

문제해결기법 7주차 과제

202033762 장민호

-Code

```
#include <stdio.h>
#include <string.h>
#define MAX 200
#define SUCCESS 0
#define FAIL 1
```

```
/*
```

```
202033762 장민호 7주차 과제
```

```
문제:
```

문자로 이루어진 String이 하나 주어진다. 이를 main_str라고 하자. 그리고 그 String에서 찾고자 하는 문자열도 주어진다. 이를 search_str이라고 하자. 목표는 main_str에서 search_str이 존재하는지 여부를 파악하는 것이다. search_str은 문자열 중간에 *가 들어갈 수도 있는데, *는 wildcard이다. main_str에서 search_str이 존재하는지 여부를 확인하자.

논리 전개:

1. 두 문자열을 입력받고, 저장한다.

2. search_str에 *가 있는지 파악한다. -> strchr(main_str, '*');

3-1. 만약 *이 존재하지 않을 경우

strstr(main_str, search_str) != NULL 인 경우 문자열이 존재한다는 뜻이다.

3-2. 만약 *이 존재할 경우

*을 기준으로, 앞에 있는 문자열을 first, 뒤에 있는 문자열을 last라고 부르자. 먼저,

strstr(main_str, first)를 통해 first문자열을 찾는다.

4-1. 만약 first가 없을 경우

찾으려는 문자열이 존재하지 않으므로 FALSE를 리턴한다.

4-2. 만약 first가 존재할 경우

first가 존재하는 위치를 저장해 놓는다. 그 위치+first 문자열의 길이 이후로부터,

다시 strstr(main_str, last)을 한다.

5-1. 만약 last가 존재할 경우

해당 문자열이 존재하므로 True를 리턴한다.

5-2. last가 존재하지 않을 경우

False를 리턴한다.

```
*/
```

```
int searchMain(char *main_str, char *search_str)
```

```
{
```

```
// 2. search_str에 *가 있는지 파악한다. -> strchr(main_str, '*');
```

```
if (strchr(search_str, '*') == NULL)
```

```
{
```

```
// *이 없으므로, 그냥 서치하면 된다.
```

```
if (strstr(main_str, search_str) != NULL)
```

```
{
```

```
return SUCCESS;
```

```
}
```

```
return FAIL;
```

```
}
```

```
else
```

```
{
```

```
// *을 기준으로, 앞에 있는 문자열을 first, 뒤에 있는 문자열을 last라고 부르자.
```

```
char *first = strtok(search_str, "**");
```

```
char *last = strtok(NULL, "**");
```

```
char *location = strstr(main_str, first);
```

```
int len = strlen(first);
```

```
// 먼저, strstr(main_str, first)를 통해 first문자열을 찾는다.
```

```
if (location == NULL)
```

```
{
```

```
return FAIL;
```

```
}
```

```
else
```

```

    {
        // first 문자열을 찾았으므로, 찾은 위치로부터 first문자열 길이를 더한 뒤 그 뒤부터 last
를 찾으면 된다.
        location += len;
        if (strstr(location, last) != NULL)
        {
            return SUCCESS;
        }
        return FAIL;
    }
}
return SUCCESS;
}
int main()
{
    char main_str[MAX];
    char search_str[MAX];
    for (int i = 0; i < 5; i++)
    {
        printf("=====\n");
        printf("example %d\n", i + 1);
        gets(main_str);
        gets(search_str);
        if (searchMain(main_str, search_str) == SUCCESS)
        {
            printf("\nmatch found\n");
        }
        else
        {
            printf("\nmatch not found\n");
        }
    }
    return 0;
}
/*
A thief named hhhong gil dong lived with friends named hong gil don and hong gil ja in a village
named hong gil dong village.
hong gil dong
sample data
sample* data
sample
sa*a
sampple
a*pple
sssaaammpppllllee
s*lle
*/

```

-Screenshot

```
mhj@mhj-IdeaPad:~/gitRepo/2022_second-semester/Problem solving techniques/week7$ ./"
hong"
=====
example 1
A thief named hhhong gil dong lived with friends named hong gil don and hong gil ja
in a village named hong gil dong village.
hong gil dong

match found
=====
example 2
sample data
sample* data

match found
=====
example 3
sample
sa*a

match not found
=====
example 4
sample
a*pple

match found
=====
example 5
sssaaammppplleee
s*lle

match found
```