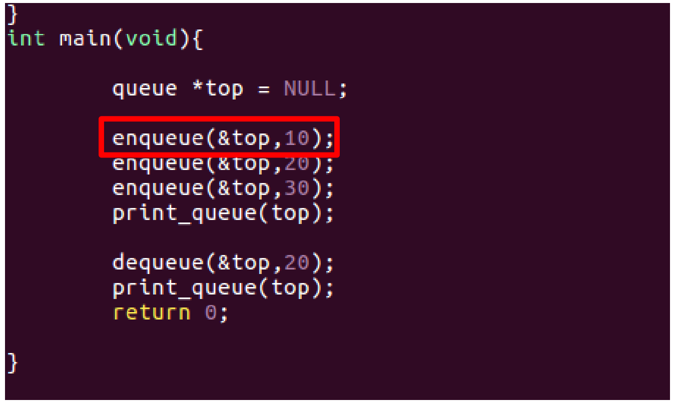
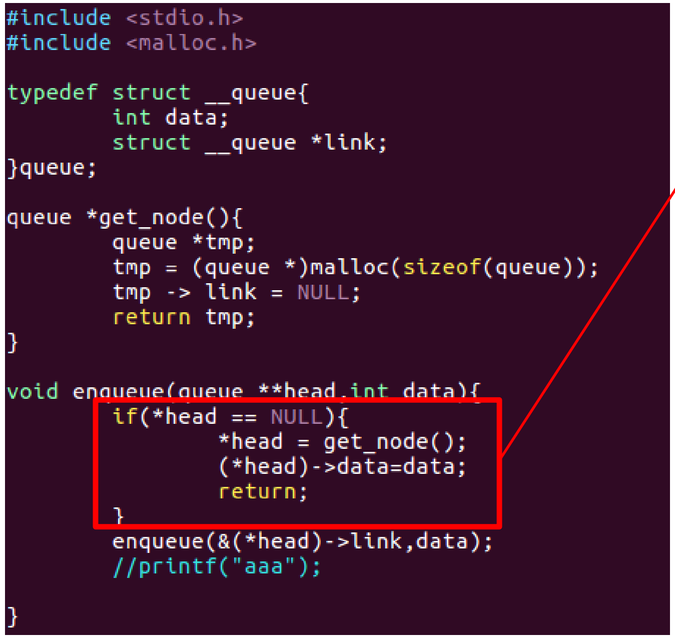
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – 문한나
mhn97@naver.com

