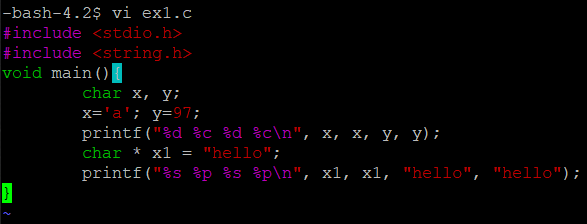
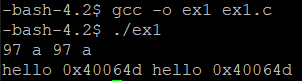
**System Programming Lect5 HW**

**001분반 / 12161756 / 윤성호**

5. Homework

1) [A char constant is an ascii number. A string constant is an address where it is stored in the string area.] Explain the result for following code.

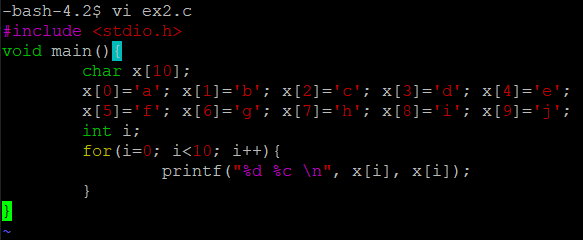
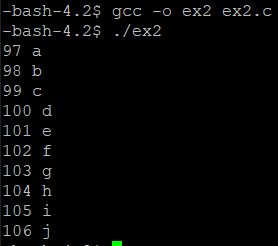
 

**= 1. vi 편집기로 C프로그래밍을 한 후 2. [gcc]로 컴파일, 3. [./]로 실행하였다.**

**= 첫 번째 printf에서 [%d]는 정수형, [%c]는 문자 1개를 출력하라는 의미이다. 따라서 x와 97의 정수표현(아스키코드값)인 ‘97’과 문자표현값 ‘a’이 번갈아 나타났다.**

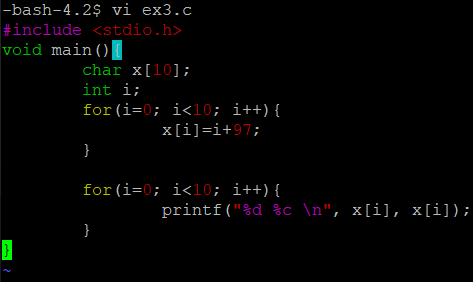
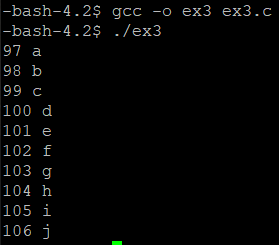
**= 두 번째 printf에서 [%s]는 문자열, [%p]는 포인터 주소를 출력하라는 의미이다. 따라서 x1과 ‘hello’의 문자열값 ‘hello’와 포인터 주소값 ‘0x40064d’이 번갈아 나타났다.**

2) [A char constant is an ascii number] Try following code and explain the result.

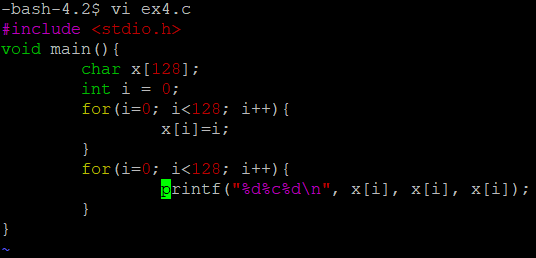
**= char x[10]을 선언한 후 a~j까지 값으로 앞에서부터 순서대로 초기화한다. 그 후 for문으로 아스키코드값과 문자값을 한 줄씩 출력한다.**

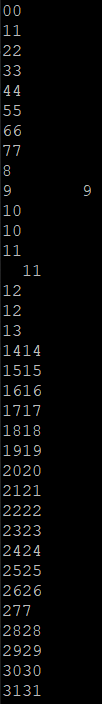
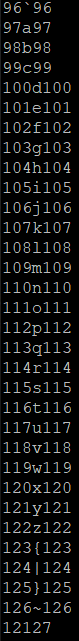
3) Try below. Compare the result with that of Problem 2).

