

SW 프로그래밍(2)

implemetation & evaluation

20175528 이민지

목차

1. 파일 내에서 해당 단어 찾기

2. 폴더 내의 파일 열기

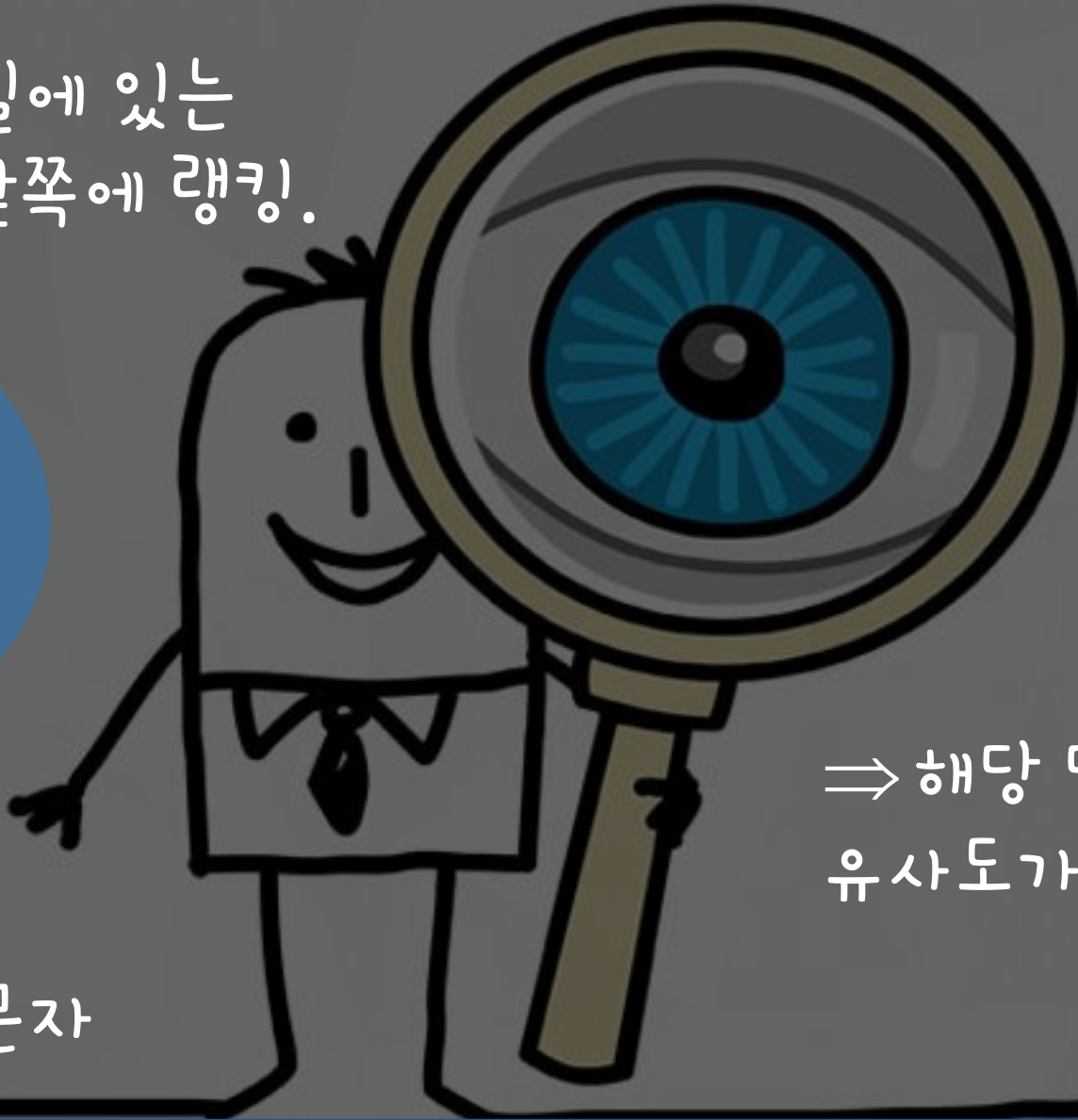
3. 출력하기

<파일 내에서 해당 단어 찾기>

⇒ 해당 단어가 파일에 있는
횟수가 높을 수록 앞쪽에 랭킹.

