**Data Structure\_2071035 Lee Somin**

**Technical Report – linked\_stack.cpp**

*Theorical Explanation of Functions in ‘linked\_stack.cpp’*

**typedef element**

This typedef represents the integer element of the list’s node.

**typedef struct DlistNode**

This structure contains the actual information of node, the data(int item) and the left and right link(DlistNode\* llink, DlistNode\* rlink).

**typedef struct LinkedStackType**

This structure contains the pointer to the head node of the circular doubly linked stack.

**init**

Input: DlistNode\* s

Return: non

This function initializes the circular doubly linked stack by setting the headnode of the stack and initializing its two link to the head node ‘s’.

**is\_empty**

Input: DlistNode\* s

Return: s->rlink==s //0 or 1

This function is for determining if the stack is empty. The stack is empty when the right link of the head node points to itself. The function returns 1 when the queue is empty and 0 when the queue is not empty.

**push**

Input: DlistNode\* s, element item

Return: non

This function is to add a new element to the top of the stack. It allocates memory or new node, ‘Dlistnode \*temp’. If the memory is allocated without error, the function checks if the stack is empty using function ‘is\_empty()’ and if it is, it sets the right link of the head node to the new node making it the first node. Then, it sets the new node’s item as the given ‘element item’ as input, sets the left link of the new node to headnode’s left link(the last node of the stack), and the right link to the headnode. After, the last node(previously)’s right link and the headnode’s left link is set to the new node. This sequence makes the new node added to the tail of the circular linked list.

**pop**

Input: DlistNode\* s

Return: element item

This function is to pop the last element added from the stack. The function checks if the stack is empty using ‘is\_empty()’, and prints error message if it is. If the stack is not empty, temp node is set to the last node of the stack(the left link of the headnode, s->llink). ‘int item’ is assigned with the item of temp for the return. Then, left link of the headnode is set to the node before the last(removed) node(s->llink->llink), and that node’s right link is updated to the headnode and the node is deallocated. When all is over, the function returns ‘int item’ which is the element of the removed last node.

**peek**

Input: DlistNode\* s

Return: element s->llink->item

This function is to return the last element added from the stack without removing any node. The function checks if the stack is empty using ‘is\_empty()’, and prints error message if it is. If the stack is not empty, it returns ‘element s->llink->item’.

**main()**

input: non

Return: non

In main, it sets the circular doubly linked stack s and initializes by calling the function 'init()'. Then, it pushes 1,2,3 in order. Finally, it pops the element of stack from top.

**<Execution Result>**

텍스트이(가) 표시된 사진

자동 생성된 설명