**= char x[10]을 선언 후 x[0]~x[9]을 단순한 숫자 97~106으로 초기화한 후 for문으로 출력하자 2)번과 동일하게 출력되었다.**

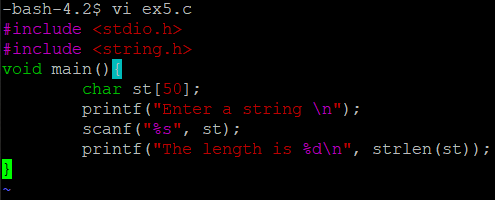
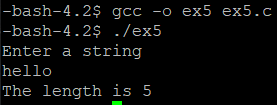
4) Declare a character array with 128 rooms. Store 0 to 127 in this array and print the corresponding character for each ascii code in the array. Find ASCII table in the Internet and confirm the results.

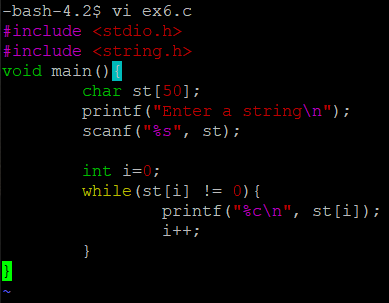
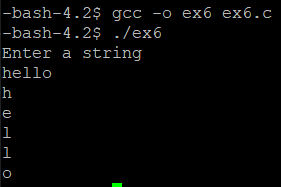
**= 우측 아스키코드표와 왼쪽 실행 결과가 일치하는 모습을 볼 수 있다.**

5) [strlen] Read a string and display its length.

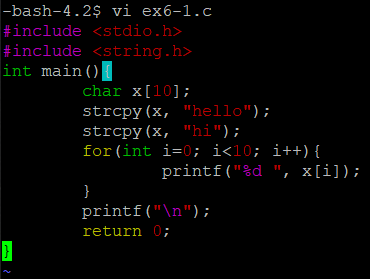
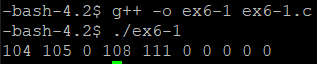
**= strlen를 이용하여 문자열의 길이를 출력한다.**

6) [A string is a char array ending with 0] Read a string and display each character in different lines.

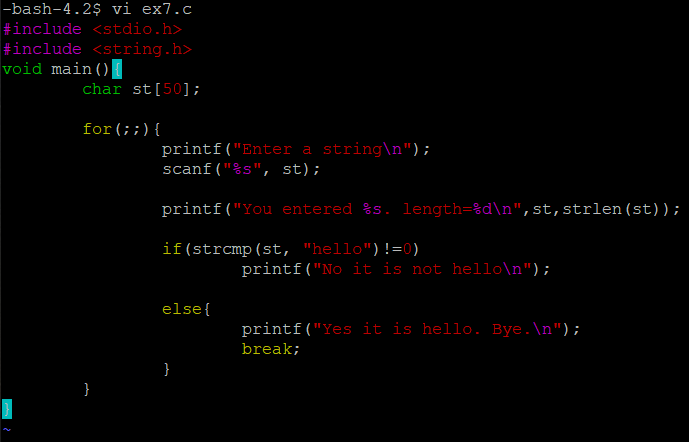
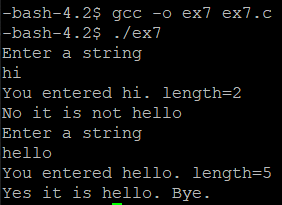
**= 문자열 끝에는 0이 자리잡는 성질을 이용하였다. 입력받은 문자열을 앞에서부터 하나씩 비교하여 0일 때 printf를 그만두는 while문을 작성하였다.**

6-1) [A string is a char array ending with 0] Try below and explain the result. Use g++ to compile.

