

**Xilinx Zynq FPGA, TI DSP, MCU 기반의
프로그래밍 및 회로 설계 전문가 과정
#8**

강사 : Innova Lee(이 상훈)

학생 : 김 시윤

1. 배운내용 복습

1) Queue(FIFO)

```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#define EMPTY 0

struct node{
    int data;
    struct node *link;
};
typedef struct node queue;

queue *get_node()
{
    queue *tmp;
    tmp = (queue *)malloc(sizeof(queue));
    tmp->link=EMPTY;
    return tmp;
}

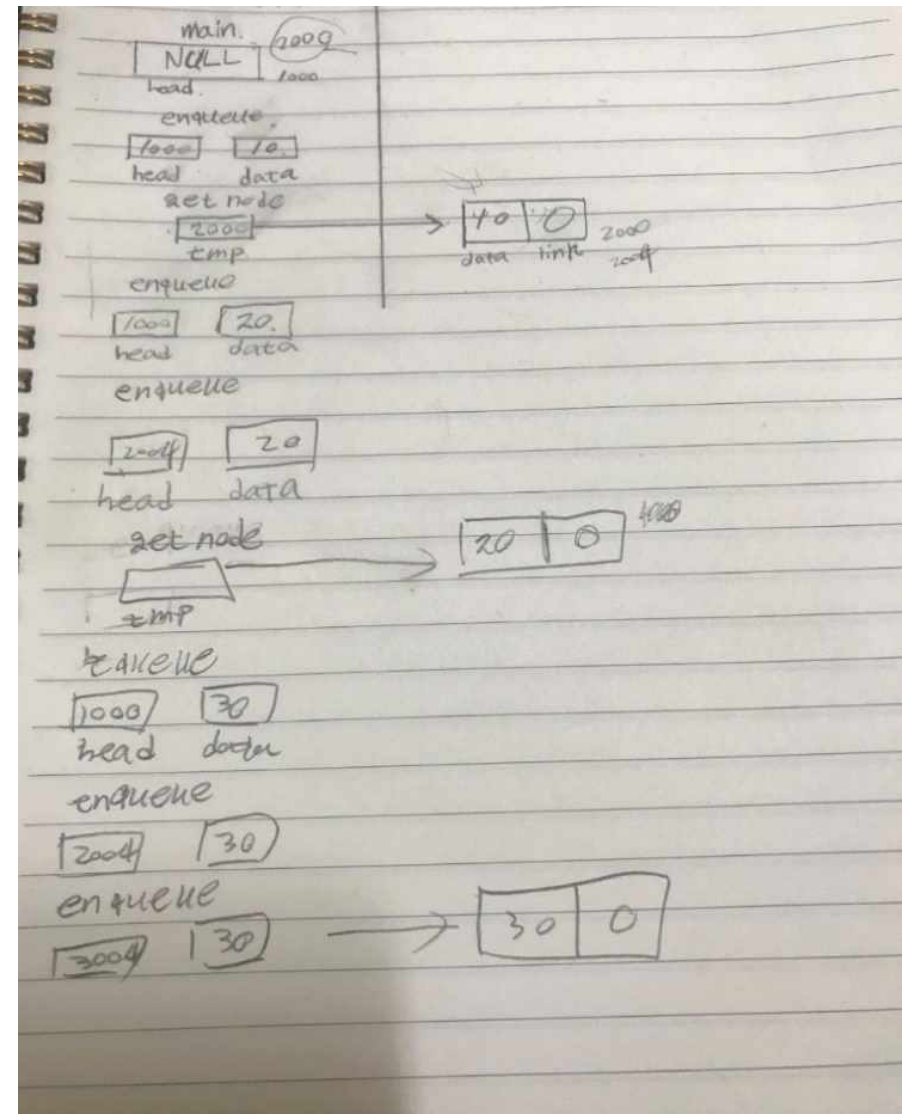
void push(queue **head,int data)
{
    queue *tmp;
    tmp = *head;
    *head = get_node();
    (*head)->data = data;
    tmp->link = *head;
}

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

void print_queue(queue *head)
{
    queue *tmp = head;
    while(tmp)
    {
        printf("%d\n",tmp->data);
        tmp=tmp->link;
    }
}

int pop(queue **head)
{
    queue *tmp;
    int num;
    tmp = *head;
    if(*head == EMPTY)
    {
        printf("Stack is empty!!\n");
        return 0;
    }
    num = tmp->data;
    *head = (*head)->link;
    free(tmp);
    return num;
}

int main(void)
{
    queue *head = EMPTY;
    enqueue(&head,10);
    enqueue(&head,20);
    enqueue(&head,30);
    printf("%d\n",pop(&head));
    printf("%d\n",pop(&head));
    printf("%d\n",pop(&head));
    printf("%d\n",pop(&head));
    return 0;
}
```



2. print 함수

```
void print_queue(queue *head)
{
    queue *tmp = head;
    while(tmp)
    {
        printf("%d\n", tmp->data);
        tmp=tmp->link;
    }
}
```

tmp 가 0이 될 때까지 반복한다.

tmp = head

head 가 0이 될 때마다 if 문을 통과하기 때문에
print 함수가 완성될 수 있다.

2. 재귀함수

enqueue(&(*head)->link, data);
자기 자신을 호출하여 계속 반복한다.
위에서는 enqueue에서 써주었지만
dequeue에서 쓰일 수 있다고한다.

아직 자료구조 연결리스트에 대해서 잘 모르겠다.
더 공부하는 시간을 갖을 것이다.

3. push

```
queue *get_node()
{
    queue *tmp;
    tmp = (queue *)malloc(sizeof(queue));
    tmp->link=EMPTY;
    return tmp;
}

void push(queue **head, int data)
{
    queue *tmp;
    tmp = *head;
    *head = get_node();
    (*head)->data = data;
    tmp->link = *head;
}
```

