

---

# Objected Oriented Programming HW1



---

C111044 김호성

## #1-1 소스코드 & 코멘트

소스코드	코멘트
<pre>HW1_1_1.cpp 1 // Object Oriented Programming 2 // HW1-1 1) C++ string class 3 // C111044 김호성 4 5 #include &lt;iostream&gt; 6 #include &lt;string&gt; 7 using namespace std; 8 9 int main() { 10     int repeat = 5; 11     string id; 12     string password; 13 14     cout &lt;&lt; "프로그램을 종료하려면 아이디와 비밀번호를 입력하세요." &lt;&lt; endl; 15 16     for (int i = 0; i &lt; repeat; i++) { 17         cout &lt;&lt; "아이디 &gt;&gt; "; 18         getline(cin, id); 19         cout &lt;&lt; "비밀번호 &gt;&gt; "; 20         getline(cin, password); 21 22         if (id == "OOP" &amp;&amp; password == "C++") { 23             cout &lt;&lt; "성공! 프로그램을 정상 종료합니다." &lt;&lt; endl; 24             return 0; 25         } 26         else { 27             cout &lt;&lt; "아이디 또는 비밀번호가 틀립니다~~" &lt;&lt; endl; 28             if (i != repeat - 1) { 29                 cout &lt;&lt; "다시 시도하세요." &lt;&lt; endl; 30             } 31         } 32     } 33     cout &lt;&lt; "실패! 5번 틀렸습니다. 프로그램을 종료합니다." &lt;&lt; endl; 34 }</pre>	<ul style="list-style-type: none"><li>■ 문제의 조건인 C++ string class를 사용하기 위해 #include &lt;string&gt; 해주었습니다.</li><li>■ repeat [5] 만큼 for문을 통해 반복해서, 아이디와 비밀번호를 입력받고 각각 "OOP" "C++" 과 일치하는지 확인합니다.</li><li>■ 아이디와 비번 모두 일치하면 성공 메시지를 출력하고 즉시 프로그램을 종료합니다.</li><li>■ 아이디나 비번 둘 중 틀린 것이 있으면 틀리다는 메시지를 출력하고 for문을 이어서 수행합니다.</li><li>■ for문의 repeat회 반복 동안 아이디와 비밀번호를 맞게 입력하지 못한다면 실패메세지와 함께 프로그램을 종료합니다.</li><li>■ 다시 시도하세요 출력에 if (i != repeat - 1) 조건이 붙어있는 것은 5회 실패시 "다시 시도하세요." 와 "실패! 5번 틀렸습니다. 프로그램을 종료합니다." 이 두 메시지가 같이 나오는 것이 아니라 실패 메시지만 나오게 하기 위함입니다.</li><li>■ C++ string class를 사용해서 변수 선언도 string id; 와 같이 간단하게 했고 string끼리의 일치여부도 ==을 사용하여 간단하게 판별했습니다.</li></ul>



## #1-1 수행 결과

수행결과	테스트 데이터
 Microsoft Visual Studio 디버그 콘솔 프로그램을 종료하려면 아이디와 비밀번호를 입력하세요. 아이디 >> OOP 비밀번호 >> hosung 아이디 또는 비밀번호가 틀립니다~~ 다시 시도하세요. 아이디 >> hosung 비밀번호 >> C++ 아이디 또는 비밀번호가 틀립니다~~ 다시 시도하세요. 아이디 >> hosung 비밀번호 >> hongik 아이디 또는 비밀번호가 틀립니다~~ 다시 시도하세요. 아이디 >> hongik 비밀번호 >> cpp 아이디 또는 비밀번호가 틀립니다~~ 다시 시도하세요. 아이디 >> hongik 비밀번호 >> univ 아이디 또는 비밀번호가 틀립니다~~ 실패! 5번 틀렸습니다. 프로그램을 종료합니다.  J:\Source\HongikObjectOrientedProgramming\HW1\HW1_1_1\x64\Debug\HW1_1_1.exe(프로세스 14452개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	OOP hosung hosung C++ hosung hongik hongik cpp hongik univ
 Microsoft Visual Studio 디버그 콘솔 프로그램을 종료하려면 아이디와 비밀번호를 입력하세요. 아이디 >> OOP 비밀번호 >> C++ 성공! 프로그램을 정상 종료합니다.  J:\Source\HongikObjectOrientedProgramming\HW1\HW1_1_1\x64\Debug\HW1_1_1.exe(프로세스 17980개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	OOP C++