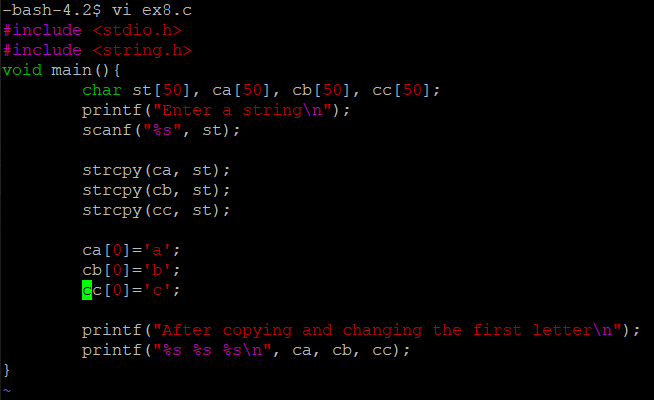
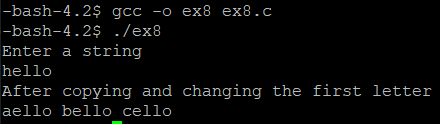
**= 처음에 char 10byte를 생성한 후 strcpy로 ‘hello’를 0번째 칸부터 한글자씩 채워넣는다. 이어지는 strcpy에서는 이미 공간이 차있는 x문자열 위에 “hi”를 덮어쓴다. 그렇기에 0번째 칸과 1번재 칸에는 아스키코드로 각각 h와 i를 의미하는 104, 105가 입력된다. 문자열은 끝에 0을 수반하기에 2번째 칸에 0이 입력된다. 출력값을 보면 3번째 칸과 4번째 칸은 덮어씌워지지 않아 108(l), 111(o)이 출력된다.**

7) [strlen, strcmp] Write a program that keeps reading a string, displaying its length, and checking whether it is "hello". If the input string is "hello", the program stops.

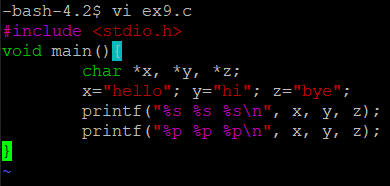
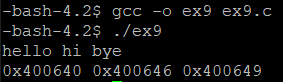
**= st라는 char 문자열을 선언한다. for 무한루프 안에서 문자열을 입력받도록 하고 strlen로 입력받은 문자열의 길이를 출력한다. 이후 strcmp로 조건문으로 입력받은 문자열과 ‘hello’를 비교한다. 비교 결과, 다른 경우 메시지 출력 후 무한루프를 돌도록하고 같은 경우(strcmp값이 0) 메시지를 출력 후 break로 무한루프를 탈출한다.**

8) [strcpy] Read a string and copy it to three other string variables and change the first letter of them to 'a', 'b', and 'c', respectivley, and display them.

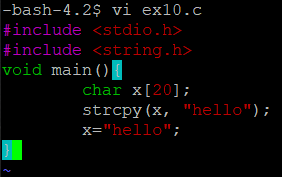
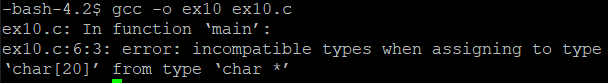
**= 4개의 문자열을 선언하고 st 문자열에 입력을 받는다. strcpy로 st를 ca, cb, cc로 복사한다. ca와 cb, cc의 각 0번째 칸에 a, b, c로 대체하여 결과를 출력한다.**

9) [string constant] A string constant such as "hello" is an address. Explain the result of following code.

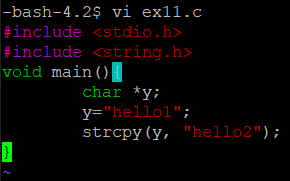
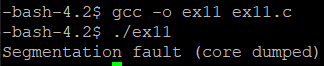
**= char형 포인터 변수 x, y, z를 선언한다. 각 포인터 변수에 ‘hello’, ‘hi’, ‘bye’의 주소를 저장한다. %s를 이용하자 포인터 변수가 가리키는 문자열값이, %p로는 포인터 변수가 저장하고 있는 주소값을 출력되었다.**

10) [string constant is an address] Try below and explain why we have an error.

