

Project 4 – Car Queueing Problem

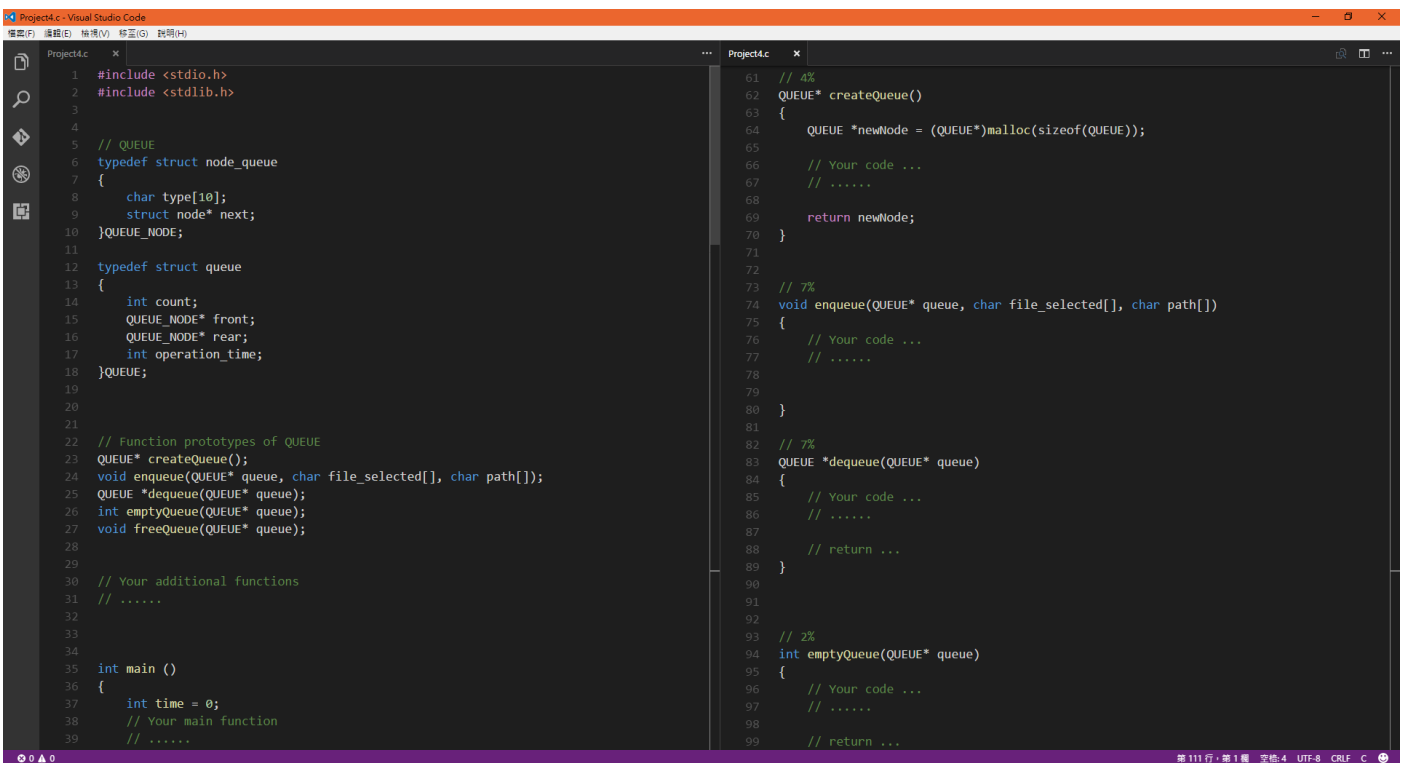
收費站是電子收費站出現前，以人工收取回數票或現金，藉以收取交通規費。實作收費站排程系統，以茲紀念科技的日新月異。

※問題描述：

使用 Queue 實作收費站排程系統，從檔案中讀取 N 個車道，車道分別有小型車專用道 (car_lane) 與大型車專用道 (truck_lane)，每個車道之收費站有不一樣的處理時間。並讀取車輛腳本產生車輛，車輛腳本為幾單位時間時，出現何種車輛。小型車輛任何車道都可以走、大型車只能夠走大車道。

