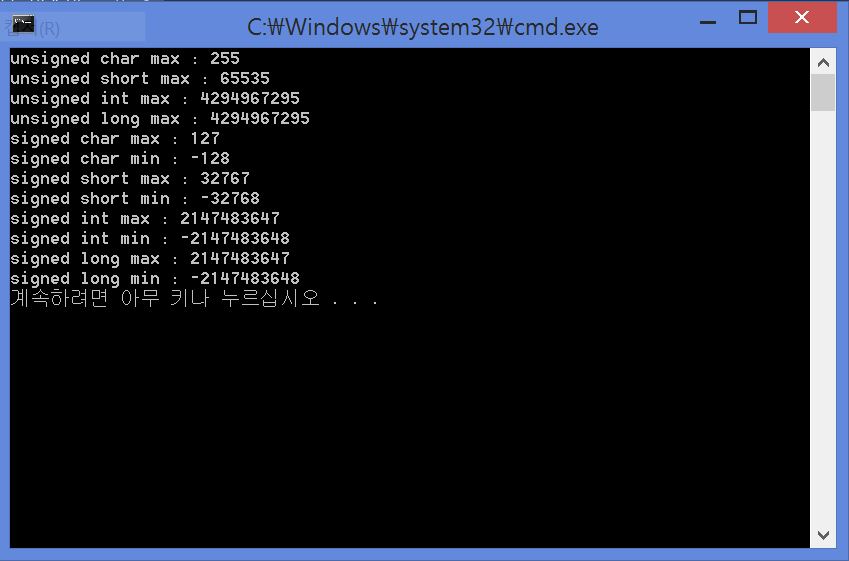
|  |  |
| --- | --- |
|  | **프로그래밍 개론Ⅰ**  **Lab04** |
| **분반** | **01** |
| **학번** | **1515655** |
| **이름** | **임소희** |
| **제출일** | **2015년 10월 8일** |

**Exercise 2-1**

**Write a program to determine the ranges of char, short, int, and long variables, both signed and unsigned, by printing appropriate values from standard headers. Harder if you compute them: determine the ranges of the various floating-point types.**

#실행 결과 (capture) 및 설명



Unsigned는 8bits 중 맨 앞에 부호 비트가 없어서 signed보다 수의 범위를 많이 포괄한다. 가장 큰 수는 1111 1111 일 때이므로 unsigned가 0이 아닐 때이다. 따라서 (unsigned char)~0을 적는다. 마찬가지로, short int long 타입도 (unsigned type)~0 형태로 적는다.

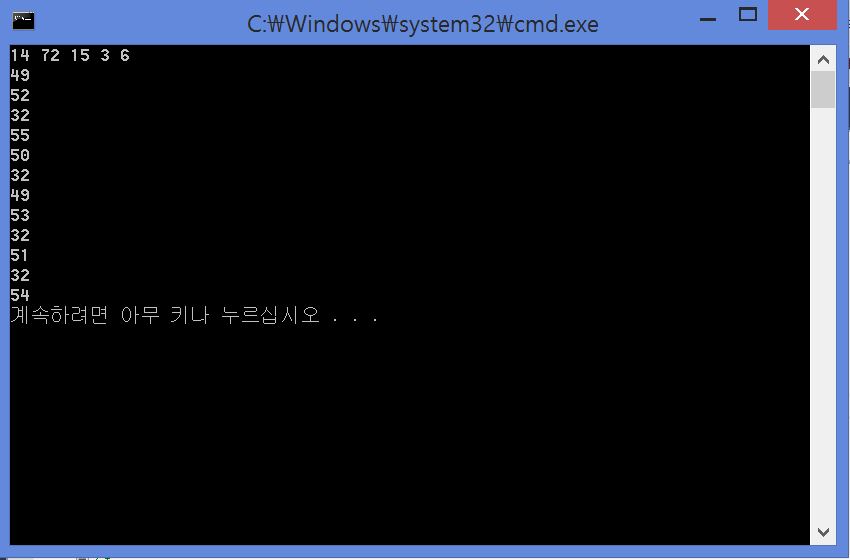
Signed는 8bits 중 맨 앞에 부호 비트가 있으므로 수의 범위가 적다. Unsigned를 이용하여 적으면 unsigned 비트들을 한 칸 뒤로 밀어서 signed처럼 만들어 준다. 위에서 구한 unsigned max타입을 한 칸 뒤로 밀면 signed max 값이 나온다. 즉, (char)((unsigned type)~0 >> 1) 형태로 최대값을 정한다.

최소값의 경우에는 signed max상태에서 +1을 하면 맨 앞의 비트가 1이되므로 음수가 된다. 따라서 -(type)((unsigned type)~0 >> 1)-1 형태로 적는다. –(-1)이되므로 +1이 되니까 더하기 1한 것과 같은 형태다.  
이들을 printf함수와 같이 사용하여 출력한다.

