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

강사 – 이상훈
gcccompil3r@gmail.com

수강생 – 서재언
20640@naver.com

1. day_8/01.c

– 스택구현

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

#define EMPTY 0

typedef struct _stack{

int data;
struct _stack *link;

}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();  //push 될때마다 top 또한 계속 생성 //a->b

   (*top)->data = data;

   (*top)->link = tmp;

}
```

1. day_8/01.c

– 스택구현

```
int pop(stack **top)
{
  int data;

  stack *tmp;

  if((*top)->link == EMPTY)
  {

     printf("stack is empty\n");
  }
   tmp = (*top);
   data = tmp->data;
  *top = tmp->link;


 free(tmp);

 return data;

}
```

1. day_8/01.c

– 스택구현

```
int main(void)
{

   stack *top;

   push(&top,10);

   push(&top,20);
   push(&top,30);

   printf("First Pop: %d\n", pop(&top));
   printf("second Pop: %d\n", pop(&top));
  //printf("third Pop: %d\n", pop(&top));


   return 0;
}
```

1. day_8/02.c

- 큐 구현

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

#define EMPTY 0

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

}queue;


void enqueue(queue **head, int data)
{

   queue *tmp;

   if(*head == NULL)
   {
     *head=(queue *)malloc(sizeof(queue));
     (*head)->data = data;

     return;
    }

    tmp = (queue *)malloc(sizeof(queue));

    tmp->data = data;

   (*head)->link = tmp;

}
```

1. day_8/02.c

- 큐 구현

```c
int dequeue(queue **head)
{
  queue *tmp;

  int data;

  tmp = *head;

  data = tmp->data;

  *head = tmp->link;

  free(tmp);

  return data;

}


int main(void)
{

  queue *head = EMPTY;

  enqueue(&head, 10);
  enqueue(&head, 20);

  printf("%d st: %d\n",1, dequeue(&head));
  printf("%d st: %d\n",2, dequeue(&head));


  return 0;
}


```

1. day_8/02.c

- 큐 구현

```c
int dequeue(queue **head)
{
    queue *tmp;

    int data;

    tmp = *head;

    data = tmp->data;

    *head = tmp->link;

    free(tmp);

    return data;

}


int main(void)
{

    queue *head = EMPTY;

    enqueue(&head, 10);
    enqueue(&head, 20);

    printf("%d st: %d\n",1, dequeue(&head));
    printf("%d st: %d\n",2, dequeue(&head));


    return 0;
}


```