

스트링 처리 프로그램 예제

```
gets_s(input, sizeof(input)) // 한 라인을 입력 받음
char *gets_s(char *buffer, size_t sizeInCharacters);
매개 변수 : 버퍼 (입력 문자열에 대한 스토리지 위치)
            sizeInCharacters (버퍼의 크기)
```

반환 값 : 성공 하면 버퍼 를 반환 합니다. NULL 포인터는 오류 또는 파일
끝 조건을 나타냅니다.

```
// crt_gets_s.c
// This program retrieves a string from the stdin and
// prints the same string to the console.
```

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

```
int main( void )
{
    char line[21]; // room for 20 chars + '\0'
    gets_s( line, 20 );
    printf( "The line entered was: %s\n", line );
}
```

strtok_s() : 구분자를 이용해서 문장을 단어로 나눈다.

char* strtok_s(char* str, const char* delimiters, char** context);

매개 변수 : str (찾을 토큰을 포함 하는 문자열)

구분 기호 (사용할 구분 기호 문자 집합)

context (함수 호출 간의 위치 정보를 저장 하는 데 사용)

반환 값 : Str 에서 찾은 다음 토큰에 대 한 포인터를

반환. 더 이상 토큰을 찾을 수 없는 경우 NULL 을 반환

tok = strtok_s(input, delimiter, &next_str);

input에서 구분자를 기준으로 나눈 다음, 첫 번째 단어는 tok 가 가르키고, 나머지 부분은 next_str이 가르킴. 구분자는 출력 안함.

delimiter는 스트링으로 주어야 하며, “ ,1\\n” 으로 주어진다면,

공백, ‘,’ , ‘\\n’ 는 모두 구분자임.

strtok_s (반환 값 : 처음 delimiters의 문자가 있는 위치를 반환, 없을 때 NULL 반환

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() : 두 개의 스트링을 비교

```
int strcmp(const char *string1, const char *string2 );
```

매개 변수 : string1, string2 (비교할 Null 종료 문자열).

반환 값

값 문자열 1과 문자열 2의 관계

< 0 string1 이 문자열 2 보다 작음

0 string1 은 문자열 2 와 동일 합니다.

> 0 string1 이 문자열 2 보다 큼니다.

```
strcmp(tok, "push")
```

두 개의 스트링을 비교함

sscanf_s() : 스트링에서 원하는 만큼만 문자열을 가져옴.

int sscanf_s(const char *buffer, const char *format [, argument].);

매개 변수 : 버퍼 (저장된 데이터)

format : 형식 컨트롤 문자열.

argument : 선택적 인수

반환 값

이러한 각 함수는 모두 성공적으로 변환되고 할당된 필드 수를 반환,
이때 읽혀졌지만 할당되지 않은 필드는 반환 값에 포함되지 않는다.
반환 값이 0이면 할당된 필드가 없음을 나타냅니다.

```
sscanf_s(input + strlen(tok) + 1, "%s %s", st1, sizeof(st1), st2,  
sizeof(st2));
```

input + strlen(tok) + 1는 문자열을 읽어올 위치임.

첫 번째 스트링을 읽어서 st1에 복사, 두 번째 스트링을 읽어 st에 복사.

```
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

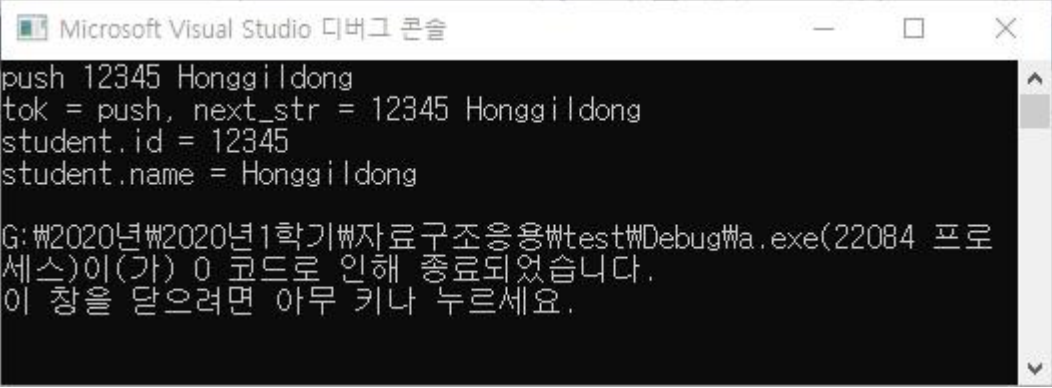
st2 = very,wonderful

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 = " "; // input a line of string
    char *next_str; // delimiter for tokenizing
    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

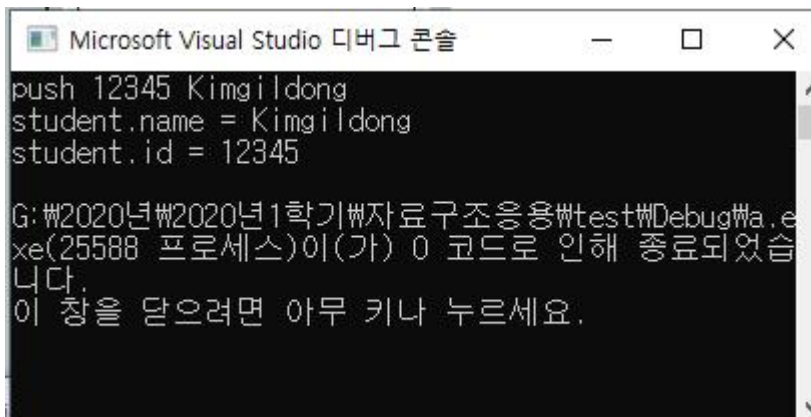
G:\2020년\2020년1학기\자료구조\응용\test\Debug\Wa.exe(22084 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

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");
}
```



```
Microsoft Visual Studio 디버그 콘솔
push 12345 Kimgildong
student.name = Kimgildong
student.id = 12345

G:\2020년\2020년1학기\자료구조\응용\test\Debug\test.exe(25588 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

fgets() : 파일에서 한 라인씩 읽어옴.