일치하는
단어가
있다면?

1. 모두 소문자
2. 모두 대문자
3. 첫글자만 대문자



일치하는
단어가
없다면?

⇒ 해당 단어와 파일의 단어
유사도가 높을수록 앞쪽에 랭킹.


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

int main()
{
    char str[30], word[30], b1[30], b2[30];
    int count = 0, size, i;

    printf("원하는 단어를 입력하시오. : ");
    scanf("%s", str);

    size = strlen(str);
```

-> 원하는 단어 입력 후 단어의 size 측정

```
if (96 < str[0] && str[0] < 123)
{
    strcpy(b1, str);
    b1[0] = str[0] - 32;
    strcpy(b2, str);
    strupr(b2);
}
```

-> 단어가 소문자로 시작하는 경우
 새로운 문자열 b1 = 모두 대문자인 경우
 새로운 문자열 b2 = 첫글자만 대문자인 경우.

```
else if (64 < str[0] && str[0] < 91)
{
    if (64 < str[1] && str[1] < 91)
    {
        strcpy(b1, str);
        strlwr(b1);
        b1[0] = b1[0] - 32;
        strcpy(b2, str);
        strlwr(b2);
    }
    else if (96 < str[1] && str[1] < 123)
    {
        strcpy(b1, str);
        b1[0] = str[0] + 32;
        strcpy(b2, str);
        strupr(b2);
    }
}
```

-> 단어가 대문자로 시작하는 경우
 1. 두번째 단어도 대문자다=그 단어는 모두 대문자다.
 새로운 문자열 b1 = 첫 글자만 대문자인 경우.
 새로운 문자열 b2 = 모두 소문자인 경우
 2. 두번째 단어가 소문자다=첫 글자만 대문자다.
 새로운 문자열 b1 = 모두 소문자인 경우.
 새로운 문자열 b2 = 모두 대문자인 경우.

```
FILE *fp = fopen("2537newsML.txt", "rt");

if (fp == NULL)
{
    printf("이 파일은 비어있습니다!");
    return 1;
}
do
{
    fscanf(fp, "%s", word);
    if (strlen(word) == size)
    {
        if (strstr(word, str) != NULL)
            count++;
        else if (strstr(word, b1) != NULL)
            count++;
        else if (strstr(word, b2) != NULL)
            count++;
    }
    else
        continue;
} while (!feof(fp));
```