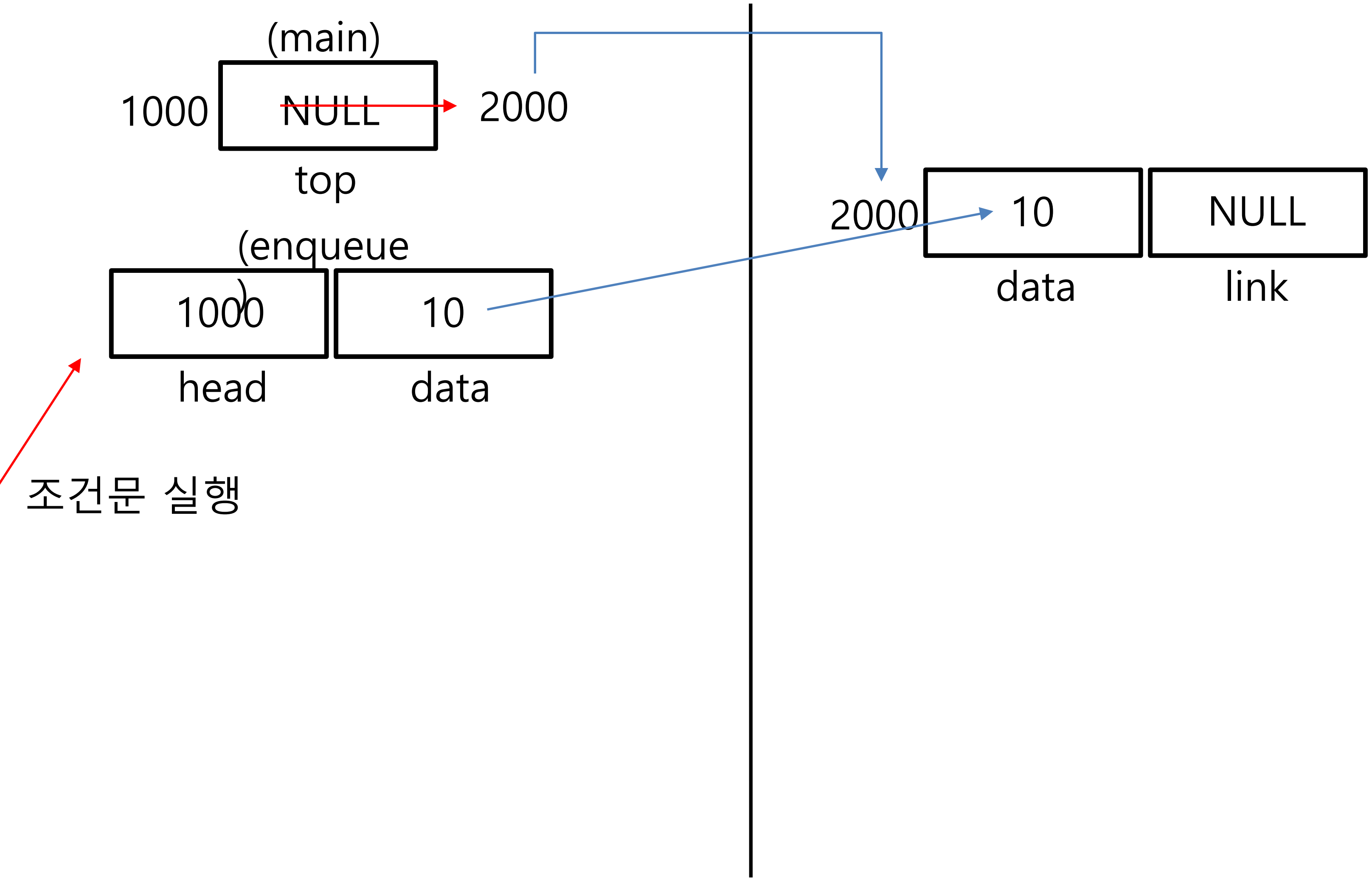
연결리스트 예제 그림 그리기

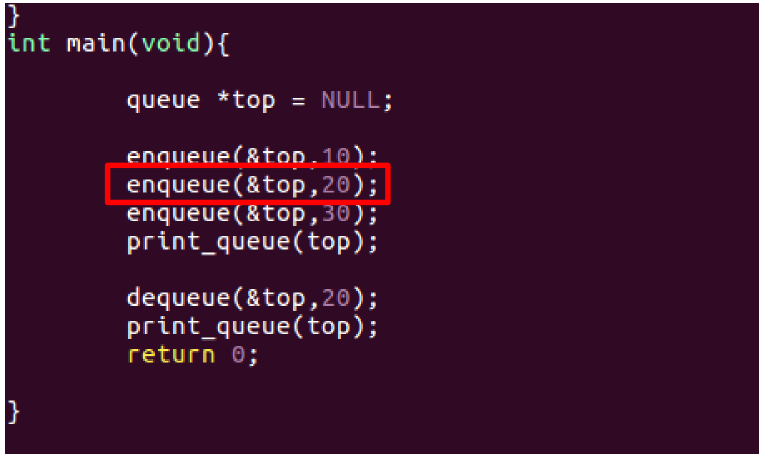


```
}
int main(void){

        queue *top = NULL;

        enqueue(&top,10);
        enqueue(&top,20);
        enqueue(&top,30);
        print_queue(top);

        dequeue(&top,20);
        print_queue(top);
        return 0;

}
```

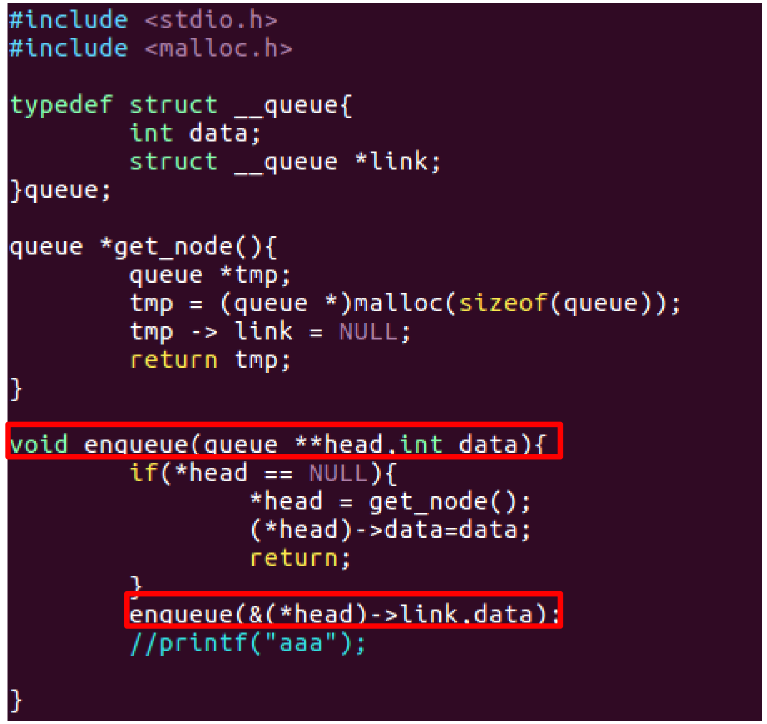


```
#include <stdio.h>
#include <malloc.h>

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

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

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

}
```