## #1-2 소스코드 & 코멘트

소스코드	코멘트
<pre>HW1_1_2.cpp HW1_1_2 (전역 범위) 1 // Object Oriented Programming 2 // HW1-1 2) C-Style String 3 // C111044 김호성 4 5 #include &lt;iostream&gt; 6 #include &lt;cstring&gt; 7 using namespace std; 8 9 int main() { 10     int repeat = 5; 11     char id[100]; 12     char password[100]; 13 14     cout &lt;&lt; "프로그램을 종료하려면 아이디와 비밀번호를 입력하세요." &lt;&lt; endl; 15 16     for (int i = 0; i &lt; repeat; i++) { 17         cout &lt;&lt; "아이디 &gt;&gt; "; 18         cin.getline(id, 100, '\n'); 19         cout &lt;&lt; "비밀번호 &gt;&gt; "; 20         cin.getline(password, 100, '\n'); 21 22         if (strcmp(id, "OOP") == 0 &amp;&amp; strcmp(password, "C++") == 0) { 23             cout &lt;&lt; "성공! 프로그램을 정상 종료합니다." &lt;&lt; endl; 24             return 0; 25         } 26         else { 27             cout &lt;&lt; "아이디 또는 비밀번호가 틀립니다~~" &lt;&lt; endl; 28             if (i != repeat - 1) { 29                 cout &lt;&lt; "다시 시도하세요." &lt;&lt; endl; 30             } 31         } 32     } 33     cout &lt;&lt; "실패! 5번 틀렸습니다. 프로그램을 종료합니다." &lt;&lt; endl; 34 }</pre>	<ul style="list-style-type: none"><li>■ 문제의 조건인 C-Style string을 사용하기 위해 #include &lt;cstring&gt; 해주었습니다.</li><li>■ char id[100];와 같이 입력받을 아이디와 비밀번호를 저장할 변수를 선언했습니다.</li><li>■ cin.getline(id, 100, '\n');을 사용해서 Enter키를 만나기 전까지 최대 99개의 문자를 읽어오게 했습니다.</li><li>■ 문자열끼리 동일 여부를 확인하기 위해 strcmp() 함수를 사용했습니다.</li><li>■ 이외는 #1-1과 동일하므로 생략하겠습니다.</li></ul>

## #1-2 수행 결과

수행결과	테스트 데이터
 Microsoft Visual Studio 디버그 콘솔 프로그램을 종료하려면 아이디와 비밀번호를 입력하세요. 아이디 >> OOP 비밀번호 >> hosung 아이디 또는 비밀번호가 틀립니다~~ 다시 시도하세요. 아이디 >> hosung 비밀번호 >> C++ 아이디 또는 비밀번호가 틀립니다~~ 다시 시도하세요. 아이디 >> hosung 비밀번호 >> hongik 아이디 또는 비밀번호가 틀립니다~~ 다시 시도하세요. 아이디 >> hongik 비밀번호 >> cpp 아이디 또는 비밀번호가 틀립니다~~ 다시 시도하세요. 아이디 >> hongik 비밀번호 >> univ 아이디 또는 비밀번호가 틀립니다~~ 실패! 5번 틀렸습니다. 프로그램을 종료합니다.  D:\Source\HongikObjectOrientedProgramming\HW1\HW1_1_2\x64\Debug\HW1_1_2.exe(프로세스 14280개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	OOP hosung hosung C++ hosung hongik hongik cpp hongik univ
 Microsoft Visual Studio 디버그 콘솔 프로그램을 종료하려면 아이디와 비밀번호를 입력하세요. 아이디 >> OOP 비밀번호 >> C++ 성공! 프로그램을 정상 종료합니다.  D:\Source\HongikObjectOrientedProgramming\HW1\HW1_1_2\x64\Debug\HW1_1_2.exe(프로세스 27332개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	OOP C++