```
printf("단어의 개수 : %d \n", count);
fclose(fp);
return 0;
}
```

-> 제대로 개수를 세는지 확인 하기 위해
Count값을 출력해보았다.

-> TEST용 FILE을 연다.
그리고 파일을 통해 입력 받은 단어가
사용자가 입력한 단어와 크기가 같다면
strstr을 통한 비교를 시작한다.
같다면 count를 한다.

“EXAMPLE”

test - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

hello Hello hello HELLO Hello hi bye hello

C:\WINDOWS\system32\cmd.exe

원하는 단어를 입력하십시오. : HELLO
단어의 개수 : 6
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe

원하는 단어를 입력하십시오. : Hello
단어의 개수 : 6
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe

원하는 단어를 입력하십시오. : hello
단어의 개수 : 6
계속하려면 아무 키나 누르십시오 . . .

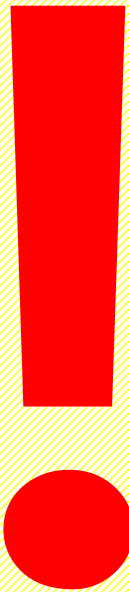
“EXAMPLE”

2537newsML - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

A break-in at the U.S. Justice Department's World Wide Web site last week highlighted Unidentified hackers gained access to the department's web page on August 16 and reported Justice officials quickly pulled the plug on the vandalised page, but the security of "The vast majority of sites are vulnerable," said Richard Power, senior analyst at the Justice Department officials said the compromised web site was not connected to any. The security breach "is just like graffiti on the outside of the building," spokesman. Other organisations have been targeted in the past. Last year, the Nation of Islam's Windows Magazine recently found security flaws at web sites of a dozen major corporations. Elgan said hackers who are exploiting some of the same flaws are motivated by anger. The battle is not completely hopeless. "You can secure a web site," Richard Power said. Some sites are using multiple layers of security, well beyond simple password protection. One site mentioned by Windows Magazine was Fidelity Investments. Fidelity's site advised Fidelity officials immediately closed the loophole identified by the magazine, a spokesman

C:\WINDOWS\system32\cmd.exe

원하는 단어를 입력하십시오. : at
단어의 개수 : 4
계속하려면 아무 키나 누르십시오 . . .



단어가 만약에 알파벳이 아닌것으로 시작하면 어떨까요?

Ex. "The princess ~"

단어의 끝이 알파벳이 아니면 크기가 다를텐데 어떨까요?

Ex. Hello!


```

void frontchecker(char str[30])
{
    int size, i;
    char front;

    size = strlen(str);

    front = str[0];

    switch (front)
    {
        case '(':
            for (i = 0; i < size - 1; i++)
                str[i] = str[i + 1];
            str[size-1] = '\0';
            break;
        case '\\\"':
            for (i = 0; i < size - 1; i++)
                str[i] = str[i + 1];
            str[size-1] = '\0';
            break;
        case '\\\'':
            for (i = 0; i < size - 1; i++)
                str[i] = str[i + 1];
            str[size-1] = '\0';
            break;
        default:
            break;
    }
}

```

```

void lastchecker(char st[30])
{
    int size; //st의 크기
    char last;

    size = strlen(st);

    last = st[size-1];

    switch (last)
    {
        case ',':
            st[size-1] = '\0';
            break;
        case '.':
            st[size-1] = '\0';
            break;
        case ')':
            st[size-1] = '\0';
            break;
        case '!':
            st[size-1] = '\0';
            break;
        case '~':
            st[size-1] = '\0';
            break;
        case '?':
            st[size-1] = '\0';
            break;
    }
}

```

-> 단어의 시작
(“ ‘

-> 단어의 끝
, .) ! ~ ? “ ‘

```

        case '\\\"':
            st[size-1] = '\0';
            break;
        case '\\\'':
            st[size-1] = '\0';
            break;
        default:
            break;
    }
}

```

```
FILE *fp = fopen("marktest2.txt", "rt");

if (fp == NULL)
{
    printf("이 파일은 비어있습니다!");
    return 1;
}
do
{
    fscanf(fp, "%s", word);
    frontchecker(word);
    lastchecker(word);

    if (strlen(word) == size)
    {
        if (strstr(word, str) != NULL)
            count++;
        else if (strstr(word, b1) != NULL)
            count++;
        else if (strstr(word, b2) != NULL)
            count++;
    }
    else
        continue;
} while (!feof(fp));

