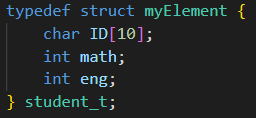
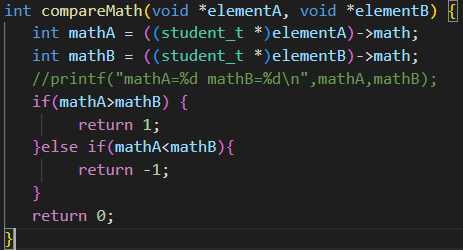
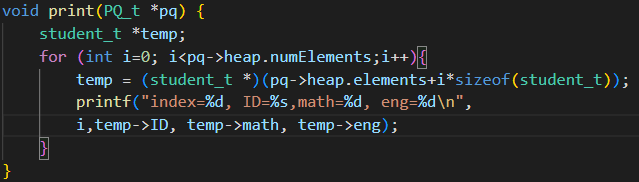
Priority Queue Library使用說明書

* 事前準備 (如何使用Priority Queue Library ? ) ：
  1. 匯入相關函式庫: #include "pq.h"。
  2. 建立自己要存放資料的結構，範例如下圖顯示:



* 1. 根據你的結構，定義清楚比較(compareMath)與打印(print)的函式，範例如下顯示:
     + 比較函式(compareMath): 讓PQ函式庫知道你要拿甚麼資料進行比較。

請務必在element A > element B 時回傳 1，反之為回傳 -1，相等時回傳 0 。

* + - 打印函式(print):此PQ函式庫是以陣列的資料結構去實現，因此請用sizeof(你存放資料的結構) 的方式來移動位址，來取得你每一顆在PQ中的資料，並將其印出。
* 了解Priority Queue (PQ) Library 與 功能使用說明 :

void **createPQ**(PQ\_t \*pq, H\_class pqClass, int elementSize, int maxSize, int (\*compare)(void\* elementA, void \*elementB));

* 函式說明: 初始化一個pq的結構，並設定其PQ的種類(pqClass)、元素大小(elementSize)、最多元素個數(maxSize) 與 比較函數(compare)。
* 輸入要求:
  1. PQ\_t \*pq: 為要**初始化**PQ結構的變數名稱，可由以下語法建立:

PQ\_t maxPQ; //建立一個PQ\_t結構，並將其命名為maxPQ

* 1. H\_class pqClass: 設定其**PQ的種類**，輸入MINHEAP or MAXHEAP來設定此PQ資料結構的**存放方式**。
  2. elementSize: 告訴lib你**每個元素(你要存放資料的結構)的大小**為多少，可以透過sizeof()來取得大小。
  3. maxSize: 輸入一個整數告訴lib你最多需要存放幾筆元素資料。
  4. compare: 利用函數指標告訴lib要**如何**將元素新增到PQ中，也就是使用事前準備好的compareMath函式，讓lib知道要以哪一個數值(此範例使用Math值)進行比較與新增元素。
* 使用範例:

createPQ(&maxPQ, MINHEAP, sizeof(student\_t), 100, compareMath);

int Enqueue(PQ\_t \*pq, void \* elementA);

* 函式說明: 將元素(elementA)加入到已經初始化過後的PQ結構中。
* 輸入要求與使用範例:

1. PQ\_t \*pq: 給定已經**初始化**過後的PQ結構的**位置**
2. elementA: 要加入到PQ結構中的元素**位置**，多筆資料的建立與新增可參考以下語法:

* 建立3比學生資料:

student\_t node[3]={

{"C120308001", 70, 100},

{"B220406001", 60, 90},

{"D120306001", 80, 95},

};

* 利用for迴圈將上述資料新增到PQ結構中:

for(int i=0;i<6;i++){

        Enqueue(&maxPQ, &node[i]);

        }

void \* Dequeue(PQ\_t \*pq);

* 函式說明: 根據pqClass的規則將元素(elementA)從PQ結構取出。
* 輸入要求:

1. PQ\_t \*pq: 給定想要取出資料元素的PQ結構的位置。

* 使用範例:

for(int i=0;i<3;i++){

    student\_t \*temp = (student\_t \*)Dequeue(&maxPQ);

  printf("Dequeue: ID=%s, math=%d,eng=%d\n",temp->ID,

temp->math,temp->eng);

    free(temp);

    }

* 一張含有 文字 的圖片

  自動產生的描述執行成果(以MINHEAP為例):

int IsEmpty(PQ\_t \*pq);// return 0: not empty, 1: empty

int IsFull(PQ\_t \*pq); // return 0: not full, 1:full

* 函式說明: 判斷PQ結構是否為空或填滿。
* 輸入要求:

1. PQ\_t \*pq: 給定想要查看PQ結構的位置

根據上述說明完成main.c的程式碼後，可以透過以下語法使用libpq.a製作

執行檔 (main\_static.exe):

gcc main.c libBST.a -o main\_static

其中: main.c 是你的程式碼、是libpq.a 是PQ的函式庫、main\_static是輸出的執行檔名稱。