

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

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 장성환

redmk1025@gmail.com

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>

typedef enum _menu{
    INPUT=1,
    STATEMENT=2,
    SORT=3,
    DELETE=4,
    EXIT=5,
    TRUE=6,
    FALSE=7
}menu;

typedef struct __grades{
    int math;
    int english;
    char name[20];
}grade;

typedef struct _queue{
    grade *data;
    struct _queue *link;
}queue;

void enqueue(queue **head,grade *data){
    queue * tmp;

    if(*head == NULL){

```

파일을 읽고, 해당 값을 읽어서 큐 자료구조에 저장하는 것까지는 완료 되었다.

복불하지 않고 처음부터 만들다 보니

시간이 부족하여 아래에 있는 나머지 내용은 완료하지 못하였다.

-나머지 남은 부분

자료구조의 있는 내용을 적절한 알고리즘을 이용하여

통계기능, 정렬기능, 입력기능, 삭제기능을 구현 후에

파일에 저장

```
    tmp = (queue*)malloc(sizeof(queue));
    tmp->link = NULL;
    tmp->data = data;
    *head = tmp;
}
else{
    enqueue(&(*head)->link,data);
}
}

grade* dequeue(queue *head){
    grade *tmp;
    if(head == NULL){
        printf("no data!\n");
        return NULL;
    }
    tmp = head->data;
    head=head->link;
    return tmp;
}

void printf_queue(queue *head){
    if(head == NULL)
        return;
    printf("이름 수학 영어\n");
    printf("%s ",head->data->name);
    printf("%3d ",head->data->math);
    printf("%3d",head->data->english);
    printf("\n");
    printf_queue(head->link);
}
```

int readOneCharacter(int db,char buf[]){ //divided by " " and "\n", output is string

```
int i=0;
int nread;
char tmp;
while(nread = read(db,&tmp,1)>0){
    if(tmp == ' '|| tmp == '\n'){
        buf[i] = '\0';
        return TRUE;
    }
    buf[i] = tmp;
    i++;
}
return FALSE;
}
```

int main(int argc, char **argv){

```
int menu;
int read_db,write_db;
char buf[1024];
queue *head=NULL;
grade *person=NULL;
int cnt =0;
```

```
read_db=open("quiz2.txt",O_RDONLY,0644);
```

```
while(menu != FALSE){
    menu=readOneCharacter(read_db,buf);
```

```

        if(cnt ==0){
            person = (grade*)malloc(sizeof(grade));
            strcpy(person->name,buf);
            cnt++;
        }
        else if(cnt == 1){
            person->math = atoi(buf);
            cnt++;
        }
        else if(cnt == 2){
            person->english = atoi(buf);
            cnt =0;
            enqueue(&head,person);
        }
    } // inital data load

//  readData(head,person,buf,read_db);
printf_queue(head);

/*
while(1){
    printf("*Grade management system*\n");
    printf("1-input grades\n2-numeric statement\n3-sort by grade\n4-
delete\n5-exit\n");
    scanf("%d",&menu);
    if(menu ==EXIT){
        printf("thank you!\n");
        break;
    }
} //continuous input

*/

```

<pre>return 0; }</pre>	
----------------------------	--