## #2 소스코드 & 코멘트

소스코드	코멘트
<pre>HW1_2.cpp HW1_2 1 // Object Oriented Programming 2 // HW1-2 3 // C111044 김호성 4 5 #include &lt;iostream&gt; 6 using namespace std; 7 8 int main() { 9     int num; 10 11     cout &lt;&lt; "하나의 정수를 입력하시오 : "; 12     cin &gt;&gt; num; 13 14     if (num &lt; 2) { 15         cout &lt;&lt; num &lt;&lt; "은(는) 소수가 아닙니다." &lt;&lt; endl; 16         return 0; 17     } 18 19     for (int i=2; i&lt;num; i++) { 20         if (num % i == 0) { 21             cout &lt;&lt; num &lt;&lt; "은(는) 소수가 아닙니다." &lt;&lt; endl; 22             return 0; 23         } 24     } 25     cout &lt;&lt; num &lt;&lt; "은(는) 소수입니다." &lt;&lt; endl; 26 27     return 0; 28 }</pre>	<ul style="list-style-type: none"><li>■ num을 입력받아서 for문으로 반복하며 num을 2~num-1로 나눕니다. 이때, 나누어떨어지는 것이 하나라도 있으면 “소수가 아닙니다” 메세지 출력 후 프로그램을 종료하고 나누어떨어지는 것 하나 없이 for문이 종료되면 “소수입니다” 메세지 출력 후 프로그램을 종료합니다.</li><li>■ 이때, for문 전에 num&lt;2인지 확인하여 만약 맞다면 “소수가 아닙니다” 메세지 출력 후 프로그램을 종료합니다. 이는 1이 소수로 출력되는 예외를 처리하기 위함입니다.</li></ul>



## #2 수행 결과

수행결과	테스트 데이터
 Microsoft Visual Studio 디버그 콘솔 하나의 정수를 입력하시오 : 157 157은(는) 소수입니다.  D:\Source\HongikObjectOrientedProgramming\HW1\HW1_2\x64\Debug\HW1_2.exe(프로세스 11212개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	157
 Microsoft Visual Studio 디버그 콘솔 하나의 정수를 입력하시오 : 1 1은(는) 소수가 아닙니다.  D:\Source\HongikObjectOrientedProgramming\HW1\HW1_2\x64\Debug\HW1_2.exe(프로세스 9236개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	1
 Microsoft Visual Studio 디버그 콘솔 하나의 정수를 입력하시오 : 6 6은(는) 소수가 아닙니다.  D:\Source\HongikObjectOrientedProgramming\HW1\HW1_2\x64\Debug\HW1_2.exe(프로세스 23304개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	6
 Microsoft Visual Studio 디버그 콘솔 하나의 정수를 입력하시오 : 2 2은(는) 소수입니다.  D:\Source\HongikObjectOrientedProgramming\HW1\HW1_2\x64\Debug\HW1_2.exe(프로세스 24412개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	2
 Microsoft Visual Studio 디버그 콘솔 하나의 정수를 입력하시오 : 7 7은(는) 소수입니다.  D:\Source\HongikObjectOrientedProgramming\HW1\HW1_2\x64\Debug\HW1_2.exe(프로세스 25152개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	7

## #3 소스코드 & 코멘트

### 소스코드




```
HW1_3.cpp
HW1_3
1 // Object Oriented Programming
2 // HW1-3
3 // C111044 김호성
4
5 #include <iostream>
6 using namespace std;
7
8 int getGCD(int num1, int num2) {
9     while (num2) {
10         int temp = num1 % num2;
11         num1 = num2;
12         num2 = temp;
13     }
14     return num1;
15 }
16
17 int main() {
18     int num1;
19     int num2;
20     int gcd;
21
22     cout << "두 수를 입력하시오 : ";
23     cin >> num1 >> num2;
24
25     gcd = getGCD(num1, num2);
26
27     cout << "두 수의 공약수 : ";
28     for (int i=1; i<=gcd; i++) {
29         if (gcd % i == 0) {
30             cout << i << " ";
31         }
32     }
33     cout << endl << "최소공배수 : " << num1*num2/gcd << endl;
34
35     return 0;
36 }
```

### 코멘트

- 유클리드 호제법을 사용하여 최대공약수를 구하는 getGCD()함수를 정의해주었습니다.
- 이 getGCD함수를 통해 공약수는 최대공약수의 약수들로, 최소공배수는 두 수의 곱에 최대공약수를 나누어주는 것으로 쉽게 공약수와 최소공배수를 구할 수 있습니다.



