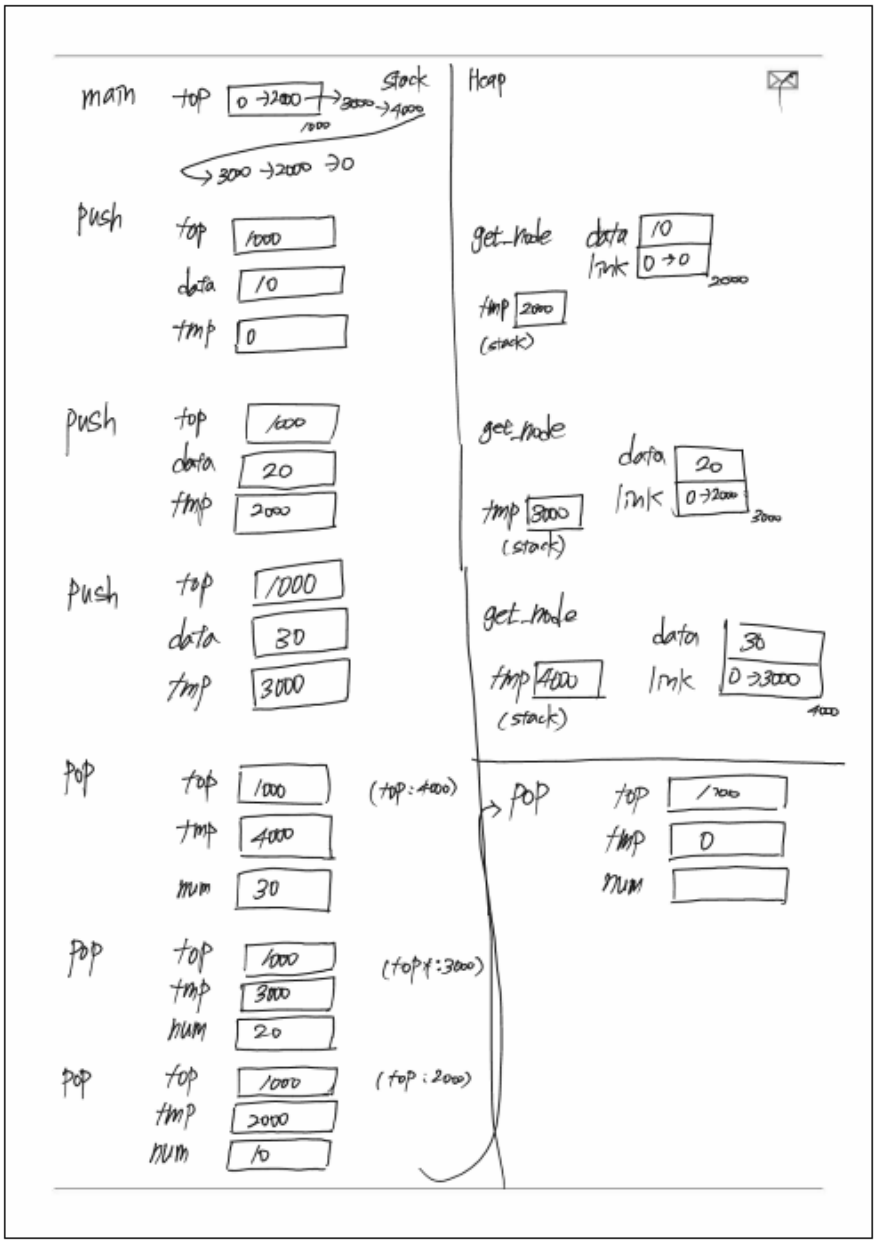
TI DSP,MCU및 Xilinux Zynq FPGA 프로그래밍 전문가 과정

이름	문지희
학생 이메일	mjh8127@naver.com
날짜	2018/3/5
수업일수	8일차
담당강사	Innova Lee(이상훈)
강사 이메일	gcccompil3r@gmail.com