printf("단어의 개수 : %d \n", count);
fclose(fp);
```

“EXAMPLE”

marktest - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

"hello" HELLO nice to meet "Hello "oh 'hi' Hello hello. Hello!

C:\> C:₩INDOWSWsystem32₩cmd.exe

원하는 단어를 입력하십시오. : hello
단어의 개수 : 6
계속하려면 아무 키나 누르십시오 . . .

C:\> C:₩INDOWSWsystem32₩cmd.exe

원하는 단어를 입력하십시오. : Hello
단어의 개수 : 6
계속하려면 아무 키나 누르십시오 . . .

C:\> C:₩INDOWSWsystem32₩cmd.exe

원하는 단어를 입력하십시오. : HELLO
단어의 개수 : 6
계속하려면 아무 키나 누르십시오 . . .

“EXAMPLE”

marktest2 - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

bello! bye. "bello" hello Bello HI BELL0 (BELL0) bello~

C:\WINDOWS\system32\cmd.exe

원하는 단어를 입력하십시오. : BELL0
단어의 개수 : 6
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe

원하는 단어를 입력하십시오. : Bello
단어의 개수 : 6
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe

원하는 단어를 입력하십시오. : bello
단어의 개수 : 6
계속하려면 아무 키나 누르십시오 . . .

<검색 가능한 단어의 종류>

1. 가능한 단어의 시작
(“ ‘
2. 가능한 단어의 끝
, .) ! ~ ? “ ‘
3. 복수형, 과거형 동사일 경우 100% 일치하는 단어가 없는 경우가 되어 유사도 측정으로 넘어가게 된다.



back

```

if (count == 0)
{
    fseek(fp, 0L, SEEK_SET);
    do
    {
        fscanf(fp, "%s", word);
        frontchecker(word);
        lastchecker(word);

        if (size < strlen(word))
        {
            if (strstr(word, str) != NULL)
                include++;
            else if (strstr(word, b1) != NULL)
                include++;
            else if (strstr(word, b2) != NULL)
                include++;
        }
        else
            continue;
    } while (!feof(fp));
}

```

-> 파일의 단어가 사용자가 입력한 단어보다 길 때,
파일의 단어 속에 그 단어가 있는지 파악하기 위해
strstr을 사용한다.

-> 포함되지 않는다면, 파일에서 입력받은 단어의
앞 부분부터 유사도가 높을수록 값을 높게 매긴다.

```

if (include == 0)
{
    fseek(fp, 0L, SEEK_SET);
    do
    {
        fscanf(fp, "%s", word);
        frontchecker(word);
        lastchecker(word);

        if (str[0] == word[0])
        {
            if (str[1] == word[1])
            {
                noword = 2;
                for (i = 2; i < size-1; i++)
                {
                    if (str[i] == word[i])
                        noword++;
                }
                if (noword > nowordmax)
                {
                    nowordmax = noword;
                    nowordmaxc = 0;
                }
                if (noword == nowordmax)
                {
                    nowordmaxc++; /
                }
            }
        }
    } while (!feof(fp));
}

```



```
printf("단어를 포함하는 경우 : %d \n", include);  
if (include == 0)  
{  
    printf("앞부분이 단어와 비슷한 경우 : %d / %d번", nowordmax, nowordmaxc);  
}
```

“EXAMPLE”

back

hand - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
handbook foot eye nose handcuff handful secondhand free finger handmade backhand

C:\WINDOWS\system32\cmd.exe
원하는 단어를 입력하시오. : hand
단어의 개수 : 0
단어를 포함하는 경우 : 6
계속하려면 아무 키나 누르십시오 . . .

C:\WINDOWS\system32\cmd.exe
원하는 단어를 입력하시오. : hard
단어의 개수 : 0
단어를 포함하는 경우 : 0
앞부분이 단어와 비슷한 경우 : 2 / 4번계속

C:\WINDOWS\system32\cmd.exe
원하는 단어를 입력하시오. : hangul
단어의 개수 : 0
단어를 포함하는 경우 : 0
앞부분이 단어와 비슷한 경우 : 3 / 4번계속하

<폴더 내의 파일 열기>

해당 폴더내의 파일 목록을 읽어 오기

=> 헤더파일 io.h의 함수 findfirst, findnext, findclose를 이용.

차례로 파일을 열며 단어의 유무 파악!