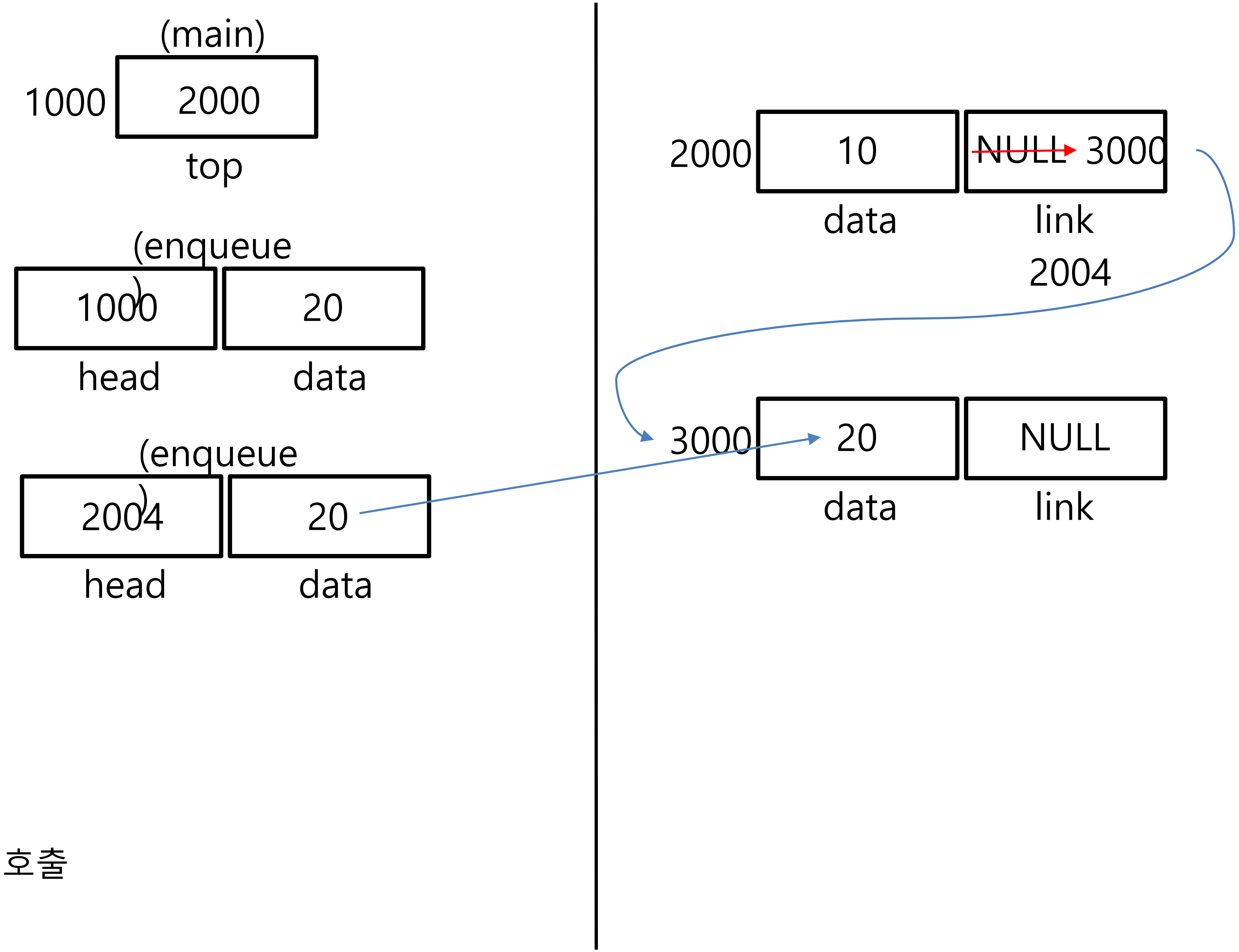


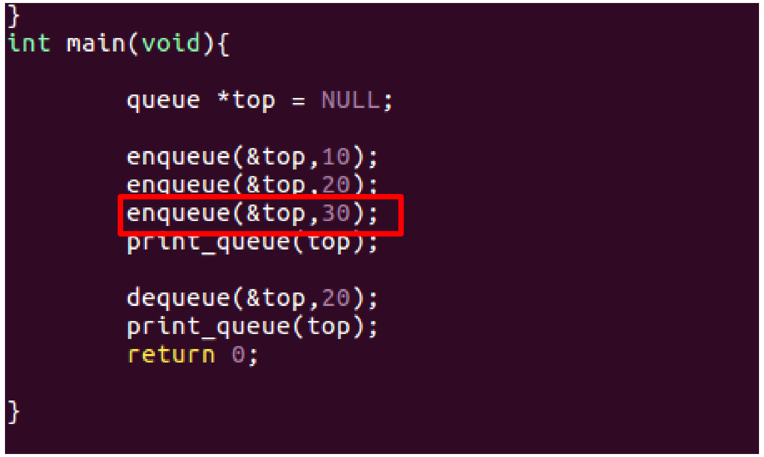
```
}
int main(void){

        queue *top = NULL;

        enqueue(&top,10);
        enqueue(&top,20);
        enqueue(&top,30);
        print_queue(top);

        dequeue(&top,20);
        print_queue(top);
        return 0;

}
```



