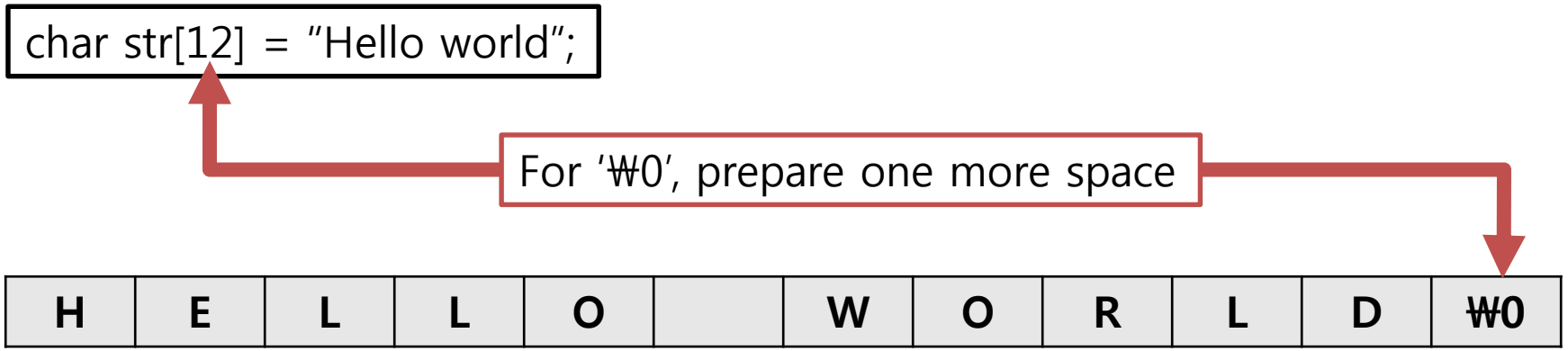
String

2017 first semester
Basis and Practice in Programming

String structure

- String is an array that stores some characters.
- Special mark, '₩0', defines the end of string.



- If you print string as **printf("%s", str)**, program show all characters until it meets NULL(₩0) mark.
- If there is no NULL mark, it read forever. And it may cause unexpected error.

Input functions in C

- **scanf**
  - Read inputs with the fixed format.
  - scanf("%s", str) reads a string until **space**, newline, EOF, NULL mark.
- **getchar**
  - getchar() reads only a character and return it immediately.
- **gets**
  - gets(str) reads a line until newline, EOF or NULL mark.
  - gets reads string including white space.
- **fgets**
  - fgets(str, size, stream) reads a line until newline, EOF or NULL mark.
  - fgets reads a line less than 'size'.
  - fgets reads string including white space.

Practice 1 : Implementing my_gets

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


int my_gets(char *str, int max_length){
    // Implement here
}

int main () {
    char str[50];

    printf("Input string : ");
    printf("# of characters: %d₩n", my_gets(str, 50));
    printf("%s₩n", str);

    return 0;
}
```

```
Input string : I have a pen, I have an apple. Uh! Apple pen.
# of characters: 45
I have a pen, I have an apple. Uh! Apple pen.
계속하려면 아무 키나 누르십시오 . . .
```

- In normal gets function, we cannot bound the length of input.
- Implement my_gets function that can bound the length of input.
- my_gets function returns the number of characters stored.
- Do not use fgets and gets function.
- getchar() is useful.

Practice 2 : palindrome

```c
#include <string.h>

// This function can check if the string is palindrome or not.
// 1. If the string is palindrome, It returns zero.
// 2. If the string is not palindrome, It returns not zero.
int Palindrome(char str[]);

int main()
{
    int i;
    char str[50];

    while (1){
        printf("Input string (exit: 0): ");
        gets(str);
        if (str[0] == '0') return 0;

        if (Palindrome(str) == 0)
            printf("[Palindrome]₩n₩n", str);
        else
            printf("[No Palindrome]₩n₩n", str);
    }
}

int Palindrome(char str[])
{
    // Fill this functions


}
```

- Palindrome is a word which is read the same backward or forward.
- Write a program that tells whether the words are palindrome or not.