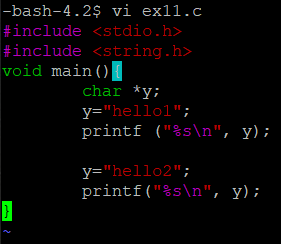
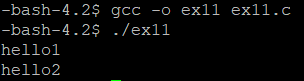
 

**= [x=”hello”;] 구문이 컴파일 에러가 나는 이유는 “hello”는 주소값이기 때문에 포인터 변수가 아닌 x에 저장할 수 없기 때문이다.**

11) [You need memory space for strcpy] Try below and explain why we have an error. How can you fix it?

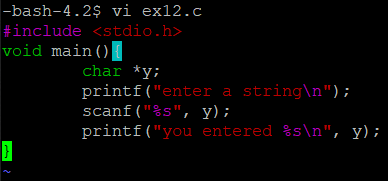
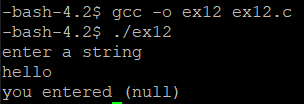
 

**= y는 포인터 변수로, 주소값을 저장한다. “hello1”의 주소를 y에 직접적으로 대입하는 방법은 가능하지만 문자열을 복사하는 strcpy 사용은 에러를 일으킨다.**

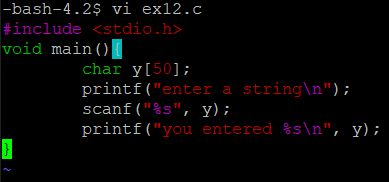
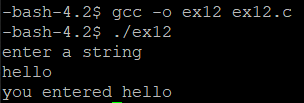
 

**= strcpy 대신 y에 직접적으로 “hello2”의 주소값을 대입하는 형식으로 수정하였다.**

12) [You need memory space for scanf] Try below and explain why you have an error. Fix it.

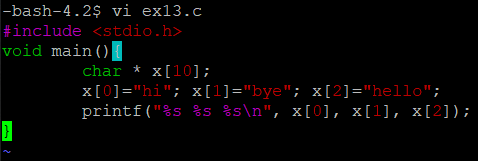
 

**= 여기서도 마찬가지로 y는 포인터 변수이므로 문자열을 저장할 수 없다.**

**= 포인터 변수로 선언했던 것을 char형 문자열로 선언하여 해결하였다.**

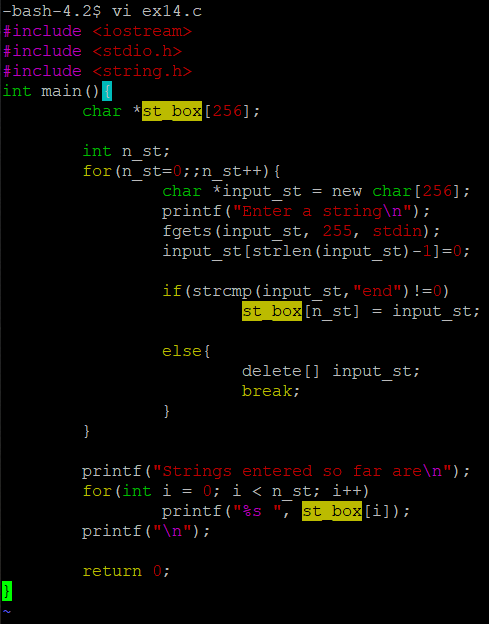
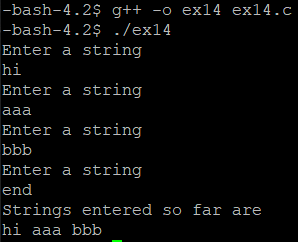
13) [char pointer array] Define a character pointer array and store/display strings as below.

