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

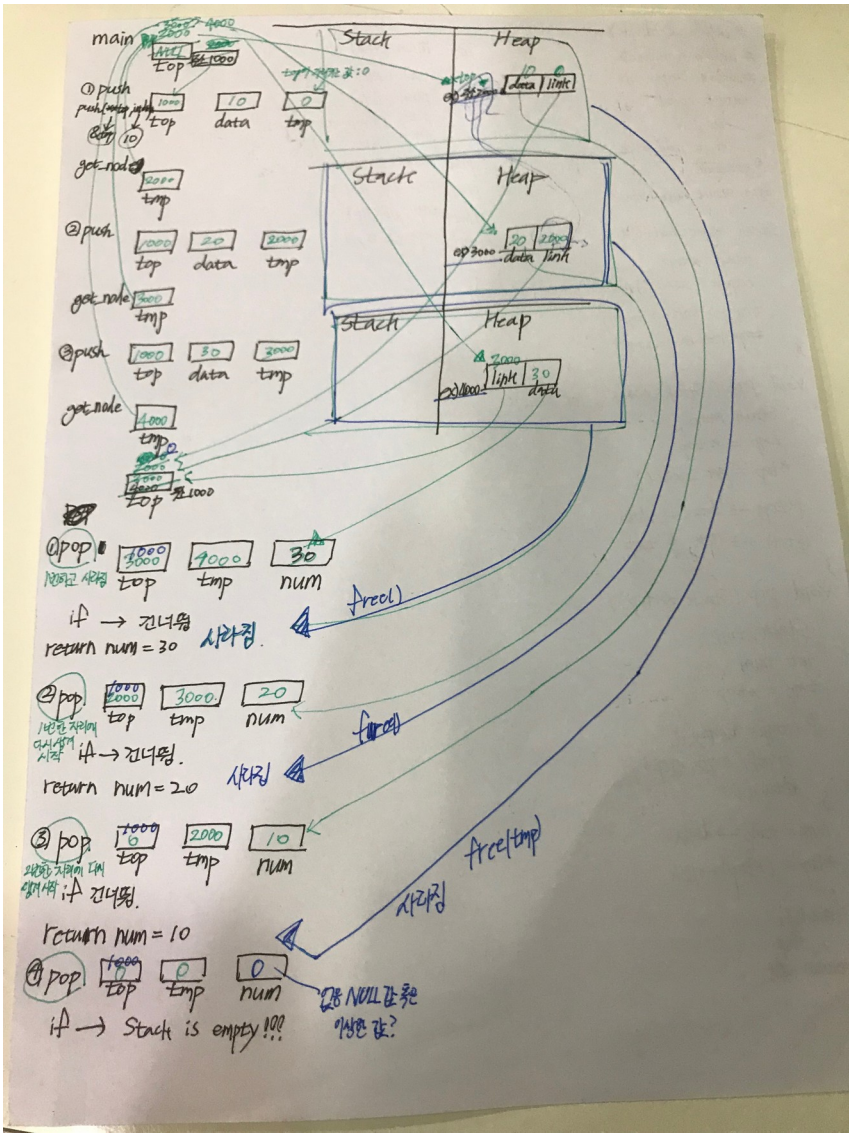
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – 정한별
hanbulkr@gmail.com

자료구조

<push_pop>

```
#include<stdio.h>
#include<malloc.h>

#define EMPTY 0

struct node{

	int data;
	struct node *link;

};

typedef struct node Stack;

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

}

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

int pop(Stack ** top)
{

		Stack *tmp;
		int num;
		tmp = *top;
		if(*top == EMPTY)
		{
			printf("Stack is empty!!!\n");
			return 0;
```

```
        }
        num = tmp ->data;
        *top=(*top)->link;
        free(tmp);
        return num;
}

int main(void)
{

        Stack *top = EMPTY;
        push(&top, 10);
        push(&top, 20);
        push( &top, 30);
        printf("%d\n", pop(&top));
        printf("%d\n", pop(&top));
        printf("%d\n", pop(&top));
        printf("%d\n", pop(&top));
        return 0;
}
```