```

#include <stdio.h>
#include <io.h>
#include <conio.h>

void main()
{
    struct _finddata_t fd;
    long handle;
    int result = 1;
    int count = 0;

    handle = _findfirst("c:\\dataset\\dataset\\*.*", &fd);
    if (handle == -1) return;
    while (result != -1)
    {
        printf("파일명 : %s, 크기:%d\n", fd.name, fd.size);
        count++;
        result = _findnext(handle, &fd);
    }
    _findclose(handle);

    printf("\n총 개수 : %d", count-2);
}

```

-> 제대로 파일 목록을 읽어 왔는지 확인하기 위해 Count를 셸다.
 마지막에 2를 뺀 이유는 현재 디렉토리, 상위 디렉토리를 나타내는 ., ..을 빼기 위해서다.



```

파 : 346831newsML.txt, 크기:3144
파 : 347226newsML.txt, 크기:2512
파 : 348501newsML.txt, 크기:5605
파 : 354020newsML.txt, 크기:2368
파 : 354135newsML.txt, 크기:3702
파 : 354285newsML.txt, 크기:3596
파 : 355589newsML.txt, 크기:2607
파 : 357147newsML.txt, 크기:3544
파 : 358675newsML.txt, 크기:2997
파 : 361792newsML.txt, 크기:2751
파 : 361793newsML.txt, 크기:3620
파 : 361820newsML.txt, 크기:3451
파 : 364638newsML.txt, 크기:4256
파 : 364643newsML.txt, 크기:3143
파 : 366020newsML.txt, 크기:3243
파 : 369570newsML.txt, 크기:3871
파 : 371380newsML.txt, 크기:2925
파 : 372744newsML.txt, 크기:2498
파 : 372989newsML.txt, 크기:3191
파 : 372995newsML.txt, 크기:2935
파 : 377885newsML.txt, 크기:3537
파 : 377897newsML.txt, 크기:3501
파 : 378457newsML.txt, 크기:2398
파 : 383585newsML.txt, 크기:2821
파 : 394237newsML.txt, 크기:4767
파 : 398094newsML.txt, 크기:3121
파 : 401260newsML.txt, 크기:4340
파 : 407599newsML.txt, 크기:3322

```

총 개수 : 500계속하려면 아무 키나 누르십시오 . . .

-> 교수님이 주신 dataset 폴더 내의 파일 목록을 불러오도록 했다.

(목록은 생략)

개수가 500개로 코드가 잘 실행 됨을 알 수 있다.



```
#include <stdio.h>
#include <io.h>
#include <conio.h>
#include <string.h>
```

```
typedef struct name
{
    char name[30];
}bello;
```

```
void main()
{
```

```
    struct _finddata_t fd;
    long handle;
    int result = 1;
    int count = 0;
    char word[30];
    int i;
    char strChangeDir[60] = { "c:testfile" };
```

```
    bello filename[6];
```

```
    handle = _findfirst("c:testfile\\*.*", &fd);
```

```
    if (handle == -1) return;
```

```
    while (result != -1)
```

```
    {
        if ((strcmp(".", fd.name)) != 0 && (strcmp("..", fd.name)) != 0)
        {
            printf("파일명 : %s, 크기:%d\n", fd.name, fd.size);
            strcpy(filename[count].name, fd.name);
            count++;
        }
    }
```

```
    result = _findnext(handle, &fd);
    findclose(handle);
}
```

이전에는 count-2를 해줬지만, 이번에는
파일 명이 ., ..일 경우를 제외시켜서
측정된 count값이 그대로 출력되도록 했다.

fopen의 경로를 바꿔주기 위해
작업 디렉토리를 변경해주었다.

