

TI DSP, MCU, Xilinx Zynq FPGA 기반의 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 김형주

mihaelkel@naver.com

What I learned (18.03.05)

- Data Structure : queue

Data Structure : Queue

- Stack : FIFO structure
- Methode : enqueue, deque

Data Structure : Queue

- Consist of main(), get_node(), enqueue(), deque()

```
howard@ubuntu: ~/HomeworkBackup/8th
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct __queue{
5     int data;
6     struct __queue *link;
7 }Queue;
8
9
10 Queue* enqueue(Queue **head, int data);
11 Queue* get_node(void);
12 Queue* deque(Queue **head, int data);
13 void printfqueue(Queue **head);
14
15 int main(void){
16     Queue* head = NULL;
17     enqueue(&head,10);
18     enqueue(&head,20);
19     enqueue(&head,30);
20     enqueue(&head,40);
21
22     deque(&head,10);
23
24     printfqueue(&head);
25
26     return 0;
27 }
```

Data Structure : Queue

- Queue* get_node()
- This function returns Queue type pointer initialized link = NULL

```
Queue* get_node(){  
    Queue* tmp;  
    tmp = (Queue*)malloc(sizeof(Queue)*1);  
    tmp->link = NULL;  
  
    return tmp;  
}
```

Data Structure : Queue

- Queue* enqueue(Queue **head, int data)
- This function accumulates data on memory. It finds the next memory space be used through recursive function.

```
Queue* enqueue(Queue **head, int data){  
    if(*head == NULL){  
        *head = get_node();  
        (*head)->data = data;  
        return 0;  
    }  
    enqueue(&((*head)->link), data);  
}
```

Data Structure : Queue

- Queue* deque(Queue **head, int data)
- This function searches where the data is, and deletes that.

```
Queue* deque(Queue **head, int data){  
    Queue* tmp;  
    tmp = *head;  
  
    if((*head)->data == data){  
        *head = tmp->link;  
        free(tmp);  
        return 0;  
    }  
    deque(&((*head)->link), data);  
}
```