

자료구조응용

07. 스택과 큐 (10점)

2020.4.6. (월)

1. [정적할당배열의 스택] 다음과 같은 스택을 생성하고 실행 예와 같이 수행되는 프로그램을 작성하라. 이를 위해, push, pop, stackEmpty, stackFull 함수를 구현하여야 한다. (5점)

[자료형과 함수의 정의]

```
#define MAX_STACK_SIZE 5
typedef struct {
    int id;                // unique id
    char name[MAX_NAME_SIZE]; // last name
} element;
element stack[MAX_STACK_SIZE];
int top = -1;
```

```
void push(element item)
{ /* add an item to the global stack */
    if (top >= MAX_STACK_SIZE-1)
        stackFull();
    stack[++top] = item;
}
```

Program 3.1: Add an item to a stack

```
element pop()
{ /* delete and return the top element from the stack */
    if (top == -1)
        return stackEmpty(); /* returns an error key */
    return stack[top--];
}
```

Program 3.2: Delete from a stack

```
void stackFull()
{
    fprintf(stderr, "Stack is full, cannot add element");
    exit(EXIT_FAILURE);
}
```

Program 3.3: Stack full

[구현조건]

- ① 사용자입력으로부터 데이터 추출을 위해 gets, strtok, strcmp, sscanf, strlen 등을 사용
 - ② push, pop 함수는 교재 코드를 수정 없이 그대로 사용할 것
 - ③ stackFull은 코드를 수정하되, pop()을 사용하여 현재 스택요소들을 출력해 준 후 exit(EXIT_FAILURE)을 호출하도록 구현할 것
 - ④ stackEmpty는 새로 정의하되, 반환형은 임의의 에러키를 가지는 element형이어야 함
- ※ EXIT_FAILURE를 사용하기 위해 stdlib.h를 인클루드해야 함
- ⑤ 잘못된 커맨드를 입력하면 에러 메시지를 출력 후 다시 사용자 입력을 받도록 함

[실행 예]

```

C:\Windows\system32\cmd.exe
<< stack operations where MAX_STACK_SIZE is 5>>
push 1 Jung
pop
*****
push 1 Jung
push 2 Hong
pop
push 3 Kim
push 4 song
pop
push 5 Lee
push 6 Min
push 7 Oh
push 8 Seo
stack is full, cannot add element
current stack elements :
7 Oh
6 Min
5 Lee
4 Kim
3 Jung
계속하려면 아무 키나 누르십시오 . . .

C:\Windows\system32\cmd.exe
<< stack operations where MAX_STACK_SIZE is 5>>
push 1 Jung
push 2 Hong
pop
*****
push 1 Jung
push 2 Hong
add
wrong command! try again!
delete
wrong command! try again!
pop
pop
pop
stack is empty, cannot delete element.
계속하려면 아무 키나 누르십시오 . . .

```

2. [정적할당배열을 이용한 선형큐] 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 queueFull은 아래와 같이 정의함
 - ✓ front == -1, rear == MAX_QUEUE_SIZE -1의 경우 아래와 같이 구현
 - “Queue is full, cannot add element!” 메시지를 출력
 - deleteq를 호출하여 현재 큐내용을 출력

exit(EXIT_FAILURE) 호출

✓ 그 이외에 대해서는 큐의 항목들을 이동

④ queueEmpty는 새로 정의하되, 반환형은 임의의 에러키를 가지는 element형이어야 함

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 . . .
```

■ 제출 형식

- 솔루션 이름 : DS 07

- 프로젝트 이름 : 1, 2

- 각 소스파일에 주석처리

“학번 이름”

“본인은 이 소스파일을 복사 없이 직접 작성하였습니다.”

- 솔루션 정리 후 솔루션 폴더를 “학번.zip”으로 압축하여 과제 게시판에 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!

- 마감 : 다음 수업일 전 자정