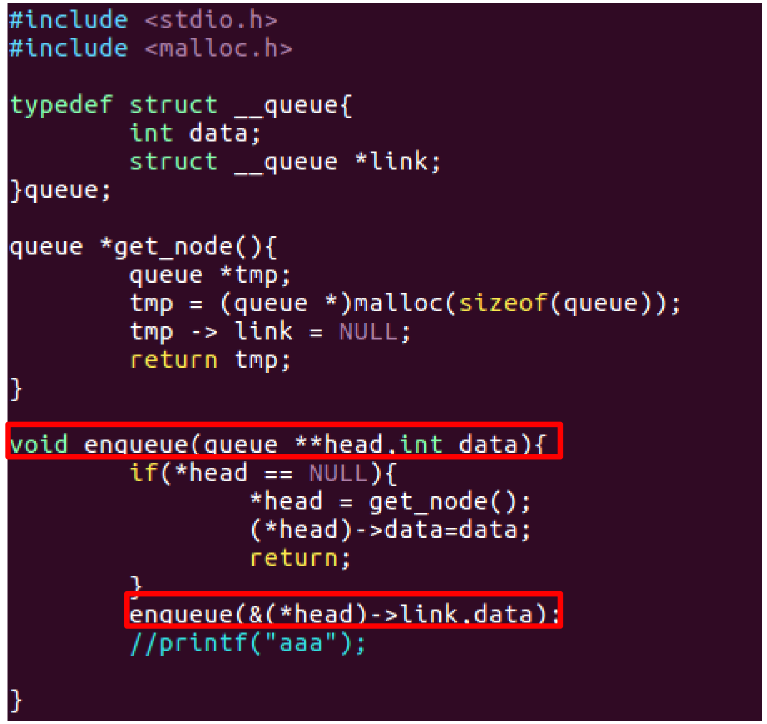
```
#include <stdio.h>
#include <malloc.h>

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

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

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

}
```