```
char *fgets (char *string, int num, FILE *fp);
```

매개 변수 : string (데이터의 스토리지 위치)
 num (읽을 최대 문자 수)
 fp (FILE 구조체에 대한 포인터)

반환 값 :
 함수는 char * 을 반환 함. 오류 또는 파일 끝 조건을 나타내기 위해 NULL 이 반환 됨.

fgets() 함수는 현재 fp 위치에서 어느 것이 먼저 오건 첫 번째 줄 바꾸기 문자(\n)까지, 스트림의 끝까지 또는 읽은 문자 수가 n-1과 같을 때까지 문자를 읽음.

fgets() 함수는 결과를 string에 저장하고 스트링 끝에 널(null) 문자(\0)를 추가합니다.

string은 줄 바꾸기 문자를 포함합니다(읽은 경우). n이 1이면 string이 비어 있습니다.

fgets() 함수는 성공하면 string 버퍼를 가리키는 포인터를 리턴합니다. NULL 리턴값은 오류 또는 파일의 끝 조건을 표시합니다.

아래 예는 파일에서 입력 행을 가져옵니다. 이 예는 스트림에서 MAX_LEN - 1개 문자만 또는 줄 바꾸기 문자까지 읽습니다.

```
#include <stdio.h>
#define MAX_LEN 100

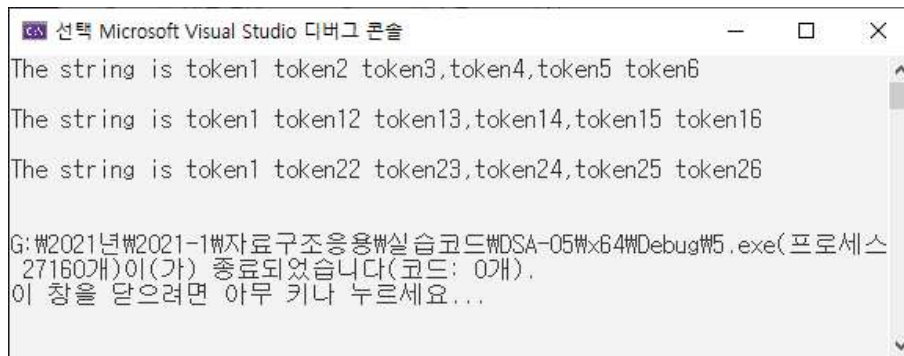
int main(void)
{
    FILE* fp;
    char line[MAX_LEN], * result;

    fopen_s(&fp, "input.txt", "r");
    while (1)
        if ((result = fgets(line, MAX_LEN, fp)))
            printf("The string is %s\n", result);
        else
            break;
    if (fclose(fp))
        fprintf_s(stderr, "fclose error");
}
```

input.txt

```
token1 token2 token3,token4,token5 token6  
token1 token12 token13,token14,token15 token16  
token1 token22 token23,token24,token25 token26
```

수행 결과



```
선택 Microsoft Visual Studio 디버그 콘솔  
The string is token1 token2 token3,token4,token5 token6  
The string is token1 token12 token13,token14,token15 token16  
The string is token1 token22 token23,token24,token25 token26  
  
G:\2021년\2021-1\자료구조응용\실습코드\DSA-05\64\Debug\5.exe (프로세스  
27160개)이(가) 종료되었습니다(코드: 0개).  
이 창을 닫으려면 아무 키나 누르세요...
```


fgets() and strtok_s()

```
// fgets()
#include <stdio.h>
#include <string.h>
#define MAXI 30
int main()
{
    FILE *fp;
    char buf[MAXI];
    char *tok, *ntok;

    fopen_s(&fp, "input.txt", "r");
    fgets(buf, sizeof(buf), fp);
    while (1)
    {
        printf("string is: %s\n", buf);
        ntok = buf;
        while (1) {
            tok = strtok_s(ntok, " \n", &ntok);
            if (tok == NULL) {
                printf("end of line \n");
                break;
            }
            printf("tok = %s\n", tok);
        }

        if(!fgets(buf, sizeof(buf), fp))
            break;
    }
    fclose(fp);
    return 0;
}
```

실행 결과

```
선택 Microsoft Visual Studio 디버그 콘솔
string is: token1 token2 token3,token4,token5 token6
tok = token1
tok = token2
tok = token3
tok = token4
tok = token5
tok = token6
end of line

string is: token1 token12 token13,token14,token15 token16
tok = token1
tok = token12
tok = token13
tok = token14
tok = token15
tok = token16
end of line

string is: token1 token22 token23,token24,token25 token26
tok = token1
tok = token22
tok = token23
tok = token24
tok = token25
tok = token26
end of line

G:\₩2021년₩2021-1₩자료구조용₩실습코드₩DSA-05₩x64₩Debug₩4.exe( 프로세스 24828개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```