```
Input string (exit: 0): madam
[Palindrome]

Input string (exit: 0): palindrome
[No Palindrome]

Input string (exit: 0): level
[Palindrome]

Input string (exit: 0): watermelon
[No Palindrome]

Input string (exit: 0): dennis sinned
[Palindrome]

Input string (exit: 0): 0
계속하려면 아무 키나 누르십시오 . . .
```

Example : string.h library

```c
#include <stdio.h>

int main(){

    char str1[12] = "Hello";
    char str2[12] = "String";
    char str3[12];
    int len;

    // copy string
    strcpy(str3, str1);
    printf("strcpy(str3, str1) : %s\n", str3);

    // compare string
    printf("strcmp(str1, str2) : %d\n", strcmp(str1, str2));
    printf("strcmp(str1, str3) : %d\n", strcmp(str1, str3));

    // get string length
    len = strlen(str1);
    printf("strlen(str1) : %s\n", str1);

    // append string
    strcat(str1, str2);
    printf("strcat(str1, str2) : %s\n", str1);

    len = strlen(str1);
    printf("strlen(str1) : %s\n", str1);


}
```

```
strcpy(str3, str1) : Hello
strcmp(str1, str2) : -1
strcmp(str1, str3) : 0
strlen(str1) : Hello
strcat(str1, str2) : HelloString
strlen(str1) : HelloString
계속하려면 아무 키나 누르십시오 . . .
```

Practice 3 : Alphabetically sorting

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

int main () {
    char words[10][20];

    for (int i = 0; i < 10; i++){
        printf("Enter 10 words (%d/10) : ", i+1);
        scanf("%s", words[i]);
    }

    printf("Before sorting₩n");
    for (int i = 0; i < 10; i++){
        printf("%s₩n", words[i]);
    }

    // Implement here


    printf("After sorting₩n");
    for (int i = 0; i < 10; i++){
        printf("%s₩n", words[i]);
    }


    return 0;
}
```

- Input 10 words. (word length is not over 20)
- Sorting all words alphabetically.
- You can use fixed size 2-D array.

```
Enter 10 words (1/10) : apple
Enter 10 words (2/10) : pen
Enter 10 words (3/10) : you
Enter 10 words (4/10) : sun
Enter 10 words (5/10) : moon
Enter 10 words (6/10) : cat
Enter 10 words (7/10) : window
Enter 10 words (8/10) : computer
Enter 10 words (9/10) : jelly
Enter 10 words (10/10) : earth
Before sorting
apple  pen  you  sun  moon  cat  window  computer  jelly  earth


After sorting
apple  cat  computer  earth  jelly  moon  pen  sun  window  you
계속하려면 아무 키나 누르십시오 . . .
```

