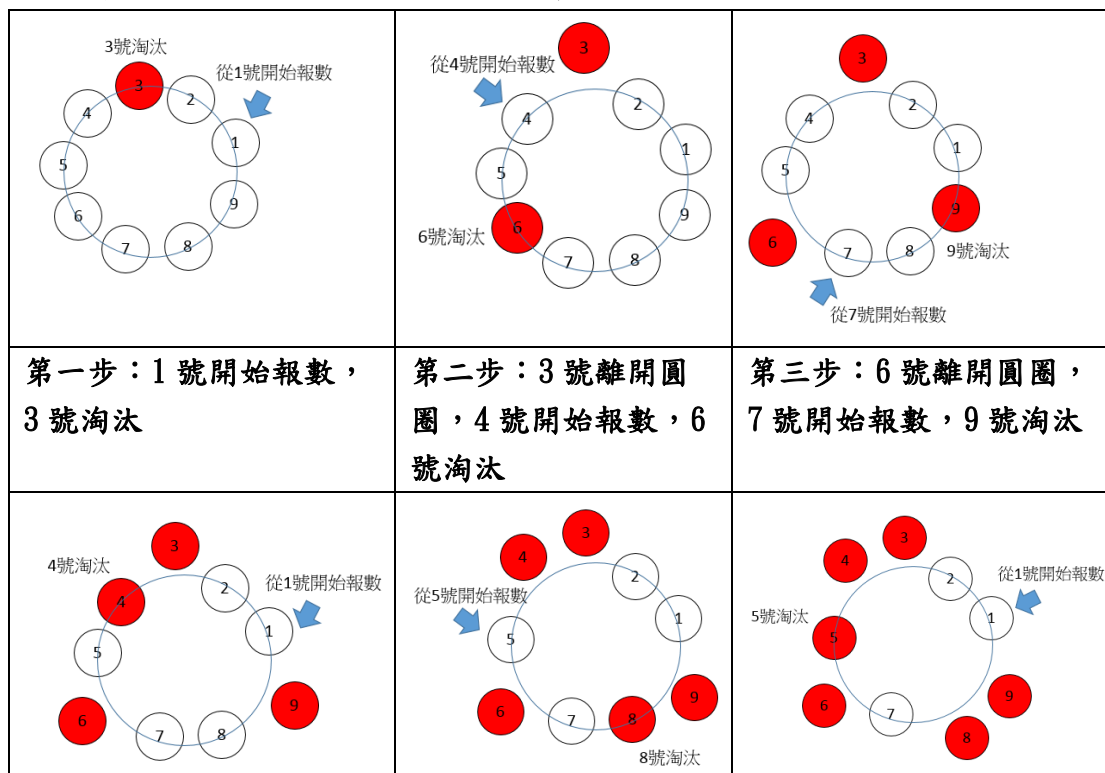
台北市107學年度高級中等學校學生（高工組）電腦軟體設計競賽

請輸入要排列的字串或數字: 排列次數:

1234567
1234576
1234657
1234756
1234675
1234765
1235467
1235476
1236457
1237456
1236475
1237465
1235647
1235746
1236547
1237546
1236745
1237645
1235674
1235764
1236574
1237564

試題 2：大地遊戲

- 問題描述：大家或許都玩過一個大地遊戲，N 個人圍成一個圓圈，從第一個人開始報數，照順序從 1 開始報數，當報到一個特定的數 M，例如 M=3，報數的那個人就被淘汰，淘汰的人就要離開圓圈，不能報數了。然後從下一個人開始從頭（1）報數，報到 3 的就淘汰。例如假設有 N=9 人（編號為 1~9）圍成一個圓圈，從 1 開始報數，報到 3 的就淘汰，如下圖，淘汰的順序是 3、6、9、4、8、5、2、7，最後勝利者是 1 號。



第四步：9 號離開圓圈，1 號開始報數，4 號淘汰	第五步：4 號離開圓圈，5 號開始報數，8 號淘汰	第六步：8 號離開圓圈，1 號開始報數，5 號淘汰
第七步：5 號離開圓圈，7 號開始報數，2 號淘汰	第七步：2 號離開圓圈，7 號開始報數，7 號淘汰	最後贏家：1 號

2. 輸出入說明：介面格式不拘，可以輸入參與大地遊戲的人數 $N(1 < N < 100)$ ，報數 $M(1 < M < 100)$ ，可以輸出淘汰的順序，以及最後贏家。

3. 範例輸入與輸出：

範例輸入	範例輸出
請輸入 N 人數：5 請輸入 M 報數：2	淘汰順序：2 4 1 5 最後贏家：3
請輸入 N 人數：5 請輸入 M 報數：3	淘汰順序：3 1 5 2 最後贏家：4
請輸入 N 人數：9 請輸入 M 報數：3	淘汰順序：3 6 9 4 8 5 2 7 最後贏家：1
請輸入 N 人數：9 請輸入 M 報數：2	淘汰順序：2 4 6 8 1 5 9 7 最後贏家：3

