

자료구조응용

05. 스택과 큐

스트링 처리 프로그램 예제

```
gets_s(input, sizeof(input)) // 한 라인을 입력 받음
tok = strtok_s(input, delimiter, &next_str);
    input에서 구분자를 참조하여 첫 번째 단어는 tok 가 가르키고,
    나머지 부분은 next_str이 가르킴
```

예

```
char input[80] = "Korea,is very,wonderful country";
char *tok, *next_str;
tok = strtok_s(input, " ", &next_str);
printf("tok = %s\nnext_str = %s\n", tok, next_str);
```

수행 결과는

```
tok = Korea
next_str = is very, wonderful country
```

```
strcmp(tok, "push")
두 개의 스트링을 비교함
```

```
sscanf_s(input + strlen(tok) + 1, "%s%s", st1, sizeof(st1), st2,
sizeof(st2));
input + strlen(tok) + 1는 문자열을 읽어올 위치임.
```

```
char st1[80], st2[80];
char input[80] = "Korea,is very,wonderful country";
char *tok, *next_str;
tok = strtok_s(input, " ", &next_str);
printf("tok = %s\nnext_str = %s\n", tok, next_str);
sscanf_s(input + strlen(tok) + 1, "%s%s", st1, sizeof(st1), st2, sizeof(st2));
printf("st1 = %s\nst2 = %s\n", st1, st2);
```

수행 결과

```
tok = Korea
next_str = is very,wonderful country
st1 = is
st2 = very,wonderful
```

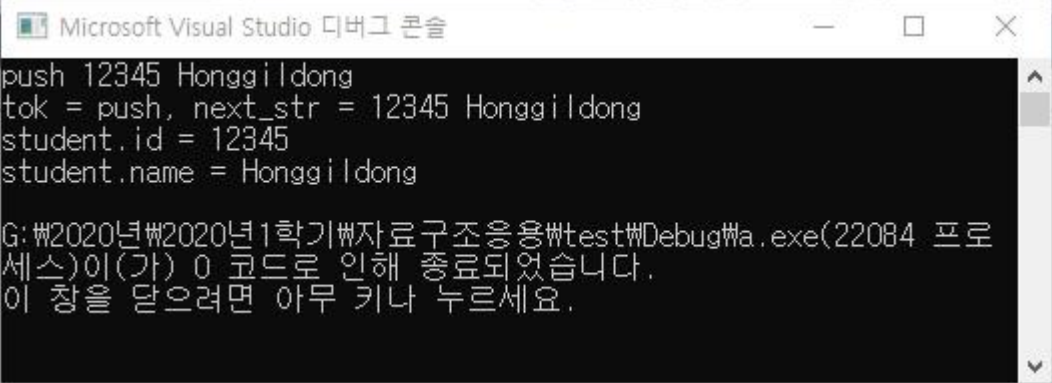
위의 함수를 사용하여 예제 프로그램

입력문자열이 “push 12345 Kingildong” 일 경우

첫 단어가 psuh이면 학생정보를 출력

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_NAME_SIZE 20
typedef struct {
    int id;
    char name[MAX_NAME_SIZE]; // unique id
}element;
void main()
{
    element student;
    char input[80];
    char *delimiter = " \n"; // input a line of string
    char *next_str;
    char *tok = NULL;

    gets_s(input, sizeof(input));
    tok = strtok_s(input, delimiter, &next_str);
    printf("tok = %s, next_str = %s\n", tok, next_str);
    if (!strcmp(tok, "push"))
    {
        sscanf_s(input + strlen(tok) + 1, "%d%s", &student.id, student.name,
sizeof(student.name));
        printf("student.id = %s\n", student.id);
        printf("student.name = %s\n", student.name);
    }
    else printf("not push");
}
```



```
Microsoft Visual Studio 디버그 콘솔

push 12345 Honggildong
tok = push, next_str = 12345 Honggildong
student.id = 12345
student.name = Honggildong