**= char형 포인터 변수 앞칸부터 “hi”, “bye”, “hello”의 주소를 채운 후 각 주소가 가리키는 값을**

**출력하였다.**

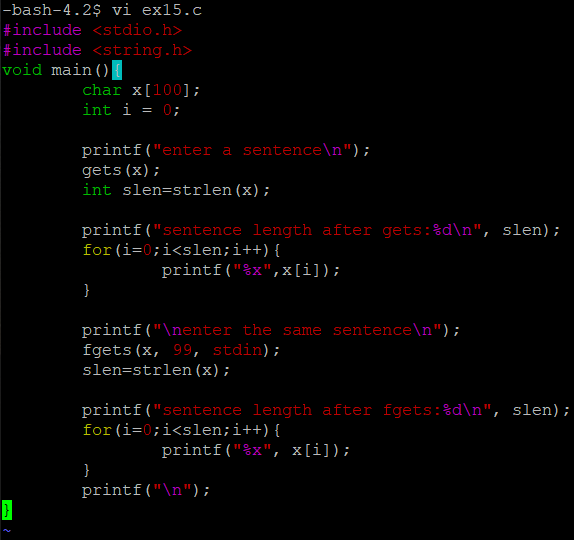
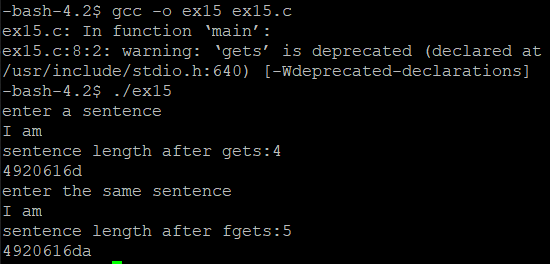
14) [char pointer array, strcmp, new] Write a program that keeps reading strings and store them in a character pointer array. It stops when the user enters "end" and displays all strings entered so far. Use "new" to allocate memory and use g++ to compile.

**= 입력할 string들을 가리킬 포인터 배열 st\_box를 선언한다. for문에서는 입력받는 string을 가리킬 input\_st라는 포인터 변수를 동적 할당한다.**

**= fgets로 string을 입력받고 strlen을 이용해 마지막 enter 입력을 제거한다. strcmp로 input\_st가 ‘end’가 아닐 때 st\_box에 input\_st를 지정해준다. ‘end’일 경우 동적 할당한 input\_st를 delete하고 for루프를 탈출한다. 이후 st\_box가 가리키는 값을 앞에서부터 차례로 출력한다.**

15) [gets, fgets] Read the same sentence with gets() and fgets() and explain the difference. (Ignore warning for gets. It is a security warning because gets can cause security problem.)

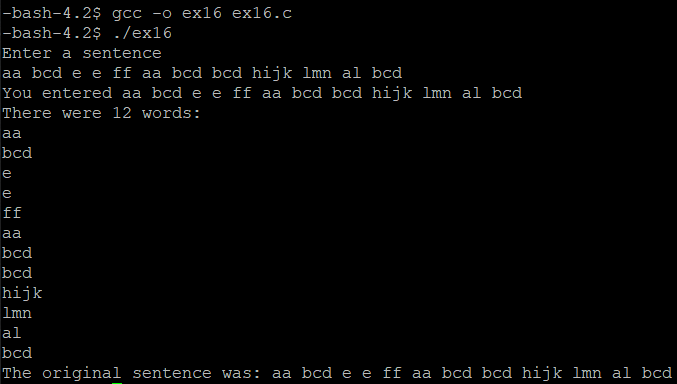
**= gets는 선언한 문자열 길이 이상으로도 입력을 받을 수 있다. 이 경우 메모리 문제가 발생한다.**

**반면 fgets는 입력받을 문자열의 길이를 매개변수로 넘겨 이런 상황을 방지한다.**

**= gets는 enter 입력을 문자열에 포함시키지 않으나** **fgets는 문자열의 일부로 본다. 그렇기에 같은 문자열을 입력해도 fgets로 입력받은 문자열의 길이는 1만큼 더 길다.**

16) [strtok] Use strtok to extract words from a sentence and store them in an array. Display the number of words as below. Note that you need to copy the sentence to another string variable before doing strtok because strtok will destroy the original sentence.