## #3 수행 결과

수행결과	테스트 데이터
 Microsoft Visual Studio 디버그 콘솔 두 수를 입력하시오 : 16 24 두 수의 공약수 : 1 2 4 8 최소공배수 : 48 D:\Source\HongikObjectOrientedProgramming\HW1\HW1_3\x64\Debug\HW1_3.exe(프로세스 13260개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	16 24
 Microsoft Visual Studio 디버그 콘솔 두 수를 입력하시오 : 3 5 두 수의 공약수 : 1 최소공배수 : 15 D:\Source\HongikObjectOrientedProgramming\HW1\HW1_3\x64\Debug\HW1_3.exe(프로세스 9880개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	3 5
 Microsoft Visual Studio 디버그 콘솔 두 수를 입력하시오 : 3 6 두 수의 공약수 : 1 3 최소공배수 : 6 D:\Source\HongikObjectOrientedProgramming\HW1\HW1_3\x64\Debug\HW1_3.exe(프로세스 14976개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	3 6

## #4 소스코드 & 코멘트




### 소스코드

```
HW1_4.cpp
HW1_4 (전역 범위)
1 // Object Oriented Programming
2 // HW1-4
3 // C111044 김호성
4
5 #include <iostream>
6 #include <string>
7 using namespace std;
8
9 string convertToHexadecimal(int decimal) {
10     string result = "";
11     char hexadecimal[16] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F' };
12     while (decimal) {
13         result = hexadecimal[decimal % 16] + result;
14         decimal = decimal / 16;
15     }
16     if (result == "") {
17         result = "0";
18     }
19     return result;
20 }
21
22 int main() {
23     int num;
24
25     cout << "십진수를 입력하시오 : ";
26     cin >> num;
27
28     cout << "17의 16진수는 " << convertToHexadecimal(num);
29
30     return 0;
31 }
```

### 코멘트

- 16진수 숫자들(0~F)을 char 배열에 저장해두었습니다.
- 입력받은 10진수를 나누어서, 나머지는 result의 왼쪽에 char 배열을 통해 16진수 숫자char로 추가해주고 몫은 decimal에 저장합니다.
- while문으로, decimal이 0이 될 때까지 위 과정을 반복합니다.
- 0이 입력되었을 때 그냥 빈 문자열 "" 가 return되는 것을 막기 위해 result== "" 이면 result= "0" 이 되도록 조건문을 하나 추가하였습니다.

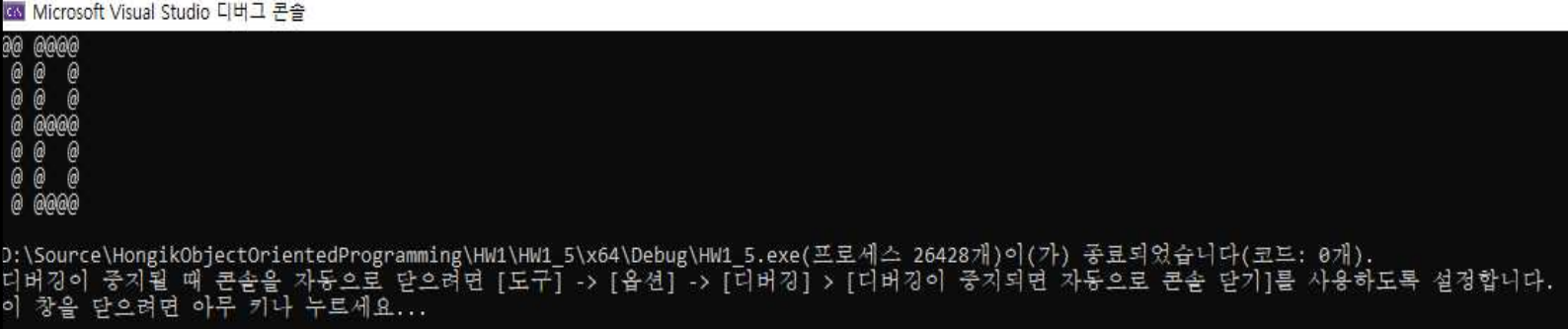
## #4 수행 결과

수행결과	테스트 데이터
 Microsoft Visual Studio 디버그 콘솔 십진수를 입력하시오 : 17 17의 16진수는 11 D:\Source\HongikObjectOrientedProgramming\HW1\HW1_4\x64\Debug\HW1_4.exe(프로세스 12484개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	17
 Microsoft Visual Studio 디버그 콘솔 십진수를 입력하시오 : 14 17의 16진수는 E D:\Source\HongikObjectOrientedProgramming\HW1\HW1_4\x64\Debug\HW1_4.exe(프로세스 15060개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	14
 Microsoft Visual Studio 디버그 콘솔 십진수를 입력하시오 : 5108 17의 16진수는 13F4 D:\Source\HongikObjectOrientedProgramming\HW1\HW1_4\x64\Debug\HW1_4.exe(프로세스 23128개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	5108