```
printf("\n총 개수 : %d\n", count);
```

```
for (i = 0; i < count; i++)
{
    printf("파일 이름 : %s \n", filename[i].name);
}
```

```
if (chdir(strChangeDir) == 0)
```

```
{
    for (i = 0; i < count; i++)
    {
        printf("\n\n파일[%d]", i+1);
        FILE *fp = fopen(filename[i].name, "rt");
```

```
        if (fp == NULL)
        {
            printf("error to opening file. \n");
            break;
        }
```

```
        while (!feof(fp))
        {
            fscanf(fp, "%s", &word);
            printf("%s ", word);
        }
```

```
        fclose(fp);
```

```
    }
    return 0;
}
```


hand

hello

love

moon

moonU

star

2017-11-28 오후 10:...

텍스트 문서

2017-11-30 오후 4:10

텍스트 문서

<- 디렉토리에 있던 파일 목록

C:\WINDOWS\system32\cmd.exe

```
파일명 : hand.txt, 크기:80
파일명 : hello.txt, 크기:64
파일명 : love.txt, 크기:45
파일명 : moon.txt, 크기:55
파일명 : moonU.txt, 크기:61
파일명 : star.txt, 크기:46
```

총 개수 : 6

```
파일 이름 : hand.txt
파일 이름 : hello.txt
파일 이름 : love.txt
파일 이름 : moon.txt
파일 이름 : moonU.txt
파일 이름 : star.txt
```

출력결과 ->

```
handbook foot eye nose handcuff handful secondhand free finger handmade backhand
hi hello HAND bye "hello" goodbye MOON hello! Hello HELLO! HELLO
love hate cute lovely LOVE like luv LOVE HAND
moonless moonish moony star moonlight highlight mooned mooned
nomoon moonlight star Moooon MOOON hi programming C language language
star Star Hand moon HELLO! city of stars star! 계속하려면 아무 키나 누르십시오 . . .
```

2537newsML

10193newsML

10264newsML

10306newsML

10650newsML

10663newsML

12228newsML

12877newsML

12897newsML

13068newsML

13069newsML

13415newsML

13547newsML

13583newsML

14014newsML

2005-10-09 오후 4:59

텍스트 문서

<- 디렉토리에 있던 파일 목록

2005-10-09 오후 4:59

텍스트 문서

총 개수 : 500

파일명 : 100385newsML.txt

파일명 : 101678newsML.txt

파일명 : 101797newsML.txt

파일명 : 10193newsML.txt

파일명 : 102444newsML.txt

파일명 : 102529newsML.txt

파일명 : 10264newsML.txt

파일명 : 102760newsML.txt

파일명 : 10306newsML.txt

파일명 : 103816newsML.txt

파일명 : 104277newsML.txt

파일명 : 104278newsML.txt

파일명 : 104405newsML.txt

파일명 : 105327newsML.txt

파일명 : 105756newsML.txt

파일명 : 105834newsML.txt

파일명 : 106247newsML.txt

파일명 : 10650newsML.txt

파일명 : 10663newsML.txt

파일명 : 106762newsML.txt

파일명 : 107525newsML.txt

파일명 : 108359newsML.txt

파일명 : 109096newsML.txt

파일명 : 109906newsML.txt

출력결과 ->

```
2 www.Bandicam.co.kr main()
16 int count = 0;
17 char word[30];
18 int i;
19 char strChangeDir[60] = { "c:dataset\\\\" }; //디렉토리 바꿔줄꺼임
20
21 bello filename[500];
22
23 //bello *filename = malloc(sizeof(bello)*count); // 처음에는 임의의 큰 수로 했다가 나중에 realloc으로 사이즈 변경!
24
25 handle = _findfirst("c:dataset\\*", &fd);
26 if (handle == -1) return;
27 while (result != -1)
28 {
29     if ((strcmp(".", fd.name)) != 0 && (strcmp("..", fd.name)) != 0) //얘네는 빼고
30     {
31         printf("파일명 : %s, 크기:%d\n", fd.name, fd.size);
32         strcpy(filename[count].name, fd.name);
33         count++;
34     }
35 }
```

보기 선택(S): 빌드



<출력하기>

필요한 요소들을 구조체에 저장!

