자료구조응용

07. 선형큐, 동적할당 환형큐 (10점)

2022.3.28.

1. [정적할당배열을 이용한 선형큐(linear queue)] 다음과 같은 선형 큐를 생성하고 실행 예와 같이 수행되는 프로그램을 작성하라. 이를 위해, addq, deleteq, queueFull, queueEmpty 함수를 구현하여야 한다.(5점)

[자료형과 함수의 정의]

```
#define MAX_QUEUE_SIZE 5
typedef struct {
            int id;                          // unique id
            char name[MAX_NAME_SIZE];        // last name
} element;
element queue[MAX_QUEUE_SIZE];
int rear = -1;
int front = -1;
```

```
void addq(element item)
{/* add an item to the queue */
   if (rear == MAX_QUEUE_SIZE-1)
      queueFull();
   queue[++rear] = item;
}
```

Program 3.5: Add to a queue

```
element deleteq()
{/* remove element at the front of the queue */
   if (front == rear)
      return queueEmpty(); /* return an error key */
   return queue[++front];
}
```

Program 3.6: Delete from a queue

[구현 조건]

① 사용자입력으로부터 데이터 추출을 위해 gets, strtok, strcmp, sscanf, strlen 등을 사용

② **addq, deleteq** 함수는 교재 코드를 **수정 없이** 그대로 사용할 것

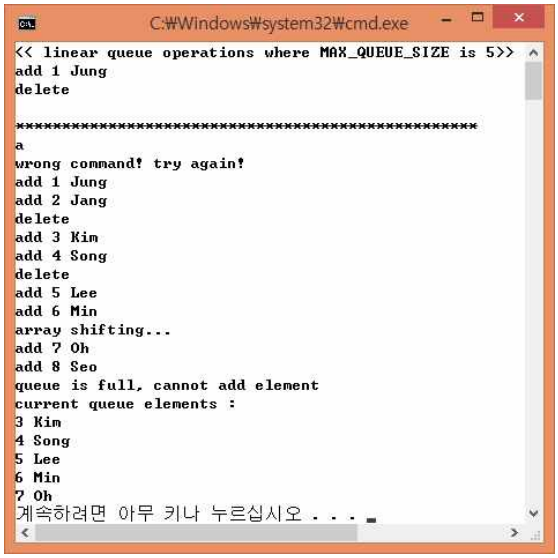
③ **queueFull**은 아래와 같이 정의함

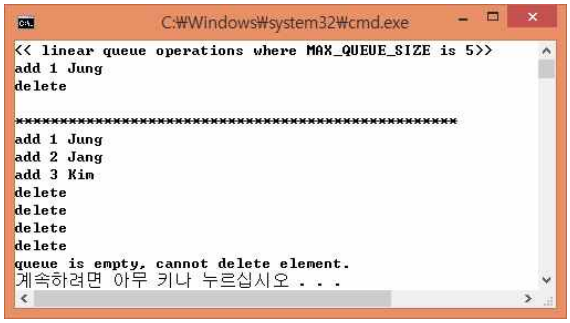
- √ front == -1 의 경우 아래와 같이 구현
  - "Queue is full, cannot add element!" 메시지를 출력
  - deleteq를 호출하여 현재 큐내용을 출력
  - exit(EXIT_FILURE) 호출
- √ 그 이외에 대해서는 큐의 항목들을 이동

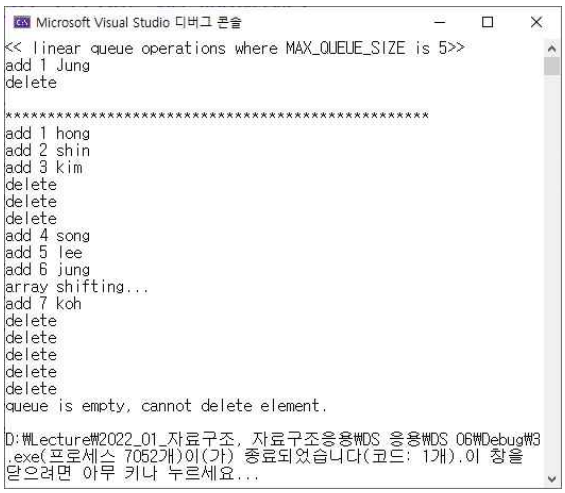
④ **queueEmpty**는 에러메시지를 출력한 후 임의의 에러키(ex: -1)를 가지는 element형 변수를 반환