<queue>

```
#include<stdio.h>
#include<malloc.h>

#define EMPTY 0

typedef struct __queue
{
	int data;
	struct __queue *link;

}queue;


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

void print(queue *head){

		queue *tmp=head;


		while(tmp){
			printf("값 : %d\n", tmp->data);
			tmp = tmp ->link;
		}

}

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

	//	printf("%d\n",data);
		return ;
	}

	enqueue(&((*head)->link),data);
}
```

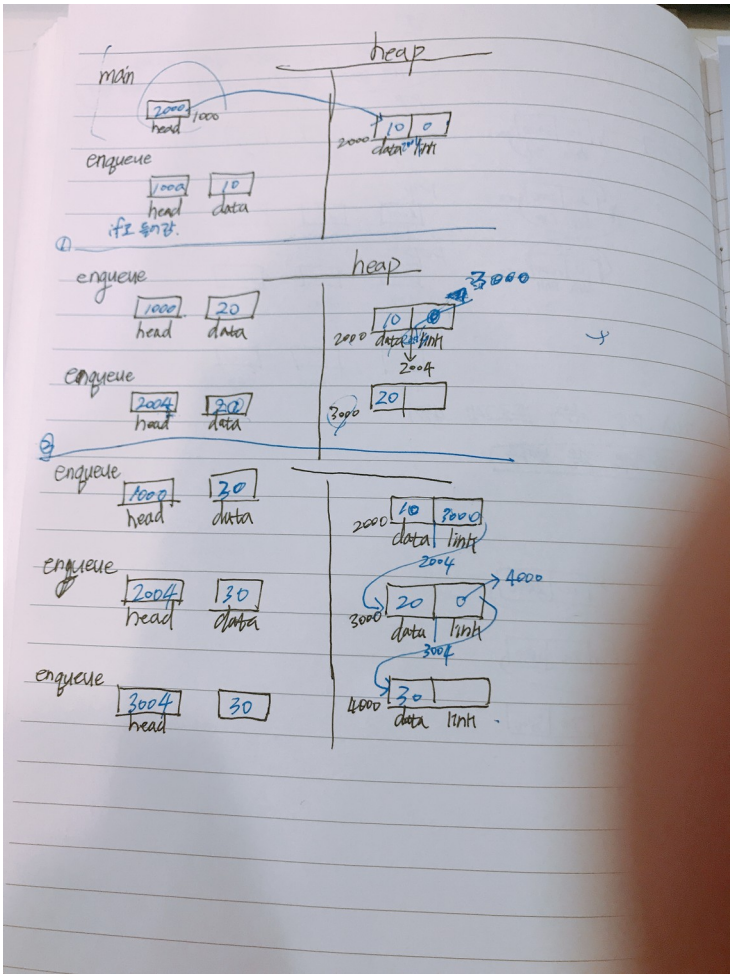
```
int main(void)
{
	//int data=10;
	queue *head=EMPTY;
	enqueue(&head, 10);
	enqueue(&head, 20);
	enqueue(&head, 30);

	print(head);
	return 0;
}
```