재귀호출

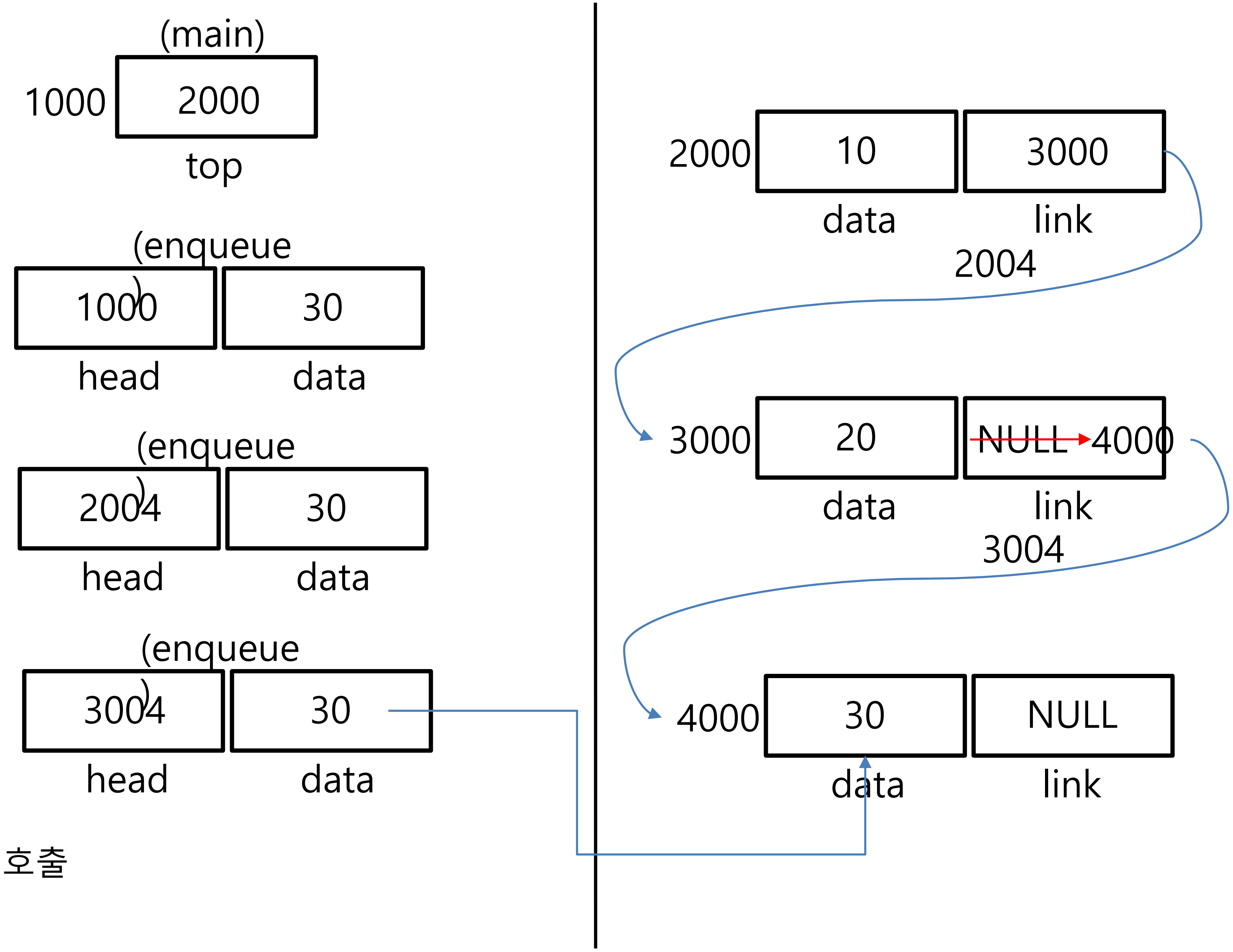




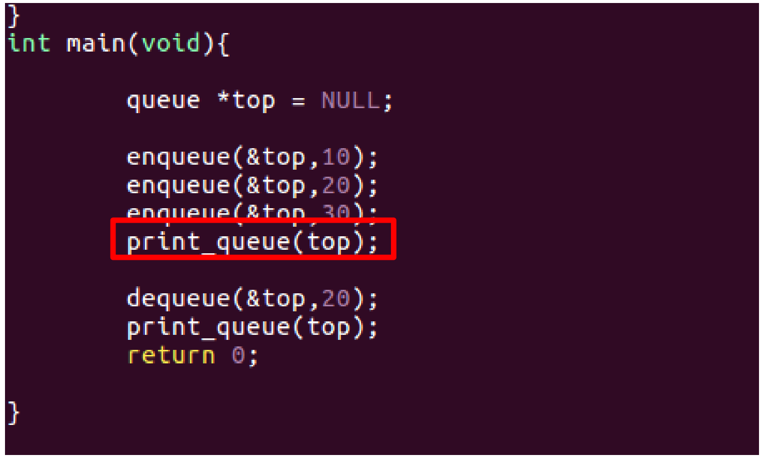
```
}
int main(void){

        queue *top = NULL;

        enqueue(&top,10);
        enqueue(&top,20);
        enqueue(&top,30);
        print_queue(top);

        dequeue(&top,20);
        print_queue(top);
        return 0;

}
```



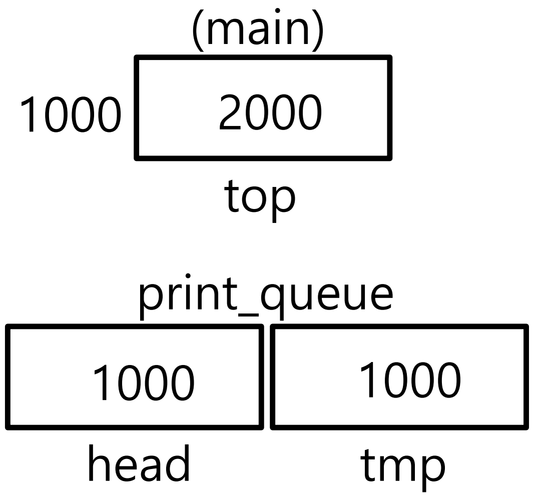
```
#include <stdio.h>
#include <malloc.h>

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

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

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

}
```