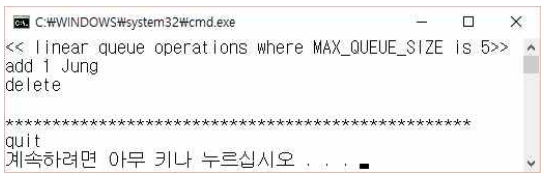
⑤ 잘못된 커맨드를 입력하면 에러메시지를 출력 후 다시 사용자 입력을 받도록 함

[실행 예]


C:₩Windows₩system32₩cmd.exe
<< linear queue operations where MAX_QUEUE_SIZE is 5>>
add 1 Jung
delete
****
a
wrong command! try again!
add 1 Jung
add 2 Jang
delete
add 3 Kim
add 4 Song
delete
add 5 Lee
add 6 Min
array shifting...
add 7 Oh
add 8 Seo
queue is full, cannot add element
current queue elements :
3 Kim
4 Song
5 Lee
6 Min
7 Oh
계속하려면 아무 키나 누르십시오 . . .


C:₩Windows₩system32₩cmd.exe
<< linear queue operations where MAX_QUEUE_SIZE is 5>>
add 1 Jung
delete
****
add 1 Jung
add 2 Jang
add 3 Kim
delete
delete
delete
delete
queue is empty, cannot delete element.
계속하려면 아무 키나 누르십시오 . . .


Microsoft Visual Studio 디버그 콘솔
<< linear queue operations where MAX_QUEUE_SIZE is 5>>
add 1 Jung
delete
****
add 1 hong
add 2 shin
add 3 kim
delete
delete
delete
add 4 song
add 5 lee
add 6 jung
array shifting...
add 7 koh
delete
delete
delete
delete
delete
queue is empty, cannot delete element.
D:₩Lecture₩2022_01_자료구조, 자료구조응용₩DS 응용₩DS 06₩Debug₩3
.exe(프로세스 7052개)이(가) 종료되었습니다(코드: 1개).이 창을
닫으려면 아무 키나 누르세요...


C:₩WINDOWS₩system32₩cmd.exe
<< linear queue operations where MAX_QUEUE_SIZE is 5>>
add 1 Jung
delete
****
quit
계속하려면 아무 키나 누르십시오 . . .

2. [동적할당배열을 이용한 환형큐(circular queue)] 1번 문제의 프로그램을 동적할당 배열을 이용한 환형큐 프로그램으로 수정하라. queueFull 함수는 queue capacity를 두 배로 확장하는 Program 3.10으로 구현한다.

[자료형과 함수의 정의]

```
typedef struct {
            int id;                          // unique id
            char name[MAX_NAME_SIZE];        //last name
} element;
element *queue;
int capacity = 2;
int rear = 0;
int front = 0;
```

```
element deleteq()
{/* remove front element from the queue */
   element item;
      if (front == rear)
         return queueEmpty(); /* return an error key */
      front = (front+1) % MAX_QUEUE_SIZE;
      return queue[front];
}
```

Program 3.8: Delete from a circular queue

```
void addq(element item)
{/* add an item to the queue */
   rear = (rear+1) % capacity;
   if (front == rear)
      queueFull(); /* double capacity */
   queue[rear] = item;
}
```

Program 3.9: Add to a circular queue

```
void queueFull()
{   int start;
    /* allocate an array with twice the capacity */
    element* newQueue;
    MALLOC(newQueue, 2 * capacity * sizeof(*queue));

    /* copy from queue to newQueue */
    start = (front+1)  % capacity;   rear--;
    if (start  <  2)
        /* no wrap around */
        copy(queue+start, queue+start+capacity-1, newQueue);
    else
    {/* queue wraps around */
        copy(queue+start, queue+capacity, newQueue);
        copy(queue, queue+rear+1, newQueue+capacity-start);
    }

    /* switch to newQueue */
    front = 2 * capacity - 1;
    rear = capacity -1 ;
    capacity *= 2;
    free(queue);
    queue = newQueue;
}
```