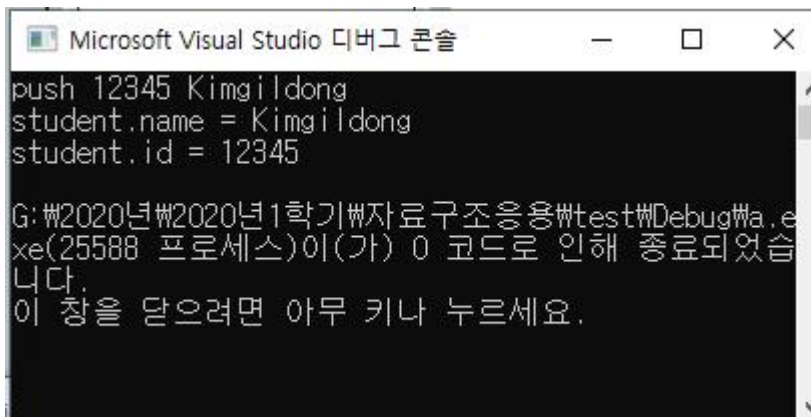
G:\2020년\2020년1학기\자료구조\응용\test\Debug\test.exe(22084 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

scanf_s(), strcmp() 함수만 사용해서 처리

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_NAME_SIZE 20
typedef struct {
    int id;
    char name[MAX_NAME_SIZE]; // unique id
} element;
void main()
{
    element student;
    char tok[10];

    scanf_s("%s%d%s", tok, sizeof(tok), &student.id, student.name,
sizeof(student.name));

    if (!strcmp(tok, "push")) {
        printf("student.name = %s\n", student.name);
        printf("student.id = %d\n", student.id);
    }
    else printf("not push");
}
```



The screenshot shows the 'Microsoft Visual Studio 디버그 콘솔' (Debug Console) window. The first two lines of output are 'push 12345 Kimgildong' and 'student.name = Kimgildong', followed by 'student.id = 12345'. Below these, a system error message is displayed in Korean: 'G:\2020년\2020년1학기\자료구조\응용\test\Debug\wa.exe(25588 프로세스)이(가) 0 코드로 인해 종료되었습니다.' (The process wa.exe(25588) has been terminated by code 0.) and '이 창을 닫으려면 아무 키나 누르세요.' (Press any key to close this window.).

1. [정적할당 배열의 스택] 다음과 같은 스택을 생성하고 실행 예와 같이 수행되는 프로그램을 작성하라. 이를 위해, push, pop, stackEmpty, stackFull, sprint(스택의 내용을 출력) 함수를 구현하여야 한다. push 할 때 스택이 full이면 스택 안의 내용을 출력하고 종료, pop 할 때 스택이 empty이면 오류 메시지를 출력하고 종료함.

[스택 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];
```

사용자 명령

push id name

pop

sprint // 스택내용 출력

[실행 예]

```
<< 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
3 Kim
1 Jung
계속하려면 아무 키나 누르십시오 . . .
```

```
<< stack operations where MAX_STACK_SIZE is 5>>
The format of the stack operation is as follows!!
push 1 Jung
pop
*****
push 1 kim
push 2 park
sprint
2 park
1 kim
opo
wrong command! try again!
pop
pop
pop
stack is empty, cannot delete element.
계속하려면 아무 키나 누르십시오 . . .
```

2. [정적할당 배열을 이용한 선형큐(linear queue)] 다음과 같은 선형 큐를 생성하고, 실행 예와 같이 수행되는 프로그램을 작성하라. 이를 위해, addq, deleteq, queueFull, queueEmpty qprint(queue의 내용을 출력) 함수를 구현하여야 한다. 큐가 full이면 큐가 full 메시지를 출력하고, 큐안의 데이터를 이동해야 할 경우 “data shift”메세지를 출력해야 함. 그리고 empty일 때 데이터를 삭제하면 Queue empty 메시지를 출력하고 종료함.