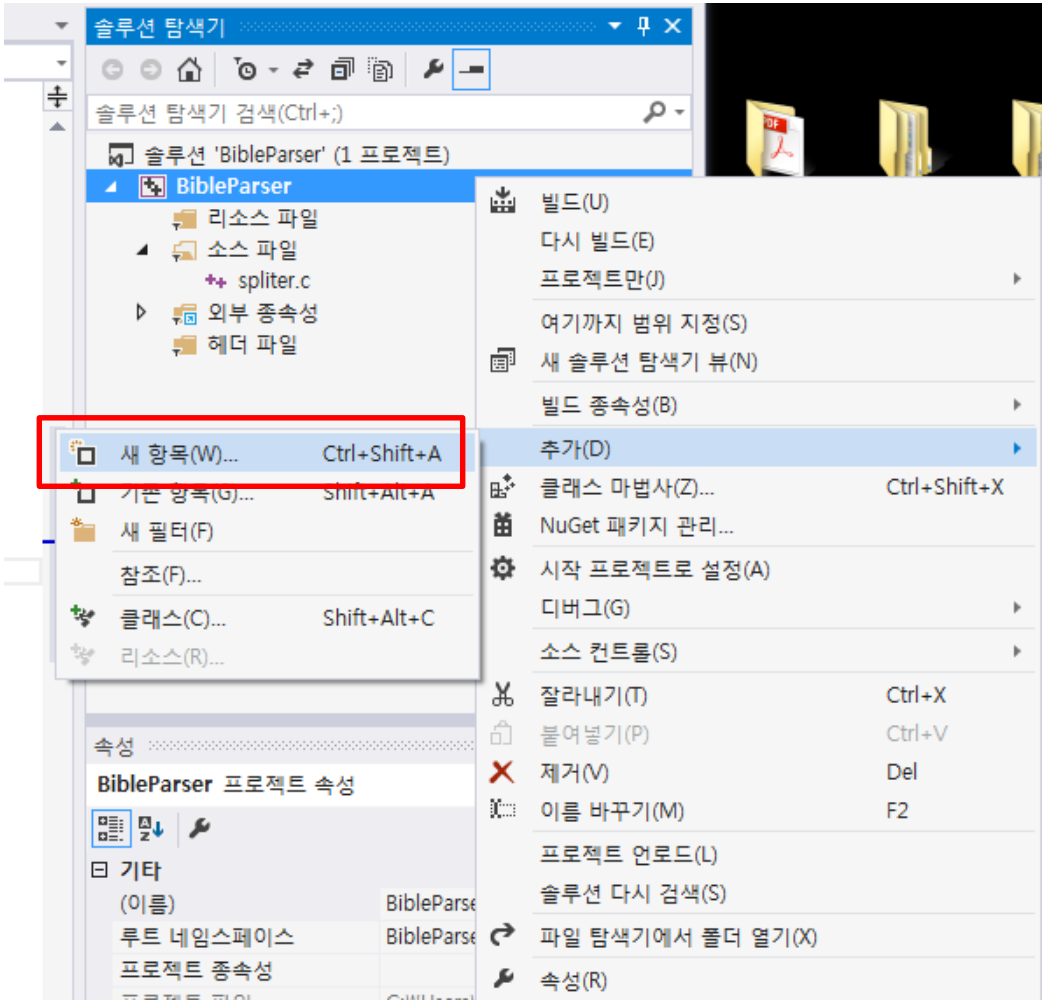
Appendix

Appendix 1 : Parsing HTML

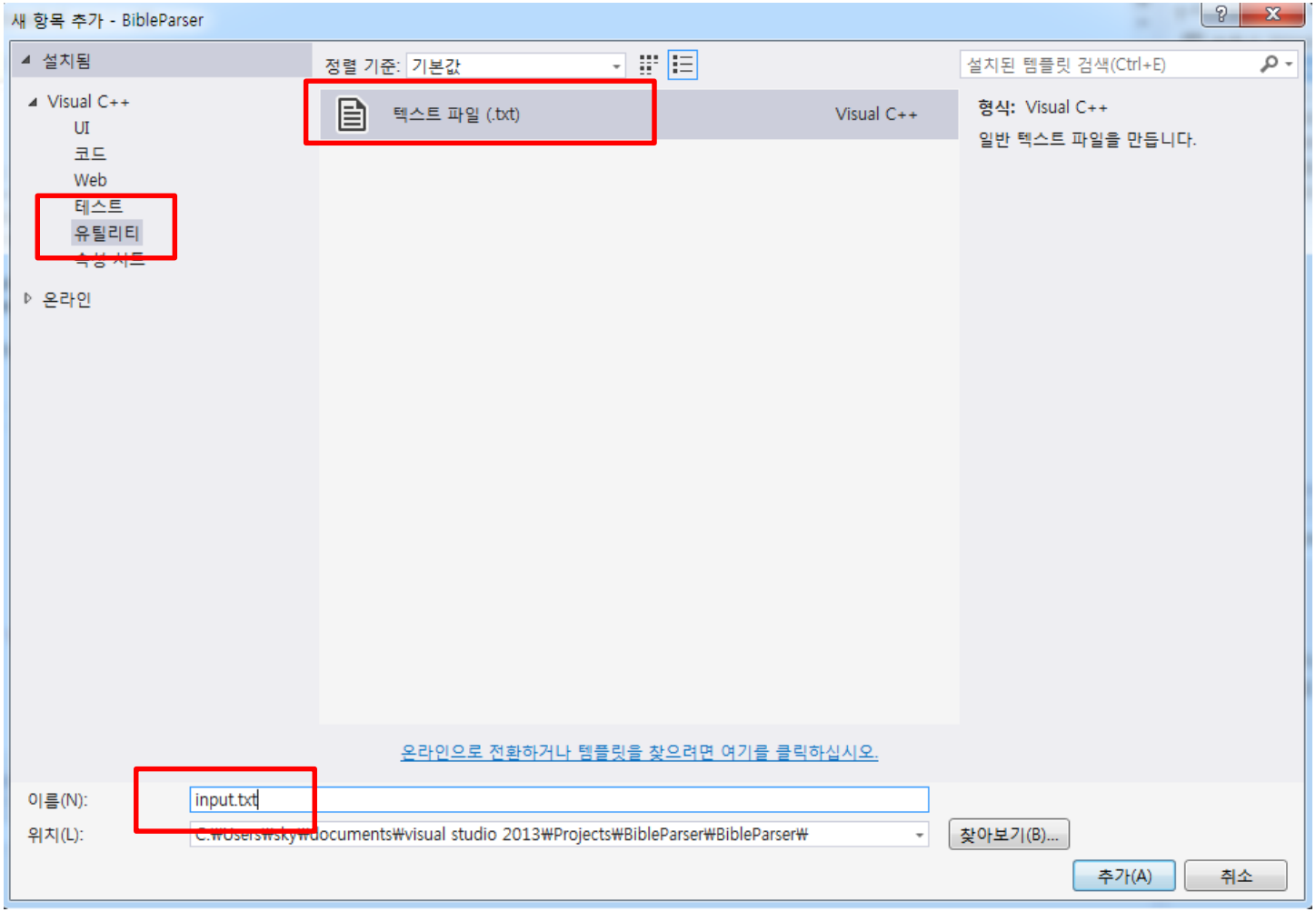
```
<li class="ah_item">
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r">1</span>
<span class="ah_k">빈지노</span>
</a>
</li>
<li class="ah_item">
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r">2</span>
<span class="ah_k">정찬우</span>
</a>
</li>
<li class="ah_item">
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r">3</span>
<span class="ah_k">재지팩트</span>
</a>
</li>
<li class="ah_item">
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r">4</span>
<span class="ah_k">서훈</span>
</a>
</li>
<li class="ah_item">
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r">5</span>
<span class="ah_k">아이콘</span>
</a>
</li>
```