**= char 문자열 st와 st의 복사본을 담을 st\_c를 선언한다. fgets로 입력받은 문장을 tokenizing 한 후 그 token들을 가리킬 포인터 변수 st\_ad도 선언한다.**

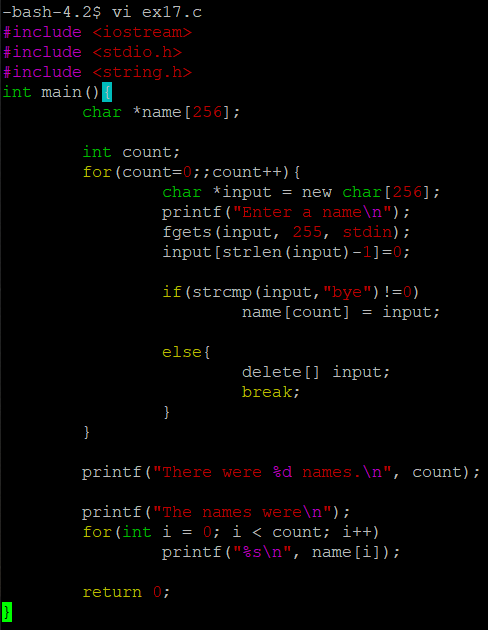
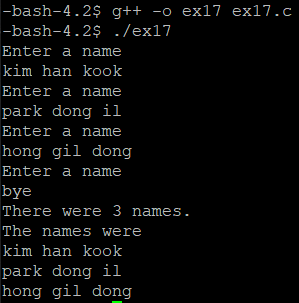
**= fgets로 문장을 입력받고 문자열 st에 저장한다. strlen를 이용해 st의 마지막 enter 입력을 0으로 바꿔준다. 그 후 st을 st\_c에 복사한다.**

**= 입력받은 st를 출력해주고 strtok로 while문을 돌면서 token이 NULL값이 될 때까지 포인터 변수 st\_ad에 각 token의 지정한다.**

**= while문에 사용되었던 변수 i로 단어 수를 출력할 수 있다. 이후 for문으로 st\_ad가 가리키는 token값을 출력한다.**

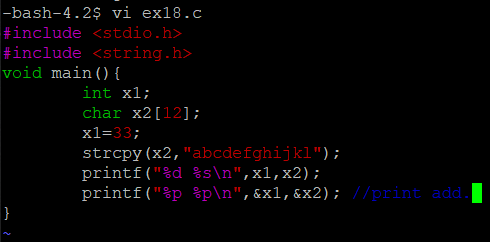
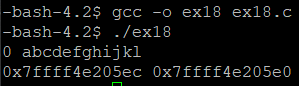
**= 원본 문장 st는 strtok를 이용하면서 ‘space’처리된 부분이 0으로 바뀌었기 때문에 st를 그대로 불러오면 문장의 첫 단어만 출력되고 그 뒤는 출력할 수 없다. 따라서 미리 복사해둔 st\_c를 이용한다.**

17) [char pointer array, new, strcmp] Write a program that keeps reading a name and stores it in a character pointer array until the user enters bye. The program should display all names after it sees "bye".

**= 알고리즘은 14번과 똑같다. 문장을 동적 할당한 input으로 입력받고 bye가 입력되면 그 전까지 입력된 이름들과 그 개수를 출력한다.**

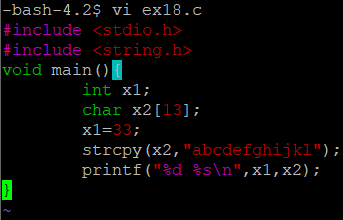
18) [There is a hidden 0 at the end of a string] Try below and explain why it behaves strange. How can you fix it?