4. 評分說明：以上每個測資佔 15%，另外有額外的測資佔 40%

5. 提示：使用循環陣列，當陣列索引超過陣列大小，索引會從頭開始。例如：假設陣列大小為 10，其索引值應為 0, 1, 2, ..., 到 9，當索引值為 10 時，必須歸 0，從 0 開始。

試題 3: 區塊之方向性梯度直方圖

說明：區塊之方向性梯度直方圖(Histogram of Oriented Gradient)可用來表示一個影像區塊的特徵。試設計一個程式可用來計算某個選定區塊之方向性梯度直方圖。假設其演算法如下所示。

1. 每個區塊大小為 8×8 。xBlock 與 yBlock 分別用來指出水平與垂直方向的第幾個區塊。例如：xBlock=0 且 yBlock=0 表示最左上角的區塊；xBlock=3 且 yBlock=1 表示水平第 3 個且垂直第 1 個區塊(如圖 1 所示)

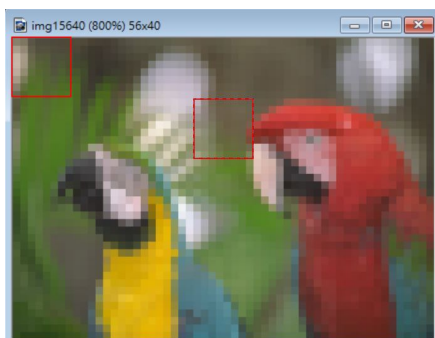


圖 1

2. 每個選定的區塊會有藍色(B)，綠色(G)，和紅色(R)三個平面的資料。利用這些資料，以下列定義計算第 I 個平面(I 為 B , G , 或 R)之水平梯度 G_x 、垂直梯度 G_y 、梯度的振幅(magnitude) M 與角度(angle) A 。

$$G_x(m, n) = I(m+1, n) - I(m-1, n)$$

$$G_y(m, n) = I(m, n+1) - I(m, n-1)$$

$$M = \sqrt{G_x^2 + G_y^2}$$

$$A = \tan^{-1} \frac{G_y}{G_x}$$

注意：當選定的區塊位於影像的邊緣時，則座標落在影像外者，其值為 0。例如：圖 2(a)為最左上角區塊之藍色(B)平面部分資料，圖 2(b)為其相對應 5 個座標點所計算出來的梯度參數。

107	99	70	63	62	60	59	58	59	57
139	118	75	65	63	61	60	56	55	
164	134	77	61	59	61	59	52	52	

(a)

m	n	$I(m, n)$	G_x	G_y	M	A
0	0	107	99	139	170.65	54.54
1	0	99	-37	118	123.66	107.41
2	1	75	-53	7	53.46	172.48
3	1	65	-12	-2	12.17	-170.54
7	1	56	-5	-6	7.81	-129.81

(b)

圖 2 (a)xBlock=0 且 yBlock=0 之 B 平面部分資料 (b)在相對應(m, n)座標所計算出來的水平梯度 G_x 、垂直梯度 G_y 、梯度的振幅 M 與角度 A

3. 比較 B , G , R 三個平面中同一個位置的梯度振幅 M ，取最大值當作是該點之梯度振幅 M_g ，所對應之角度即為該點之梯度方向 A_g 。例如：在(0, 0)位置， B 平面之 $M=170.65$, $A=54.54$; G 平面之 $M=197.61$, $A=52.61$; R 平面之 $M=211.60$, $A=53.46$ ，則 $M_g = 211.60$ 且 $A_g=53.46$ 。將 M_g 與 A_g 分別加上 0.5 後，取小於該結果的整數，以便得到整數版本之梯度振幅 M_b 與梯度方向 A_b (例如上面的例子將產生 $M_b=212$ 與 $A_b=53$)。將整個區塊之 M_b 與 A_b 列印出來。如圖 3 所示，當按下「Calculate」鍵時將列印出 xBlock=0 且 yBlock=0 之整數版本之梯度振