## #5 소스코드 & 코멘트

소스코드	코멘트
<pre> 1  // Object Oriented Programming 2  // HW1-5 3  // C111044 김호성 4 5  #include &lt;iostream&gt; 6  using namespace std; 7 8  int main() { 9      int a[7][7] = { 10         { 77, 28, 31, 392, 224, 84, 161}, 11         { 60, 84, 53, 476, 32, 255, 7}, 12         { 515, 147, 76, 455, 12, 243, 21}, 13         { 9, 224, 141, 287, 56, 42, 441}, 14         { 87, 161, 142, 224, 8, 9, 231}, 15         { 243, 21, 23, 35, 111, 87, 357}, 16         { 255, 7, 90, 70, 63, 224, 147} 17     }; 18     char b[7][7]; 19 20     for (int i=0; i&lt;7; i++) { 21         for (int j=0; j&lt;7; j++) { 22             if (a[i][j] % 7 == 0) { 23                 b[i][j] = '@'; 24             } else { 25                 b[i][j] = ' '; 26             } 27         } 28     } 29     for (int i=0; i&lt;7; i++) { 30         for (int j=0; j&lt;7; j++) { 31             cout &lt;&lt; b[i][j]; 32         } 33         cout &lt;&lt; endl; 34     } 35 } </pre>	<ul style="list-style-type: none"> <li>■ 이중 for문을 통해 a배열을 하나씩 읽었습니다. 만약 a[i][j]가 7로 나누어떨어진다면 같은 크기의 char배열 b의 같은 위치에 '@' 를 넣어주고, 나누어떨어지지 않는다면 공백 ' ' 을 넣어주었습니다.</li> <li>■ a배열을 읽은 것과 동일하게 이중for문을 통해 b배열을 읽어주고 하나하나 출력했습니다. 한줄이 끝나면 endl을 출력하여 줄도 바꿔했습니다.</li> </ul>

## #5 수행 결과

수행결과	테스트 데이터
 <p>D:\Source\HongikObjectOrientedProgramming\HW1\HW1_5\x64\Debug\HW1_5.exe(프로세스 26428개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -&gt; [옵션] -&gt; [디버깅] &gt; [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...</p>	-

## #6 소스코드

```
HW1_6.cpp # X
HW1_6 (전역 범위) main()

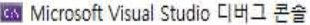
1 // Object Oriented Programming
2 // HW1-6
3 // C111044 김호성
4
5 #include <iostream>
6 using namespace std;
7
8 int count_words(const char* str) {
9     int count = 0;
10    int flag = 0;
11    for (const char* i=str; *i!='\0'; i++) {
12        if (*i == ' ') {
13            if (flag) {
14                count++;
15            }
16            flag = 0;
17        } else {
18            flag = 1;
19        }
20    }
21    return count+flag;
22 }
23
24 int count_sentences(const char* str) {
25     int count = 0;
26     int flag = 0;
27     for (const char* i=str; *i!='\0'; i++) {
28         if (*i >= 'A' && *i <= 'Z') {
29             if (flag == 0) {
30                 flag = 1;
31             }
32         } else if (*i >= 'a' && *i <= 'z') {
33             if (flag == 0) {
34                 flag = -1;
35             }
36         } else if (*i == '.') {
37             if (flag == 1) {
38                 count++;
39             }
40             flag = 0;
41         }
42     }
43     return count;
44 }
45
46 int main() {
47     const char* str1 = "Object-oriented programming is a programming paradigm based on the concept of objects, which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. A featur
48     const char* str2 = "melon, apple, grape, kiwi, pear, orange";
49     const char* str3 = " ";
50
51     cout << "첫 번째 문자열에 포함된 단어 수 : " << count_words(str1) << endl;
52     cout << "두 번째 문자열에 포함된 단어 수 : " << count_words(str2) << endl;
53     cout << "세 번째 문자열에 포함된 단어 수 : " << count_words(str3) << endl << endl;
54
55     cout << "첫 번째 문자열에 포함된 문장 수 : " << count_sentences(str1) << endl;
56     cout << "두 번째 문자열에 포함된 문장 수 : " << count_sentences(str2) << endl;
57     cout << "세 번째 문자열에 포함된 문장 수 : " << count_sentences(str3) << endl;
58 }
```