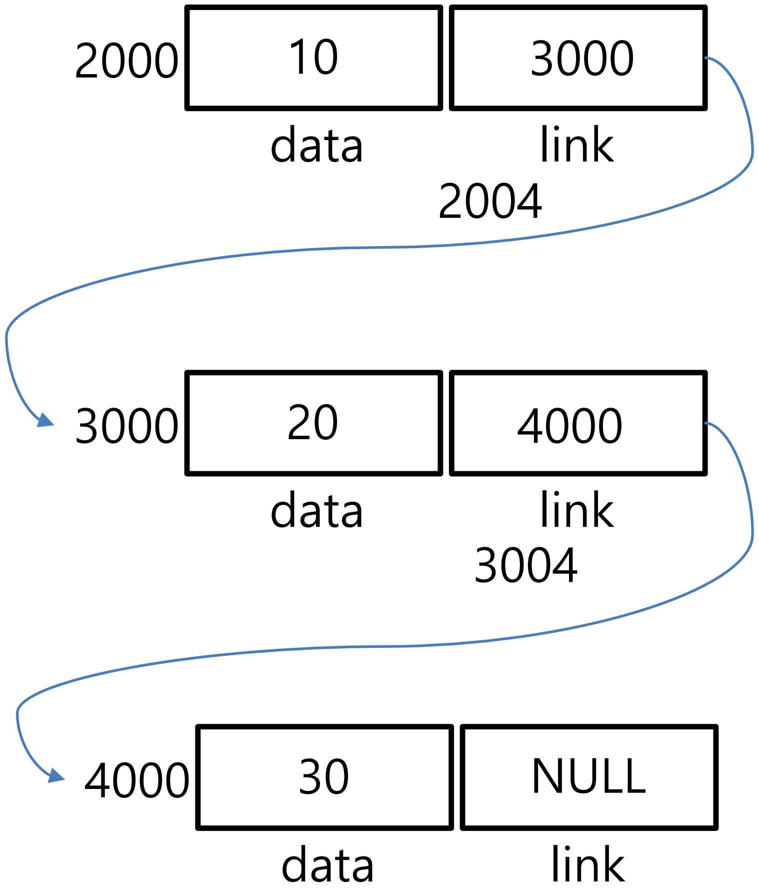


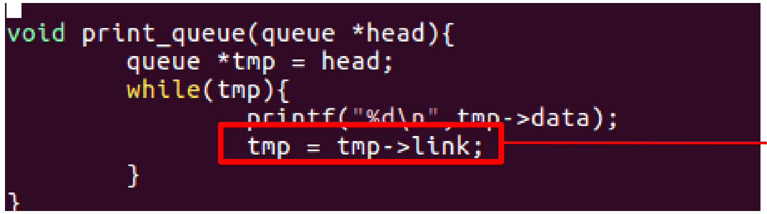
재귀호출



```
}
int main(void){

        queue *top = NULL;

        enqueue(&top,10);
        enqueue(&top,20);
        enqueue(&top,30);
        print_queue(top);

        dequeue(&top,20);
        print_queue(top);
        return 0;

}
```

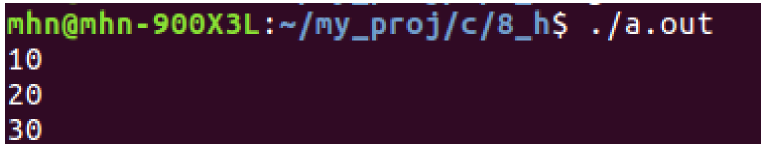




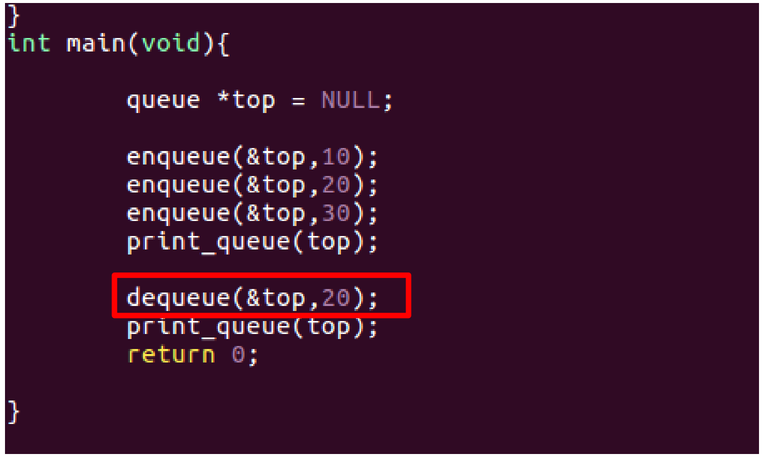


```
void print_queue(queue *head){
        queue *tmp = head;
        while(tmp){
                printf("%d\n", tmp->data);
                tmp = tmp->link;
        }
}
```

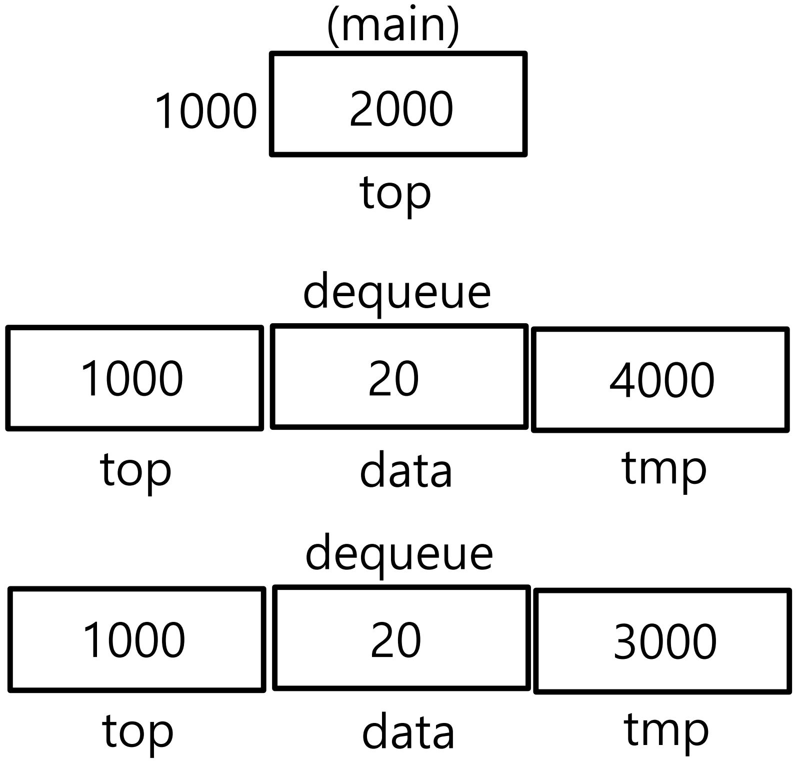
링크를 따라가며 값을 찍는다



```
mhn@mhn-900X3L:~/my_proj/c/8_h$ ./a.out
10
20
30
```



```
}
int main(void){

        queue *top = NULL;

        enqueue(&top,10);
        enqueue(&top,20);
        enqueue(&top,30);
        print_queue(top);

        dequeue(&top,20);
        print_queue(top);
        return 0;

}
```