[자료형과 함수의 정의]

```
#define MAX_QUEUE_SIZE 3
```

```
typedef struct {
    int id;                // unique id
    char name[MAX_NAME_SIZE]; //last name
} element;
element queue[MAX_QUEUE_SIZE];
```

사용자 명령

```
add id name    // queue에 데이터 삽입
delete        // queue에 데이터 삭제
qprint        // queue의 내용을 프린트(먼저 들어온 순서대로)
```

[실행 예]

```
선택 Microsoft Visual Studio 디버그 콘솔
<< Circular queue operations where MAX_QUEUE_SIZE is 3>>
add 1 Jung
delete

*****
add 1 kim
add 2 park
add 3 kweon
add 4 gang
queue is full, not added
delete
qprint
2 park
3 kweon

add 5 na
data shift
qprint
2 park
3 kweon
5 na

delete
add 6 ma
data shift
qprint
3 kweon
5 na
6 ma

delete
delete
delete
delete
queue is empty, cannot delete element.

G:\2020년\2020년1학기\자료구조\응용\DSA-05\64\Debug\2.exe(25400 프로세스)이(
가) 1 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

3. [환형큐(circular queue)] 2번 문제의 프로그램을 환형큐 프로그램으로 수정하라. add, delete, qprint 명령어를 구현하시오.

[자료형과 함수의 정의]

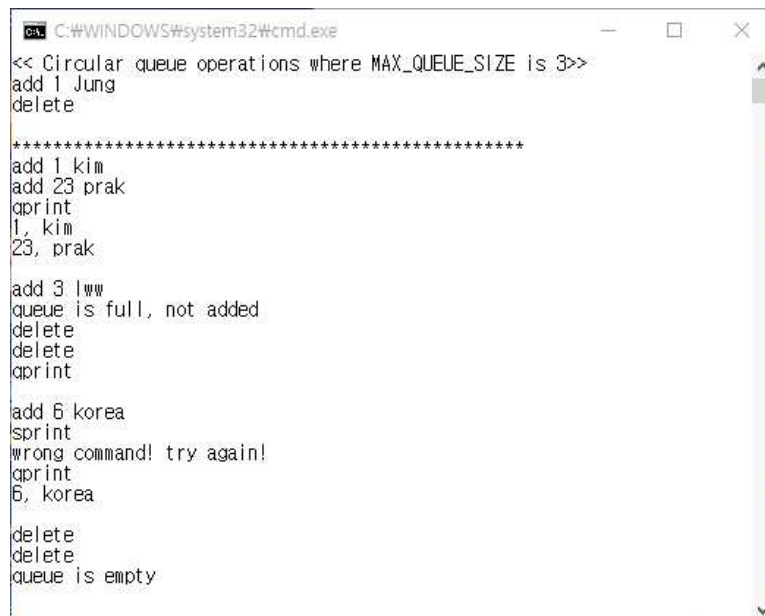
```
#define MAX_QUEUE_SIZE 3
#define MAX_NAME_SIZE 20
```

```
typedef struct {
    int id;                // unique id
    char name[MAX_NAME_SIZE]; // last name
}element;
element queue[MAX_QUEUE_SIZE];
int rear, front;
```

[구현 조건]

- ① 사용자입력으로부터 데이터 추출을 위해 gets_s, strtok_s, strcmp, sscanf_s, strlen 등을 사용
- ② 전역변수 front, rear의 초기값은 각각 0, 0
- ③ MAX_QUEUE_SIZE 3

[실행 예]



```
C:\WINDOWS\system32\cmd.exe
<< Circular queue operations where MAX_QUEUE_SIZE is 3>>
add 1 Jung
delete

*****
add 1 kim
add 23 prak
qprint
1, kim
23, prak

add 3 lww
queue is full, not added
delete
delete
qprint

add 6 korea
sprint
wrong command! try again!
qprint
6, korea

delete
delete
queue is empty
```

■ 제출 형식

- 솔루션 이름 : DS_05
- 프로젝트 이름 : 1, 2, 3
- 솔루션 폴더를 압축하여 제출할 것.
- 학습관리시스템에 과제를 올릴 때 제목:
1차 제출: 학번_이름_DS_05(1), 2차 제출: 학번_이름_DS_05(2)
제출은 2회걸쳐 가능(수정 시간 기준으로 처리)