```
typedef struct hello
{
    char filename[30]; //해당 파일의 이름
    int yescount; //단어가 있는 경우 몇번 있는지
    int contain; //단어를 몇번 포함하는지
    int coin; //몇 글자가 일치하는지
    int nocount; //일치하는 경우 몇번 있는지
}hello;
```

랭킹 기준에 따른 출력

1. 해당 단어가 파일 내에 있는 경우 -> yescount가 클수록 앞에
2. 해당 단어가 파일 내에 없는 경우
 - 1) contain이 클수록 앞에
 - 2) coin이 클수록(일치하는 문자의 수)
 - 3) coin이 같을 경우 nocount가 클수록(ex. 파일 내에서 일치하는 문자의 수 3개가 최대일때 3개가 일치하는 경우의 수)

코리아정보보안아카데미 - koreaisacademy - koreaisacademy.com

[광고] www.koreaisacademy.com/ ▼ 1588-5890

c언어, 2017교육브랜드대상, 100%국비지원, 취업대비과정, 수강료조회

C(프로그래밍 언어) - 나무위키

[https://namu.wiki/w/C\(프로그래밍%20언어\)](https://namu.wiki/w/C(프로그래밍%20언어)) ▼

2017. 11. 9. - 어플리케이션 영역이 아닌 운영체제를 어셈블리어가 아닌 언어로 작성한다는것은 당시엔 일종의 금기에 가까웠고[6], C 언어와 유닉스는 소수의 ...

C (프로그래밍 언어) - 위키백과, 우리 모두의 백과사전

[https://ko.wikipedia.org/wiki/C_\(프로그래밍_언어\)](https://ko.wikipedia.org/wiki/C_(프로그래밍_언어)) ▼

C는 1972년 켄 톰슨과 데니스 리치가 벨 연구소에서 일할 당시 새로 개발된 유닉스 운영 체제에서 사용하기 위해 개발한 프로그래밍 언어이다. 켄 톰슨은 BCPL언어 ...

역사 · 메모리 관리 · 라이브러리 · 개발 도구

=>추후 가능하다면 여기에 해당 단어를 포함한 문장도 저장해서 위의 예시처럼 출력할 것이다.




```

if (chdir(strChangeDir) == 0)
{
    for (i = 0; i < filecount; i++)
    {
        count = 0;
        include = 0;
        nowordmax = 0;
        nowordmaxc = 0;
        FILE *fp = fopen(score[i].filename, "rt");

        if (fp == NULL)
        {
            printf("이 파일은 비어있습니다!");
            return 1;
        }
        do
        {
            fscanf(fp, "%s", word);
            frontchecker(word);
            lastchecker(word);

            if (strlen(word) == size)
            {
                if (strstr(word, str) != NULL)
                    count++;
                else if (strstr(word, b1) != NULL)
                    count++;
                else if (strstr(word, b2) != NULL)
                    count++;
            }
            else
                continue;
        } while (!feof(fp));

        score[i].yescount = count;
    }
}

```

<단어가 있는 경우>

count를 계산 후 구조체에 그 값을
저장해 준다.

```

if (score[i].yescount == 0)
{
    fseek(fp, 0L, SEEK_SET);
    do
    {
        fscanf(fp, "%s", word);
        frontchecker(word);
        lastchecker(word);

        if (size < strlen(word))
        {
            if (strstr(word, str) != NULL)
                include++;
            else if (strstr(word, b1) != NULL)
                include++;
            else if (strstr(word, b2) != NULL)
                include++;
        }
        else
            continue;
    } while (!feof(fp));
    score[i].contain = include;
}

```

<단어가 없는 경우.>

구조체에 저장된 count의 값이 0일 경우이다.

Strstr을 이용해서 단어가 포함되어 있는지 구하고, 포함된 횟수(include)를 증가시킨다.

그리고 그 뒤에 구조체에 저장한다.