(main)

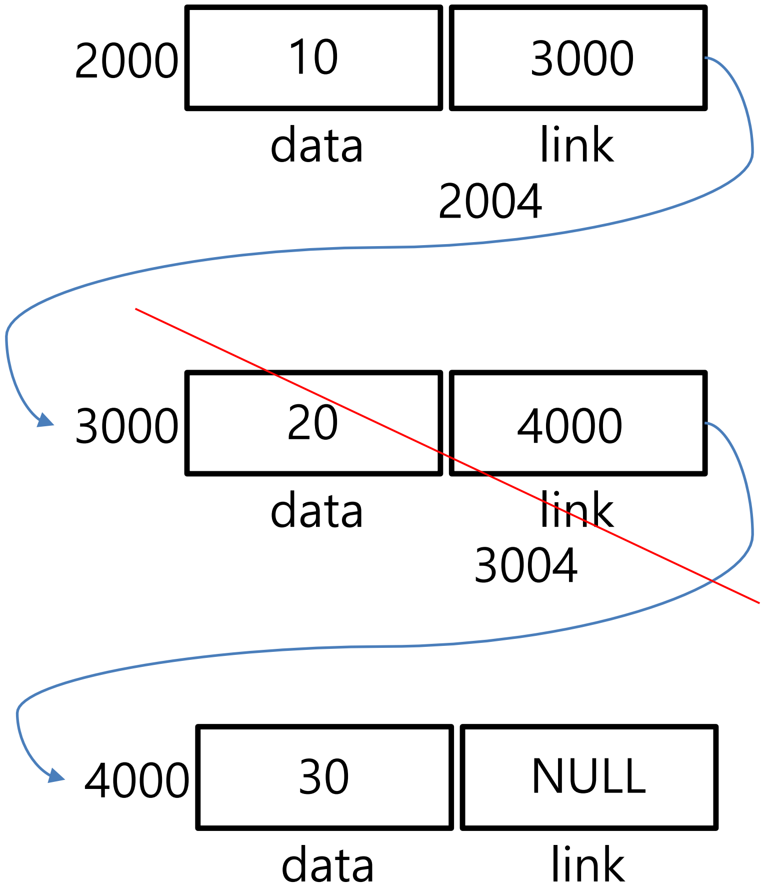
1000 2000 top

dequeue

1000 top | 20 data | 4000 tmp

dequeue

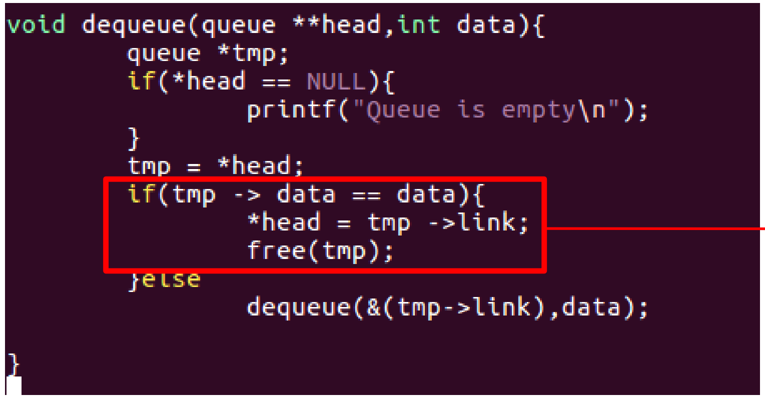
1000 top | 20 data | 3000 tmp



2000 10 data | 3000 link 2004

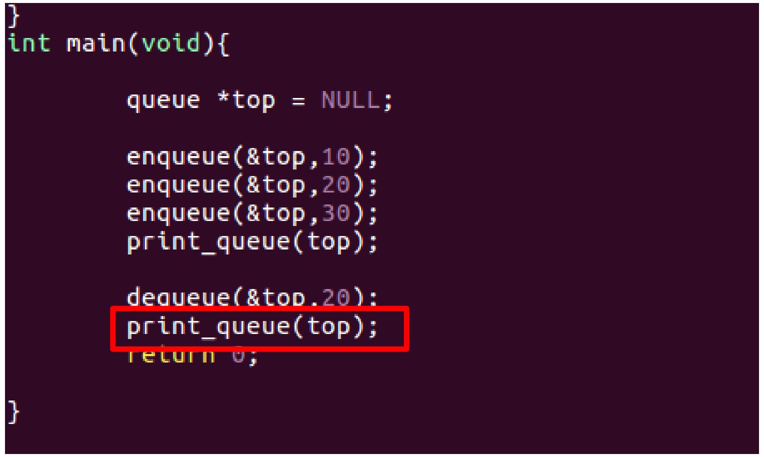
3000 20 data | 4000 link 3004

4000 30 data | NULL link

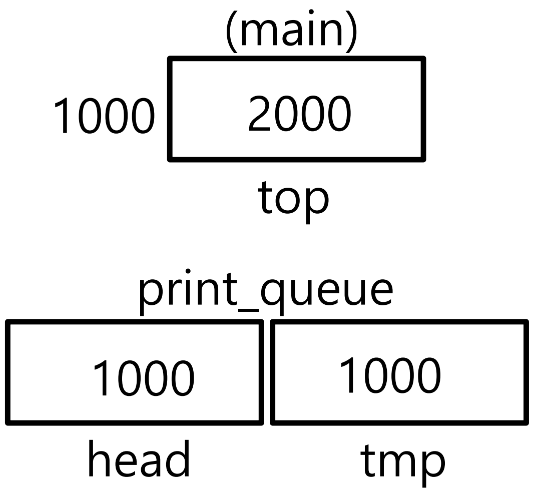


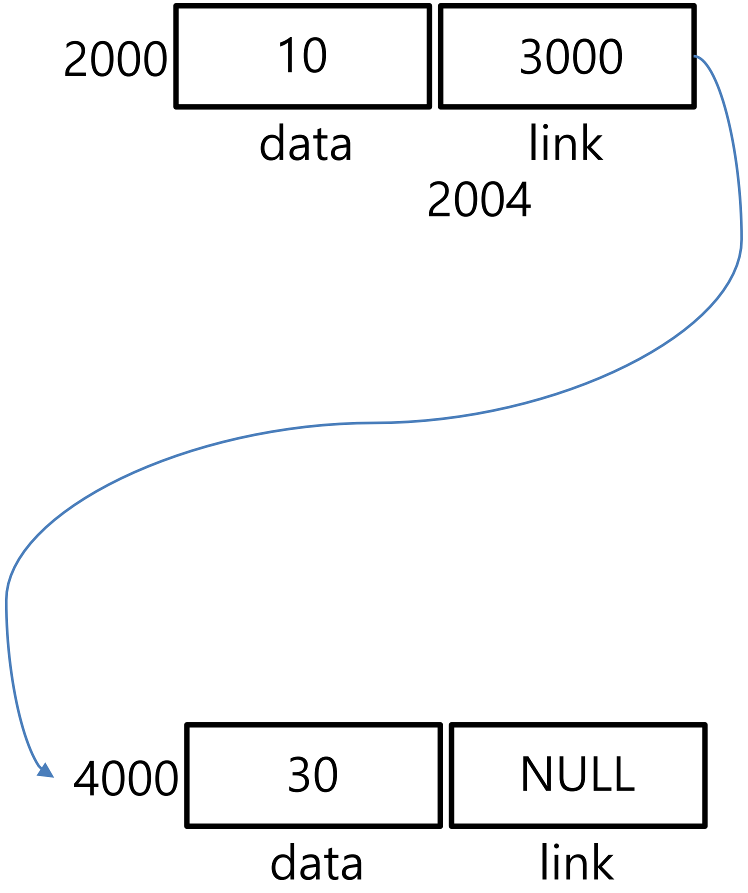
```
void dequeue(queue **head,int data){
        queue *tmp;
        if(*head == NULL){
                printf("Queue is empty\n");
        }
        tmp = *head;
        if(tmp -> data == data){
                *head = tmp ->link;
                free(tmp);
        }else