幅 M 與梯度方向 A 。

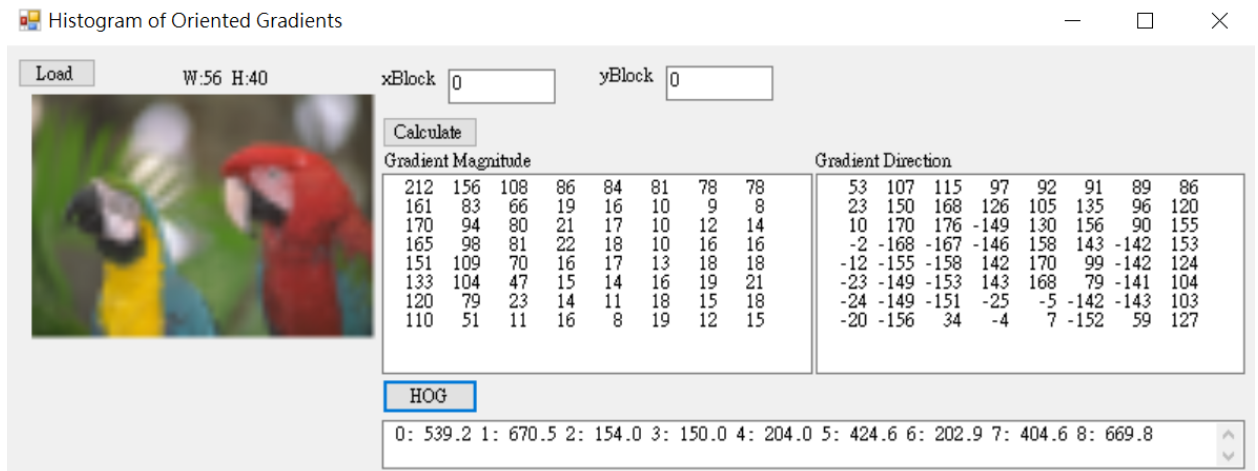


圖 3

4. 梯度方向 A 取絕對值，並且分成 9 個箱子(0~8)，每個箱子的範圍為 20 度，再將梯度振幅 M 依比例分配到這些箱子中。例如：如圖 4 所示，梯度方向 $A=10$ ，其所對應之梯度振幅 $M=170$ ，將分別分至 0 號箱子的值為 85 與 1 號箱子的值為 85；梯度方向 $A=120$ ，其所對應之梯度振幅 $M=8$ ，將直接分至 6 號箱子；而梯度方向 $A=-168$ ，取絕對值後為 168，其所對應之梯度振幅 $M=98$ ，將分配至 8 號箱子的值為 58.8 且分配至 0 號箱子的值為 39.2，以此類推。分配至同一個箱子的振幅將累加起來。如圖 3 所示，當按下「HOG」鍵時，將列印出累加之後的結果，顯示此區塊之方向性梯度直方圖之數值。

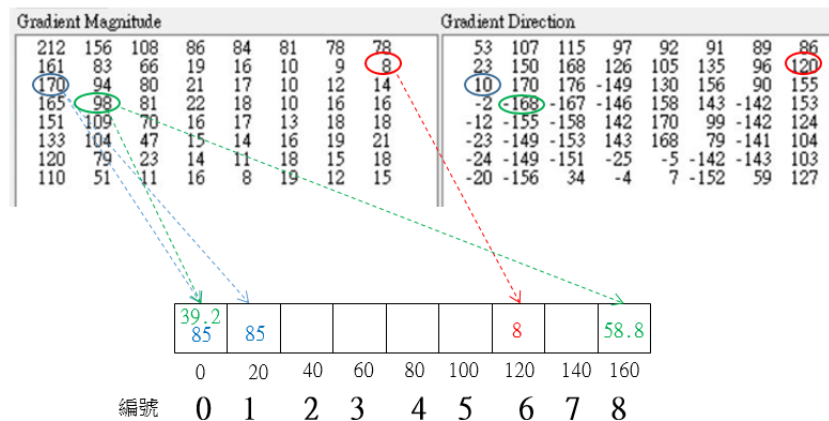


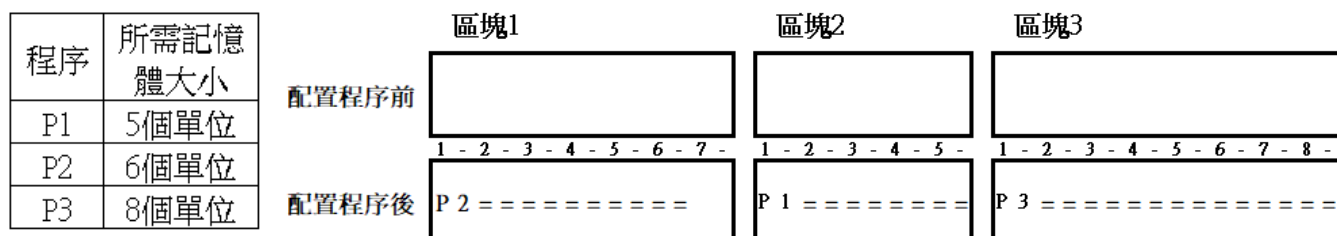
圖 4

評分項目：

1. 可以正確地顯示某個選定區塊之梯度振幅 M 。(7 分)
2. 可以正確地顯示某個選定區塊之梯度方向 A 。(8 分)
3. 可以正確地顯示某個選定區塊之方向性梯度直方圖之數值。(10 分)