**= x1이 33이 나올 것으로 예상했으나 0이 출력되었다. 원인을 찾기 위해 x1과 x2의 주소값을 출력하는 구문 한 줄을 맨 아래 추가하였다. 0이 출력되는 과정을 표로 정리하면 다음과 같다.**

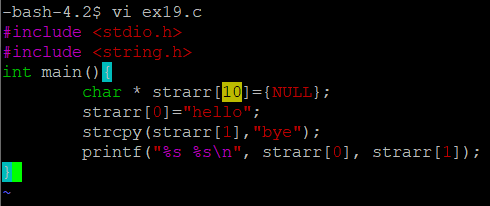
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **주소 맨끝자리(-205e-)** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **a** | **b** | **c** |
| **x1=33;** |  |  |  |  |  |  |  |  |  |  |  |  | **33** |
| **strcpy(x2, “abc~”);** | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **K** | **l** | **\0** |
| **최종 출력** | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** | **K** | **l** | **0** |

**= strcpy를 이용하면 문자열을 구분짓는 널값도 같이 복사가 된다. 따라서 미리 선언한 x2의 12바이트를 초과하여 13번째 칸에 널값을 입력하게된다. 이 과정에서 널값이 미리 초기화한 x1의 33값에 덮어씌워진다. 그래서 x1값이 0으로 출력되는 것이다.**

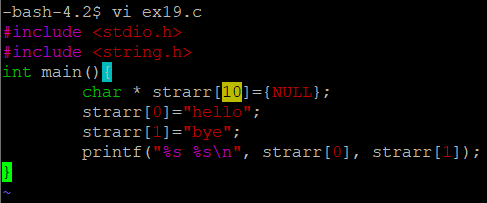
 

**= 이를 해결하기 위해 x2를 위한 공간을 1byte 더 추가하여 선언하였다. strcpy를 사용하면 결과 문자열은 입력한 문자수보다 +1된다는 사실을 항상 인지하고 있어야겠다.**

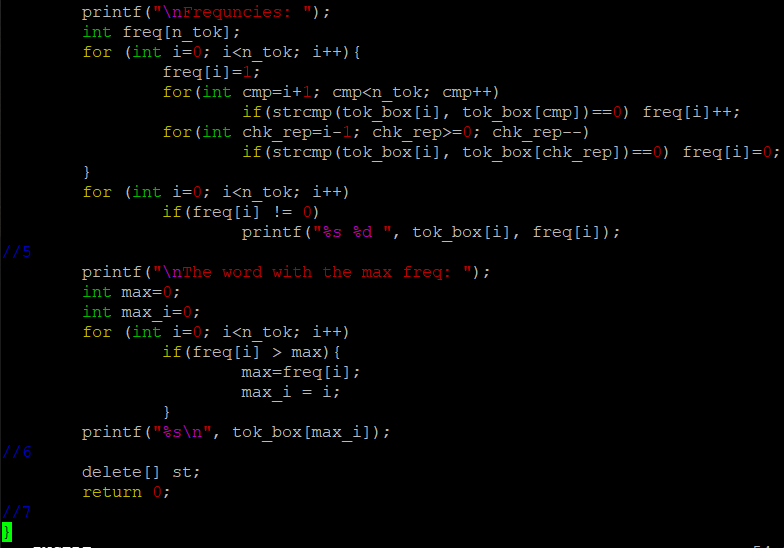
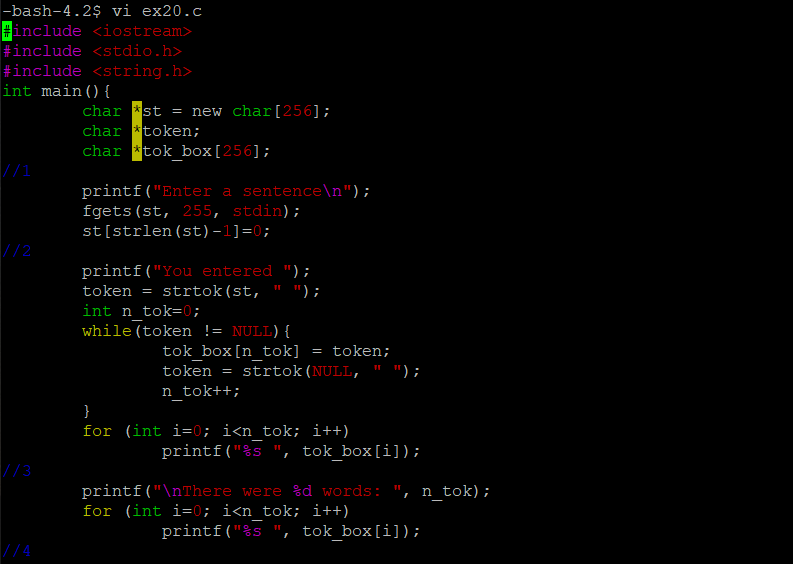
19) [You need memeory space for strcpy] What is wrong with the following program? How can you fix it?

