函數使用說明書（pdf 檔）

```
dllNode_t * DLL_init();
```

此函式會產生一個 node(head)，並將此 node 的 next 與 prev 初始化(設為 NULL)後 return。 請注意此 head node 是不能被刪除的，且不當資料點。

```
int DLL_isEmpty(const dllNode_t *head);
```

此函式接收一個鏈結串列的起始位置(head node)，當為空時回傳 1，反之回傳 0。

```
dllNode_t * DLL_next_node(const dllNode_t * node);
```

此函式會回傳輸入之 node 的下一個節點。

```
dllNode_t * DLL_prev_node(const dllNode_t * node);
```

此函式會回傳輸入之 node 的上一個節點。

```
unsigned int DLL_num_nodes(const dllNode_t *head);
```

此函式會計算一個鏈結串列的起始位置(head node)中有幾個節點。

```
void DLL_add_first(dllNode_t * new_node, dllNode_t * head);
```

此函式會將新節點 new_node 加入到一個鏈結串列的起始位置(head node)的第一個位置。

```
void DLL_add_tail(dllNode_t * new_node, dllNode_t *head);
```

此函式會將新 new_node 加入到一個鏈結串列的起始位置(head node)的最後一個位置。

```
void DLL_addto_prev(dllNode_t *new_node, dllNode_t *node);
```

此函式會將新 new_node 加入到 node 的前一個位置。

```
void DLL_addto_next(dllNode_t *new_node, dllNode_t *node);
```

此函式會將新 new_node 加入到 node 的後一個位置。

```
void DLL_delete(dllNode_t * node);
```

此函式會從 node 所在的 Linked List 中刪除此節點位置，請注意使用者需要自行把該節點使用的記憶體空間釋放。

```
dllNode_t * DLL_concate(dllNode_t *srcList, dllNode_t * dstList);
```

此函式會將一個鏈結串列的起始位置(srcList) 串在另一個鏈結串列的起始位置

(dstList)之後。