```

if (score[i].contain == 0)
{
    fseek(fp, 0L, SEEK_SET);
    do
    {
        fscanf(fp, "%s", word);
        frontchecker(word);
        lastchecker(word);

        if (str[0] == word[0] || b1[0] == word[0])
        {
            if (str[1] == word[1])
            {
                noword = 2;
                for (num = 2; num < size; num++)
                {
                    if (str[num] == word[num])
                        noword++;
                }
                if (noword > nowordmax)
                {
                    nowordmax = noword;
                    nowordmaxc = 0;
                }
                if (noword == nowordmax)
                {
                    nowordmaxc++;
                }
            }
        }
    }
}

```

```

    } while (!feof(fp));
    score[i].coin = nowordmax;
    score[i].nocount = nowordmaxc;
}
fclose(fp);
}

```

<단어가 없는 경우이자 포함되지 않는 경우.>

구조체에 저장된 contain의 값이 0 일 경우이다.

두번째 글자 이상이 같아야 유사하다고 말 할 수 있다고 생각해서 조건을 설정해 주었다.

nowordmax는 일치하는 글자수
nowordmaxc는 일치하는 글자수가 nowordmax일때 파일에 그 단어가 몇 번 있는지 세는 경우이다.


```

for (i = 0; i < filecount; i++)
{
    if (score[i].yescount > 0)
        printf("파일명 : %s에는 단어가 있습니다!   있는 개수 : %d  \n", score[i].filename, score[i].yescount);
    else if (score[i].yescount == 0)
    {
        if (score[i].contain > 0)
            printf("파일명 : %s에는 단어가 없습니다!, 포함 횟수 : %d  \n", score[i].filename, score[i].contain);
        else if (score[i].contain == 0 && score[i].nocount != 0)
            printf("파일명 : %s에는 단어가 없습니다!, 일치하는 단어 수 : %d, 횟수 : %d  \n", score[i].filename, score[i].coin, score[i].nocount);
        else
            printf("파일명 : %s에는 그 어떤 비슷한 단어도 없습니다!\n", score[i].filename);
    }
}

```

=> 구조체에 제대로 저장되었는지 확인 하기 위한 코드

“EXAMPLE”

hand - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
handbook foot eye nose handcuff handful secondhand free finger handmade backhand

hello - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
hi hello HAND bye "hello" goodbye MOON hello! Hello HELLO! HELLO

love - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
love hate cute lovely LOVE like luv LOVE HAND

moon - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
moonless moonish moony star moonlight highlight mooned

star - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
star Star Hand moon HELLO! city of stars star!

“EXAMPLE”

```
C:\> C:\WINDOWS\system32\cmd.exe
원하는 단어를 입력하십시오. : hand
파일명 : hand.txt, 포함 횟수 : 6
파일명 : hello.txt, 있는 개수 : 1
파일명 : love.txt, 있는 개수 : 1
파일명 : moon.txt에는 그 어떤 비슷한 단어도 없습니다!
파일명 : star.txt, 있는 개수 : 1
계속하려면 아무 키나 누르십시오 . . .
```

```
C:\> C:\WINDOWS\system32\cmd.exe
원하는 단어를 입력하십시오. : moon
파일명 : hand.txt에는 그 어떤 비슷한 단어도 없습니다!
파일명 : hello.txt에는 단어가 있습니다! 있는 개수 : 1
파일명 : love.txt에는 그 어떤 비슷한 단어도 없습니다!
파일명 : moon.txt에는 단어가 없습니다!, 포함 횟수 : 5
파일명 : star.txt에는 단어가 있습니다! 있는 개수 : 1
계속하려면 아무 키나 누르십시오 . . .
```


=>앞으로는 구조체에 넣은 값을 이용하여 앞에서 말한 랭킹 기준에 따라 출력하는 코드를 작성하도록 할 것 입니다.

1순위 count가 큰 경우

2순위 contain이 큰 경우

3순위 nowordmax가 큰 경우

(그 값이 같다면 nowordmax가 클 수록 앞에!)



감사합니다!!