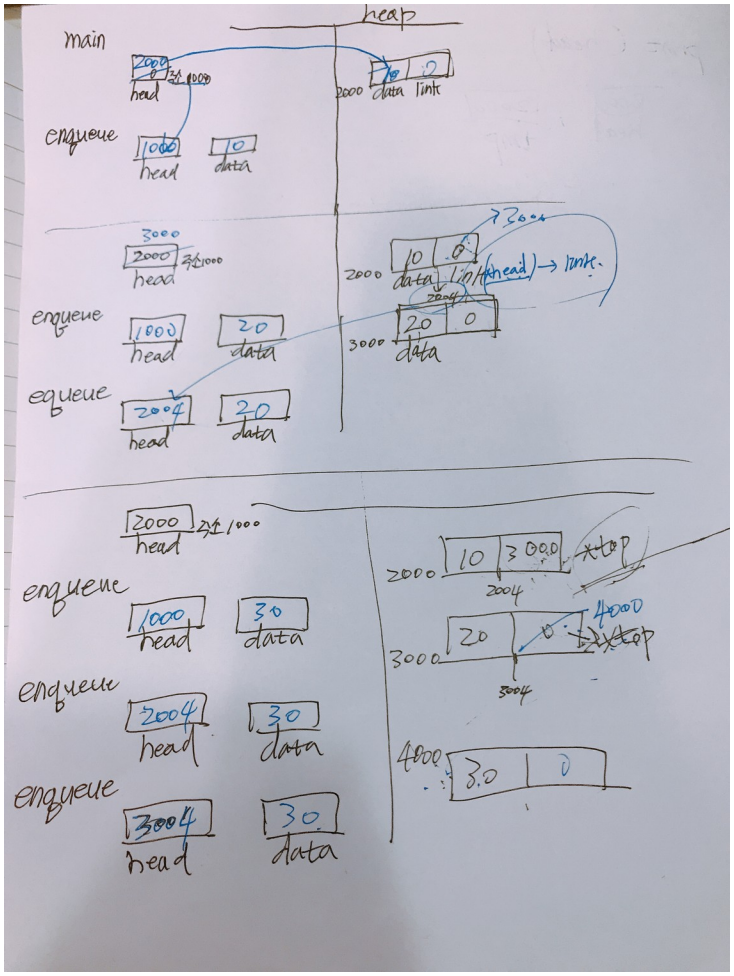
<delete>

```
#include <stdio.h>
#include <malloc.h>

#define EMPTY 0

typedef struct __queue
{
	int data;
	struct __queue *link;

}queue;

queue *get_node(){

	queue *tmp;
	tmp = (queue *)malloc(sizeof(queue));
	tmp -> link = EMPTY;

	return tmp;
}

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

}
void queue_delete(queue *head,int data)
{
	queue *tmp;
	tmp = head;
	while(tmp)
	{
		if((tmp -> data) == data){
		//	printf("같습니다.%d\n",data);
			tmp = tmp ->link;
		}
		else
		{
			printf("%d\n", tmp->data);
			tmp = tmp ->link;
		}
	}
}

void queue_delete2(queue *head, int data)
{
	queue *tmp;
	tmp = head;
	if((tmp->data) == data)
	{
		//tmp = tmp -> link;
	}
	else
	{
		tmp = tmp -> link;
	}

	queue_delete2( (tmp->link) , data);
}

int main(void){

	queue *heap = EMPTY;
	enqueue(&heap, 10);
```

```
        enqueue(&heap, 20);
        enqueue(&heap, 30);
        print_queue(heap);

//      queue_delete2(heap,20);
//      print_queue(heap);
        queue_delete(heap,20);
        print_queue(heap);

        return 0;
}
```





```c
#include <stdio.h>
#include <malloc.h>
#define Empty 0
struct node{
    int data;
    struct node *link;
};
typedef struct node Stack

Stack *get_node(){
    Stack *tmp;
    tmp = (Stack*) malloc(sizeof(Stack));
    tmp -> link = Empty;
    return tmp;
}

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

void pop(Stack **top){
    Stack tmp;
    int num;
    tmp = *top;
    if (tmp = Empty){
        printf("값이 없다.");
        return 0;
    }
    num = tmp -> data;
    *top = (*top) -> link;
    free();
    tmp
return num;
}

int main(void)
    Stack *top = Empty;
    push(&top, 10);
    push(&top, 20);
    push(&top, 30);
    print(pop(&top));
    printf(pop(&top));
    printf(pop(&top));
    printf(pop(&top));
```