※功能需求 – Basic functions：

完成 Queue functions (24%)，評分如程式碼中的評分。



```
Project4.c - Visual Studio Code
檔案(F) 編輯(E) 檢視(V) 格式(O) 說明(H)

Project4.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 // QUEUE
6 typedef struct node_queue
7 {
8     char type[10];
9     struct node* next;
10 }QUEUE_NODE;
11
12 typedef struct queue
13 {
14     int count;
15     QUEUE_NODE* front;
16     QUEUE_NODE* rear;
17     int operation_time;
18 }QUEUE;
19
20
21
22 // Function prototypes of QUEUE
23 QUEUE* createQueue();
24 void enqueue(QUEUE* queue, char file_selected[], char path[]);
25 QUEUE *dequeue(QUEUE* queue);
26 int emptyQueue(QUEUE* queue);
27 void freeQueue(QUEUE* queue);
28
29
30 // Your additional functions
31 // .....
32
33
34
35 int main ()
36 {
37     int time = 0;
38     // Your main function
39     // .....
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61 // 4%
62 QUEUE* createQueue()
63 {
64     QUEUE *newNode = (QUEUE*)malloc(sizeof(QUEUE));
65
66     // Your code ...
67     // .....
68
69     return newNode;
70 }
71
72
73 // 7%
74 void enqueue(QUEUE* queue, char file_selected[], char path[])
75 {
76     // Your code ...
77     // .....
78
79
80 }
81
82 // 7%
83 QUEUE *dequeue(QUEUE* queue)
84 {
85     // Your code ...
86     // .....
87
88     // return ...
89 }
90
91
92
93 // 2%
94 int emptyQueue(QUEUE* queue)
95 {
96     // Your code ...
97     // .....
98
99     // return ...
100 }
```

一個車道收費站 (6%)，讀取車道檔案 lane.txt 中的第一個車道，與所有車輛檔案 cars.txt，輸出一個 output.txt，內容為：什麼時間、哪個車道通過了什麼類型的車輛。

```
lane - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
lane_1 truck_lane 5
lane_2 car_lane 5
lane_3 car_lane 10
lane_4 car_lane 6
lane_5 truck_lane 6
lane_6 car_lane 6
lane_7 truck_lane 60
lane_8 car_lane 6
```

第一個欄位為第幾個車道、第二個欄位為什麼種類的車道、第三個欄位為處理時間。

```
cars - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
0 car
0 truck
0 car
0 car
0 car
1 car
1 car
2 car
2 car
3 car
4 car
5 car
6 car
7 car
8 truck
8 car
9 car
10 car
```

第一個欄位為系統單位時間、第二個欄位為什麼車種。

```
output - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
5 lane_1 car
10 lane_1 truck
15 lane_1 car
20 lane_1 car
25 lane_1 car
30 lane_1 car
35 lane_1 car
40 lane_1 car
45 lane_1 car
```

第一個欄位為系統單位時間、第二個欄位為第幾個車道、第三個欄位為何種車種。

※功能需求 – Main functions：

多個車道收費站 (5%)，處理多個車道，如果所有車輛都只通過其中一個車道，不算分。

多個車道的前提下，駕駛不會跑錯車道 (5%)，大型車無法通過小型車車道。

最佳化排程 (10%)，全班所有有做出來的組別統一評分，總等待時間最短的為 10 分，最慢者為 6 分。

※功能需求 – Extra Bonus：

使用 `system("cls")`，來呈現動畫。或是呼叫其他 library 繪圖，也可以使用遊戲引擎製作。

※配分 (110%) :

Basic functions (30%) :

完成 Queue functions (24%)

一個車道收費站 (6%)

Main functions (20%) :

多個車道收費站 (5%)

多個車道的前提下，駕駛不會跑錯車道 (5%)

最佳化排程 (10%)

Extra bonus (10%) :

動畫呈現收費站、車道與車子。

Documents + Interview (50%) :

Design Document (flow chart, functions)

User manual (how to use your project)

Time log & team work (every team member)

Interview (every team member)

※說明：

1. 需要計算單位時間，隨程式運作累積計算花費時間。
2. Main functions 為 20 分，Document + Interview 為 50 分。
3. 範例程是僅供參考，可自由發揮，但必須完成 Queue 的功能。

※條件限制：

1. 所有車輛、車道必須讀檔，結果必須產生一個 output.txt。
2. 必須使用 C 語言實作，Bonus 除外。
3. 禁止使用「全域變數」。

※重要：

1. Deadline：2016/12/22(四) 上午 10 點，以 e-course 最後上傳的版本為主，不接受遲交。
2. Compile 不過，Project 0 分，上傳之前自己要對自己的 Project 負責，整合後務必自行測試（做不出來的功能就先 Comment 起來，以免整個 Project 沒有分數）。
3. Interview deadline：2016/12/29(四) 下午 3 點前。
4. Interview 時間最晚必須提早前一天與助教約，助教答覆後才算約成。遲到者該次 Interview 5 折，遲到超過 30 分鐘或爽約該次 Interview 0 分。
5. 需符合以上條件限制，功能才算達成。
6. Project 如果有問題，可以詢問助教，請求協助。



補充

中華民國收費站列表：

<https://zh.wikipedia.org/wiki/%E4%B8%AD%E8%8F%AF%E6%B0%91%E5%9C%8B%E5%9C%8B%E9%81%93%E6%94%B6%E8%B2%BB%E7%AB%99%E5%88%97%E8%A1%A8>

Monogame 遊戲引擎，使用 C#：

<http://www.monogame.net/>

Unity 3D 遊戲引擎，使用 C#：

<https://unity3d.com/>

有興趣的同學可以先學習物件導向程式設計方式，對於未來寫程式會有很大的幫助。