목차

1. Stack

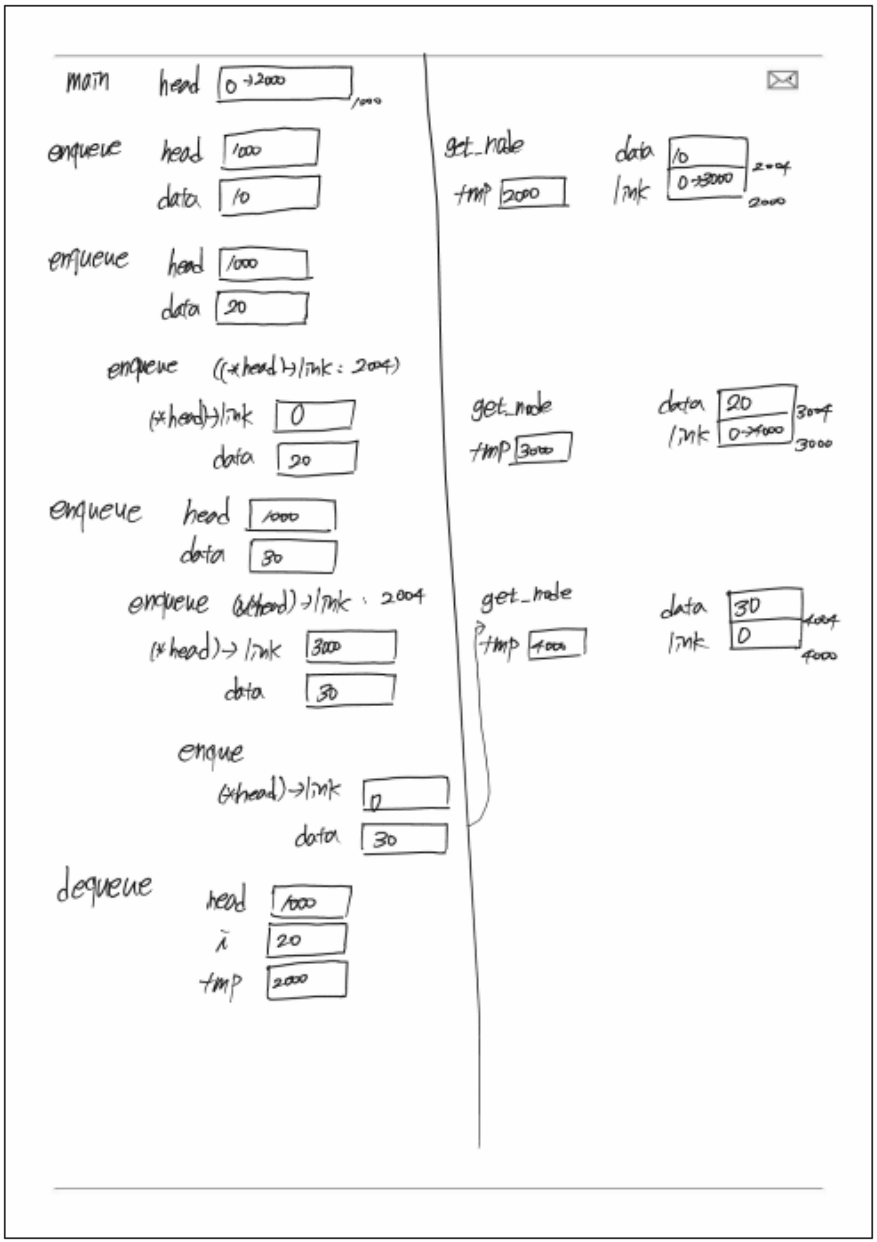
```
#include<stdio.h>
#include<malloc.h>

#define EMPTY 0

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

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

int pop(Stack **top)
{
Stack *tmp;
int num;
tmp=*top;

if(*top==EMPTY)
{
        printf("Stack is EMPTY₩n");
        return 0;
}
*top=(*top)->link;
num=tmp->data;
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;
}
```


Stack
Heap
main
top
0 → 1000
1000
3000
4000
3000 → 2000 → 0
push
top
1000
data
10
tmp
0
get_node
data
10
link
0 → 0
2000
tmp
2000
(stack)
push
top
1000
data
20
tmp
2000
get_node
data
20
link
0 → 2000
3000
tmp
3000
(stack)
push
top
1000
data
30
tmp
3000
get_node
data
30
link
0 → 3000
4000
tmp
4000
(stack)
pop
top
1000
(top: 4000)
tmp
4000
num
30
pop
top
1000
tmp
0
num
pop
top
1000
(top: 3000)
tmp
3000
num
20
pop
top
1000
(top: 2000)
tmp
2000
num
10

2. queue

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

typedef struct __queue{
        int data;
        struct __queue *link;
}queue;
queue *get_node(void){
        queue *tmp;
        tmp=(queue*)malloc(sizeof(queue));
        tmp->link=EMPTY;
        return tmp;
}
void enqueue(queue **head,int data){
        if(*head==EMPTY)
                *head=get_node();
                (*head)->data=data;
                return ;
        }
        enqueue(&(*head)->link, data);
}
void dequeue(queue **head,int i){
        queue *tmp=*head;
        while(tmp)
        {
                if(i==tmp->data)
                        { tmp=tmp->link;}
                else
                        { printf("%d₩n",tmp->data);
                          tmp=tmp->link; }
        }
}
```

```
#include<stdio.h>


void print(queue **head){
        queue *tmp=*head;
        while(tmp)
        {
                printf("%d₩n",tmp->data);
                tmp=tmp->link;
        }
}
int main(void){
        int i=20;

        queue *head=EMPTY;
        enqueue(&head,10);
        enqueue(&head,20);
        enqueue(&head,30);
        print(&head);
        dequeue(&head,i);
        return 0;
}
```


main
head
0 → 2000
1000
enqueue
head
1000
data
10
get_node
tmp
2000
data
10
link
0 → 3000
2004
2000
enqueue
head
1000
data
20
enqueue
((*head)→link : 2004)
(*head)→link
0
data
20
get_node
tmp
3000
data
20
link
0 → 4000
3004
3000
enqueue
head
1000
data
30
enqueue
(&head)→link : 2004
(*head)→link
3000
data
30
get_node
tmp
4000
data
30
link
0
4004
4000
enque
(*head)→link
0
data
30
dequeue
head
1000
i
20
tmp
2000