- Naver webpage's real time searching keyword ranking source is implemented as left code.
- Parsing that source and print the keywords.

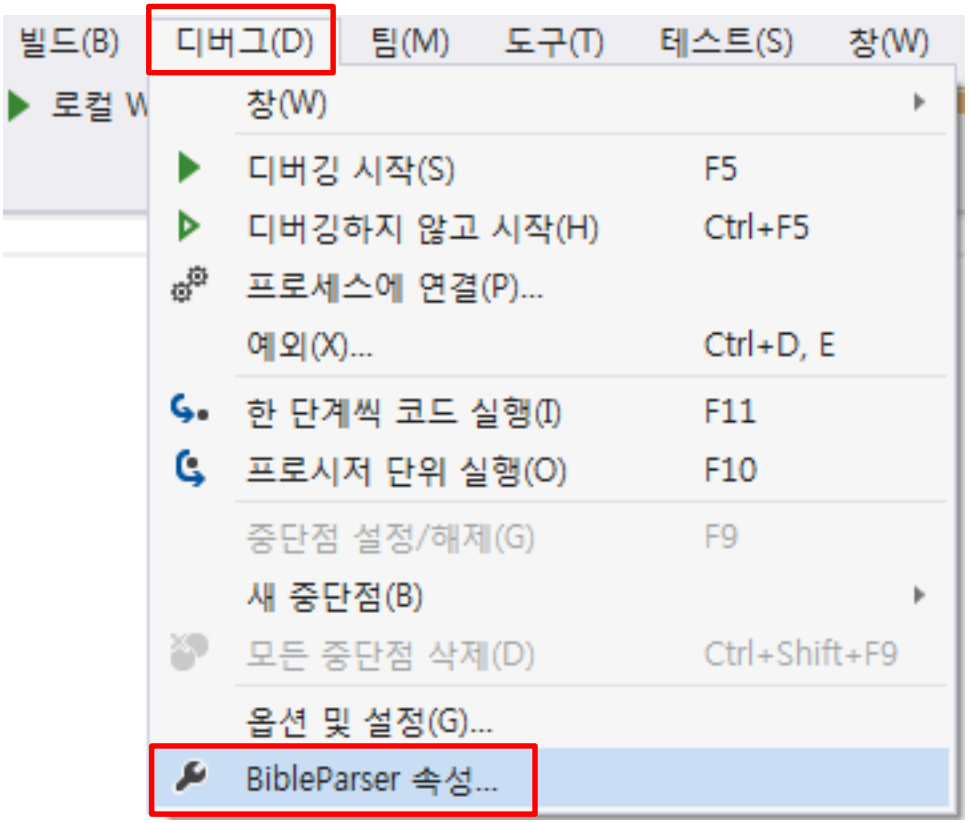
Input with text file



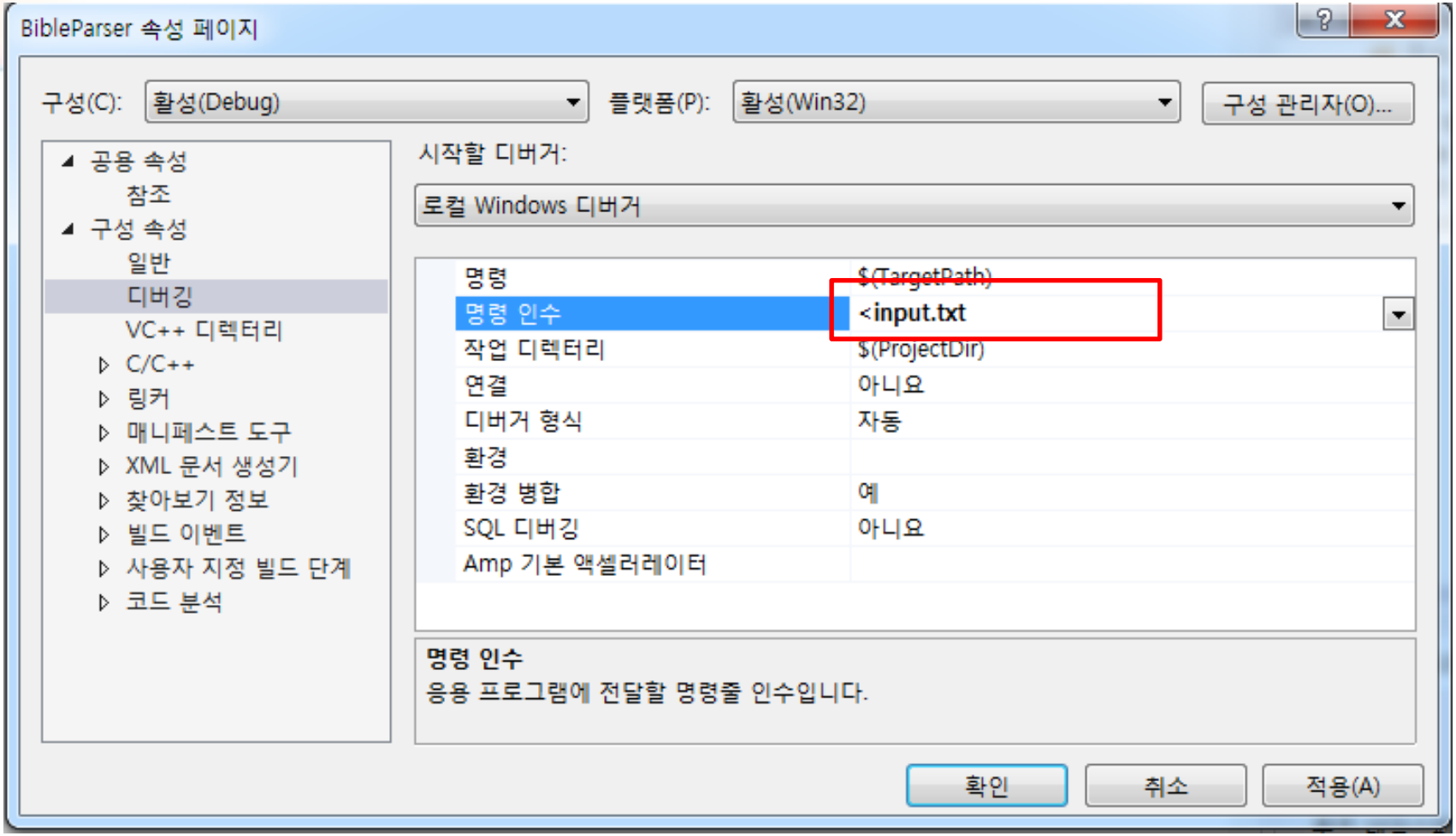
Input with text file



Input with text file



Input with text file



THANK YOU!!!