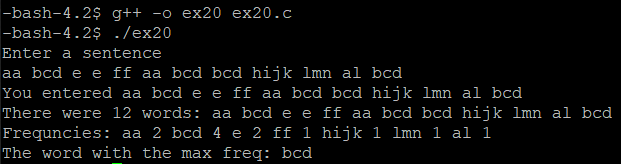
 

**= 주어진 코드를 실행하면 오류가 발생한다. 포인터 변수에 strcpy를 이용하여 문자열 copy를 시도했기 때문이다. 아래처럼 strcpy대신 strarr[1]에 직접적으로 “bye”의 주소값을 대입하면된다.**



20) [char pointer array, new, strcmp] Write a program that reads a long sentence and displays the frequency of each word as below. It also prints the word that has the maximum frequency.





**= phase0**

**st : 아래에서 fgets로 입력받은 문자열을 저장하는 공간이다. new로 할당한다.**

**token : strtok로 넘겨줄 token들을 가리키는 포인터 변수이다.**

**tok\_box : token들이 저장될 포인터 문자열이다.**

**= phase1**

**문자열을 fgets로 입력받아 st에 저장한다. strlen를 이용해 st의 마지막 enter 입력을 0값으로 바꾼다.**

**= phase2**

**strtok를 이용해 token들을 추출하여 이를 tok\_box가 가리키도록 지정한다. for문으로 tok\_box가 가리키는 단어들을 출력한다.**

**= phase3**

**phase2의 while문에서 count했던 n\_tok로 단어의 수를 출력한다. 이후 다시 for문으로 단어들을 출력한다.**

**= phase4**

**단어 수(n\_tok)에 맞춰 freq배열을 선언한다. for문에서 각 단어 순서에 맞게 freq배열에 빈도수를 넣어준다. for문은 현재칸 freq[i]를 1로 초기화하며 시작한다.**

**그 후 첫 for문에서는 현재 tok\_box[i]와 i보다 오른쪽에 있는 tok\_box를 strcmp로 비교한다. 일치하는 문자가 있을 때마다 freq[i]값을 1씩 추가한다.**

**두 번째 for문에서는 freq값의 중복을 제거한다. i보다 왼쪽에 있는 tok\_box를 모두 돌면서 현재 tok\_box[i]와 같은 값이 있으면 freq[i]를 0으로 초기화해준다.**

**두 개의 for문을 포함한 첫 for문이 종료되면 i를 0부터 단어 수까지 돌면서 freq[i]가 0이 아닌 i에 대해서만 tok\_box[i]를 출력하고 이에 따른 freq[i]를 출력한다.**

**= phase5**

**for문을 이용해 freq[i]값들 중 최빈값을 찾고 그에 맞는 tok\_box[i]값을 출력한다.**

**= phase6**

**동적 할당했던 st를 delete한 후 0을 리턴하여 코드를 종료한다.**