                dequeue(&(tmp->link),data);
}
```

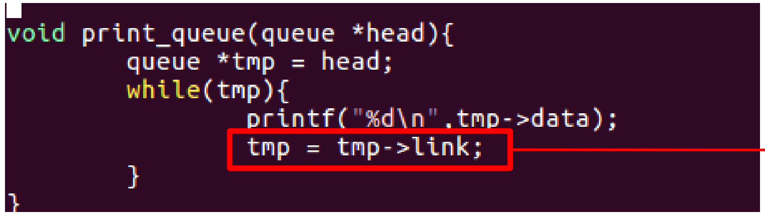
heap에 들어있는 data와
값이 일치하면 데이터할당을 해제한다



```
}
int main(void){

        queue *top = NULL;

        enqueue(&top,10);
        enqueue(&top,20);
        enqueue(&top,30);
        print_queue(top);

        dequeue(&top,20);
        print_queue(top);
        return 0;

}
```

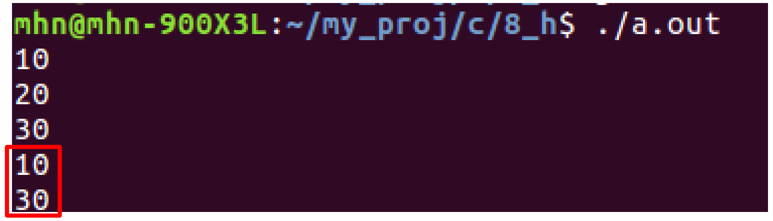






```
void print_queue(queue *head){
        queue *tmp = head;
        while(tmp){
                printf("%d\n",tmp->data);
                tmp = tmp->link;
        }
}
```

링크를 따라가며 값을 찍는다



```
mhn@mhn-900X3L:~/my_proj/c/8_h$ ./a.out
10
20
30
10
30
```