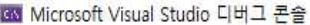


## #6 코멘트

count_words	count_sentences
<ul style="list-style-type: none"><li>■ countWord에서 for문을 통해 문자열을 한글자씩 확인할 때 마지막으로 확인한 글자가 space면 flag는 0이고 space가 아닌 문자면 1입니다.</li><li>■ 따라서 flag가 1-&gt;0이 될 때를 세면 단어의 개수를 쉽게 셀 수 있습니다.</li><li>■ 위의 아이디어를 활용하여 <code>for (const char* i=str; *i!='\0'; i++)</code> 다음과 같은 for문으로 문자열의 시작부터 끝까지 한글자씩 확인합니다. i가 띄어쓰기면 flag가 1인지 확인하여 1이라면 count를 1늘려줍니다. i가 띄어쓰기였으므로 flag에 0을 넣어줍니다. i가 띄어쓰기가 아니라면 flag에 1을 넣고 계속해서 for문을 수행합니다.</li><li>■ count를 return하는 것이 아니라 count+flag를 return하는 이유는 띄어쓰기일 때만 count를 늘려주기 때문에 마지막에 띄어쓰기로 끝나지 않을 때 마지막 단어를 세는 것이 누락되는 것을 막기 위함입니다.</li></ul>	<ul style="list-style-type: none"><li>■ flag = 0은 마지막으로 확인한 글자가 마침표(.)인 경우이고, flag = 1은 마침표 이후 대문자가 나와서 정상적인 문장이 진행 중인 경우이고, flag = -1은 마침표 이후 소문자가 나와서 비정상적인(문장으로 count할 수 없는) 문장이 진행 중인 경우입니다.</li><li>■ 따라서 flag가 1-&gt;0이 될 때를 세면 문장의 개수를 쉽게 셀 수 있습니다.</li><li>■ 위의 아이디어를 활용하여 <code>for (const char* i=str; *i!='\0'; i++)</code> 다음과 같은 for문으로 문자열의 시작부터 끝까지 한글자씩 확인합니다. i가 대문자고 flag가 0이라면 flag에 1을 넣어주고, i가 소문자고 flag가 0이라면 -1을 넣어줍니다. i가 마침표라면 flag가 1인지 확인하여 1이라면 count를 1늘려줍니다. i가 마침표였으므로 flag에 0을 넣어줍니다.</li></ul>

## #6 수행 결과

수행결과	테스트 데이터
 Microsoft Visual Studio 디버그 콘솔 첫 번째 문자열에 포함된 단어 수 : 105 두 번째 문자열에 포함된 단어 수 : 6 세 번째 문자열에 포함된 단어 수 : 0  첫 번째 문자열에 포함된 문장 수 : 4 두 번째 문자열에 포함된 문장 수 : 0 세 번째 문자열에 포함된 문장 수 : 0  D:\Source\HongikObjectOrientedProgramming\HW1\HW1_6\x64\Debug\HW1_6.exe(프로세스 28588개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	-

추가 수행결과	테스트 데이터
 Microsoft Visual Studio 디버그 콘솔 첫 번째 문자열에 포함된 단어 수 : 9 두 번째 문자열에 포함된 단어 수 : 6 세 번째 문자열에 포함된 단어 수 : 0  첫 번째 문자열에 포함된 문장 수 : 1 두 번째 문자열에 포함된 문장 수 : 0 세 번째 문자열에 포함된 문장 수 : 0  D:\Source\HongikObjectOrientedProgramming\HW1\HW1_6\x64\Debug\HW1_6.exe(프로세스 12780개)이(가) 종료되었습니다(코드: 0개). 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다. 이 창을 닫으려면 아무 키나 누르세요...	첫 번째 문자열str1에 “My name is HosungKim. hi. Nice to meet you” 를 대신 대입. [소문자로 시작하는 문장, 마침표로 끝나지 않는 문장 테스트]