**臺灣師範大學機電工程系 106 程式設計 Classwork 8**

1. **請將 Classwork 8 的程式碼整合進入Classwork1-7的選單程式中。**
2. **陣列為整數陣列，陣列元素個數由外界輸入，最大為1000。每個元素的值介於0~1000之間。**
3. 將以下所列之 binary search 的程式碼用遞迴 (recursive) 的方法進行改寫。

data : 陣列; n: 陣列的大小; search: 欲尋找的數字

回傳值 : search 在 data 中的位置

|  |
| --- |
| int binarySearch(const int data[], int search, int n)  {  int low = 0, high = n - 1;  int mid;  while (low <= high)  {  mid = (low + high) / 2;  if (data[mid] == search)  return mid;  else if (search < data[mid])  high = mid - 1;  else if (search > data[mid])  low = mid + 1;  }  return -1;  } |

1. 請寫出一個程式，當輸入一個整數n時，它將亂數產生一個具有 n個整數的陣列，並計算其四分位數。

四分位數的演算法如下:

1. 利用中位數使數據分成兩列（不要把中位數放入已分好的數列），
2. 第一四分位數為第一組數列的中位數；第三四分位數為第二組數列的中位數。

|  |  |  |
| --- | --- | --- |
| 範例1 | 範例2 | 範例3 |
| 數據總量:  6, 47, 49, 15, 42, 41, 7, 39, 43, 40, 36  由小到大排列的結果:  6, 7, 15, 36, 39, 40, 41, 42, 43, 47, 49 | 數據總量: 7, 15, 36, 39, 40, 41 | 數據總量: 1, 2, 3, 4 |
| \begin{cases} Q_1  = 15 \\ Q_2  = 40 \\ Q_3 = 43 \end{cases} | \begin{cases} Q_1 = 15 \\ Q_2 = 37.5 \\ Q_3 = 40 \end{cases} | \begin{cases} Q_1  = 1.5 \\ Q_2  = 2.5 \\ Q_3 = 3.5 \end{cases} |
|  |  |  |

3. 給定一個整數陣列(可用亂數產生)，求出現頻率最高的元素，並計算其出現的次數。

|  |
| --- |
| 測試 :   1. 給定以下數列: 3,1,1,2,3,2,2,2,4,5,2,8,10 2. 排序: 1,1,2,2,2,2,2,3,3,4,5,8,10 3. 出現頻率最高的數字為 2, 出現的次數為 5 |
| 函式原型:  void longestPlateau(int nArr[], int size, int ans[]);  其中:  nArr 為傳入的陣列; size 為nArr陣列的元素個數;  ans 用來儲存計算結果, ans[0], ans[1] 分別為出現次數最高元素的值及其次數。 |