Program 3.10: Doubling queue capacity

[구현 조건]

① 사용자입력으로부터 데이터 추출을 위해 gets, strtok, strcmp, sscanf, strlen 등을 사용

② 전역변수 capacity, front, rear의 초기값은 각각 2, 0, 0

③ **addq, deleteq** 함수는 **수정 없이** 사용하기

(단, deleteq의 MAX_QUEUE_SIZE를 capacity로 수정함)

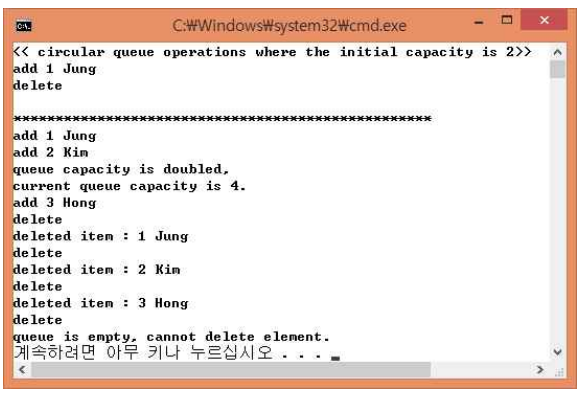
④ **circular queue**를 **전역변수 element *queue;**로 선언

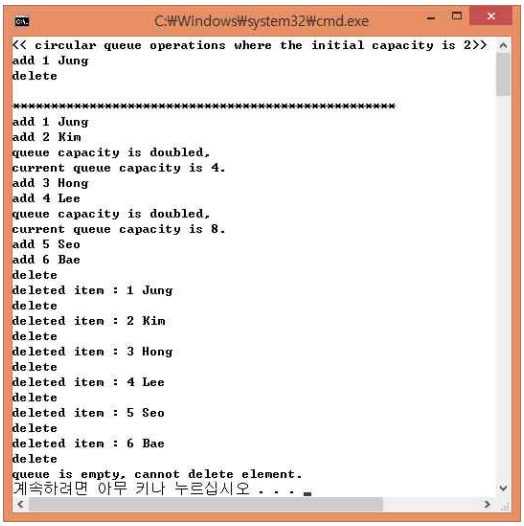
⑤ main에서 동적할당으로 capacity 2의 초기queue를 생성함

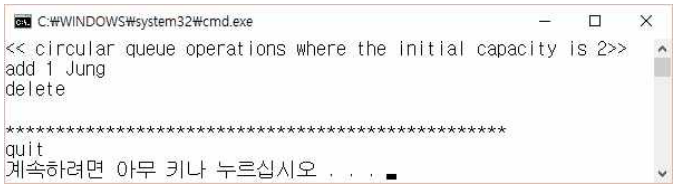
⑥ **copy 함수**를 **직접 정의**해야 함

⑦ 필요하다면 queueFull 함수의 아래 부분에 정보출력코드를 일부 추가할 수 있음

[실행 예]







■ 제출 형식

- 솔루션 이름 : DS 07
- 프로젝트 이름 : 1, 2
- 각 소스파일에 주석처리

  "학번 이름"

  "본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다."
- 실행화면을 캡쳐한 보고서를 작성 후 pdf 파일로 변환하여 솔루션 폴더에 포함
- 솔루션 정리 메뉴를 수행 후 전체 솔루션을 "학번.zip"으로 압축하여 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!

- 1차 마감 (LMS 과제 마감일) : 수업일 자정

- 2차 마감 (LMS 과제 열람기한) : 수업 익일 자정(만점의 50%, 반올림)

- 1, 2차 마감 이외의 제출은 허용하지 않습니다. (이메일 제출 불가!)