試題 4：簡易**最適**配置法之記憶體管理 (Memory Management---Best fit)程式說明：在作業系統中，**最適**配置法之記憶體管理單元能依需求分配夠大而且最接近需求的閒置空間來放置程序 (Process) 程式。

例如：有 3 個程序(如下左圖)所需佔用的記憶體空間分別為 5、6、8 個單位。閒置的記憶體空間如下右圖，有 3 個區塊、大小分別為 7、5、9 個單位，一個記憶體單位大小等同 2 個等號。



程序配置前的記憶體空間是空白的 (如上右圖上半部分)，程序配置後的記憶體空間是顯示程序號碼及數個等號 (如上右圖下半部分)。因採用**最適配置法**，**P1** 程序需要 5 個單位所以被配置到 ≥ 5 單位的區塊 2，並用**==號**填滿整區塊。**P2** 程序需要 6 個單位所以被配置到 ≥ 6 單位且最接近 6 個單位的區塊 1，並用**==號**填入該區塊，只留下 1 單位空間。**P3** 程序需要 8 個單位所以被配置到 ≥ 8 單位且最接近 8 個單位的區塊 3，並用**==號**填入該區塊，只留下 1 單位空間。

目標：

1. 請寫一支程式如右圖所示，區塊數 <6 和程序數 <6 ，顯示相關訊息 (2 分)。
2. 可輸入記憶體區塊數量及程序數量(2 分)。
3. 接著輸入每個記憶體區塊的大小，最後一個區塊輸入後才換行 (4 分)。
4. 接著輸入每個程序所需的區塊大小，最後一個程序輸入後才換行(4 分)。

記憶體管理程式-最適配置法(Best Fit)

請輸入記憶體區塊數(<6): 3

請輸入程序數(<6): 3

請輸入各區塊大小(<10)---

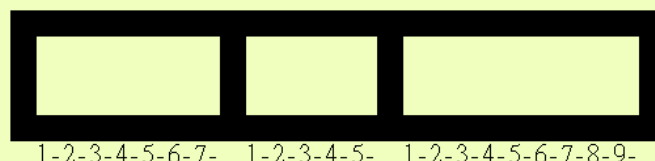
區塊1: 7, 區塊2: 5, 區塊3: 9,

請輸入各程序所需的大小(<10)---

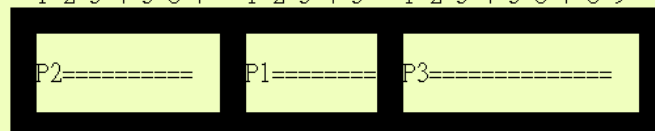
程序1: 5, 程序2: 6, 程序3: 8,

程序編號	所需大小	區塊編號	區塊大小	剩餘空間
1	5	2	5	0
2	6	1	7	1
3	8	3	9	1

配置程序前



程序配置後



請按任意鍵繼續 . . .

5. 接著採**最適配置法**依程序順序將相關資訊顯示出來(13 分)。

注意：不用處理資料輸入錯誤問題。

另一例子如右圖：

因採用**最適配置法**，

P1 程序需要 8 個單位所以被配置到 ≥ 8 單位的區塊 2，並用**==號**填滿整區塊。

P2 程序需要 7 個單位，因沒有 ≥ 7 單位的空間，所以未被配置。

P3 程序需要 6 個單位所以被配置到 ≥ 6 單位的區塊 1，並用**==號**填滿整區塊。

P4 程序需要 6 個單位，因沒有 ≥ 6 單位的空間，所以未被配置。

記憶體管理程式-最適配置法(Best Fit)

請輸入記憶體區塊數(<6): 3

請輸入程序數(<6): 4

請輸入各區塊大小(<10)---

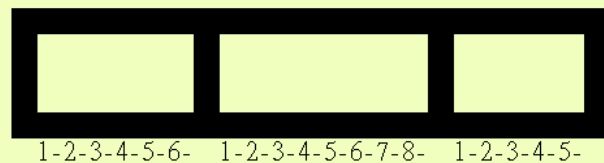
區塊1: 6, 區塊2: 8, 區塊3: 5,

請輸入各程序所需的大小(<10)---

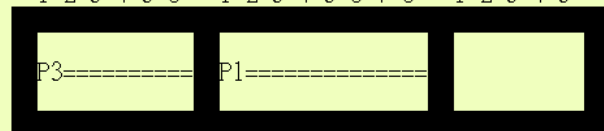
程序1: 8, 程序2: 7, 程序3: 6, 程序4: 6,

程序編號	所需大小	區塊編號	區塊大小	剩餘空間
1	8	2	8	0
2	7	未分配記憶體區塊		
3	6	1	6	0
4	6	未分配記憶體區塊		

配置程序前



程序配置後



請按任意鍵繼續 . . . ■