**Exercise 2-2**

**Write a loop equivalent to the for loop above without using && or ||.**

#실행 결과 (capture). 및 설명



입력한 문자(공백 포함)를 문자가 아닌 아스키코드에 해당하는 정수로 출력하는 프로그램이다.

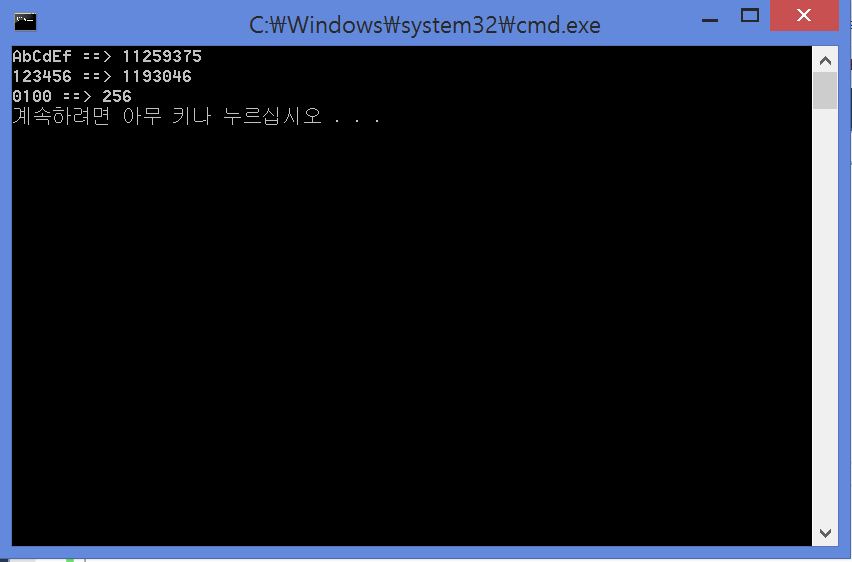
관계연산자를 사용하지 않고 for문과 중첩 if문을 사용하였다.

for문에 i의 초기값, 조건, 결과를 설정하고, 그 안에 if 문 문자 입력 시 엔터가 아닐 동안, 또 그 안에 if문을 넣어 실행종료가 되기까지 일을 수행(정수 출력)하도록 설정하였다. 엔터를 누르거나 실행종료가 되면 break를 써서 조건문을 빠져나오게 하였다.

**Exercise 2-3**

**Write a function htoi(s), which converts a string of hexadecimal digits into its equivalent integer value. The allowable digits are 0 through 9, a through f, and A through F.**

#실행 결과 (capture) 및 설명



16진수->10진수로 변환하는 프로그램이다.

16진수는 0~9, a~f, A~F까지 있으므로 조건을 각 각 따로 설정해야 한다.

우선 htoi라는 함수를 만들어야 하므로 int htoi(char s[])를 선언한다.

그 안에 int I, d를 선언하고 result값을 0으로 초기화한다.

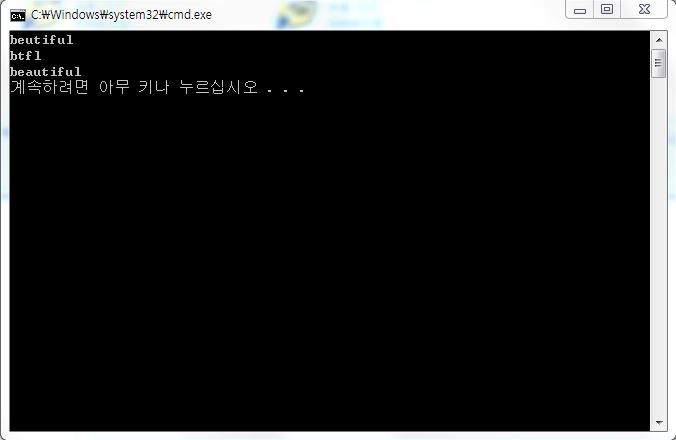
I가 문자열이 null문자를 만나기 전까지 실행하면서, 실행할 때마다 i++해주는 for문을 만든다.

그 속에 if문으로 각 각의 조건을 설정한다. 0~9일 때 s[i]-‘0’, a~f일 때 s[i]-‘a’+10, A~F일 때 s[i]-‘A’+10을 해주면 d는 문자가 아닌 정수로 바뀐다. 이렇게 나온 d를 16->10진수로 바꾸는 공식을 이용하여 결과를 출력해낸다. (result = result \* 16 + d)

**(숙제) Exercise 2-4**

**Write an alternative version of squeeze(s1,s2) that deletes each character in s1 that matches any character in the string s2.**

#실행 결과 (capture) 및 설명



문자열 s를 주었을 때, 또 다른 문자열 c가 포함되면 s에서 c를 지우는 프로그램이다.

이를 구하기 위해, squeeze(char s[], char c[])라는 함수를 만든다.

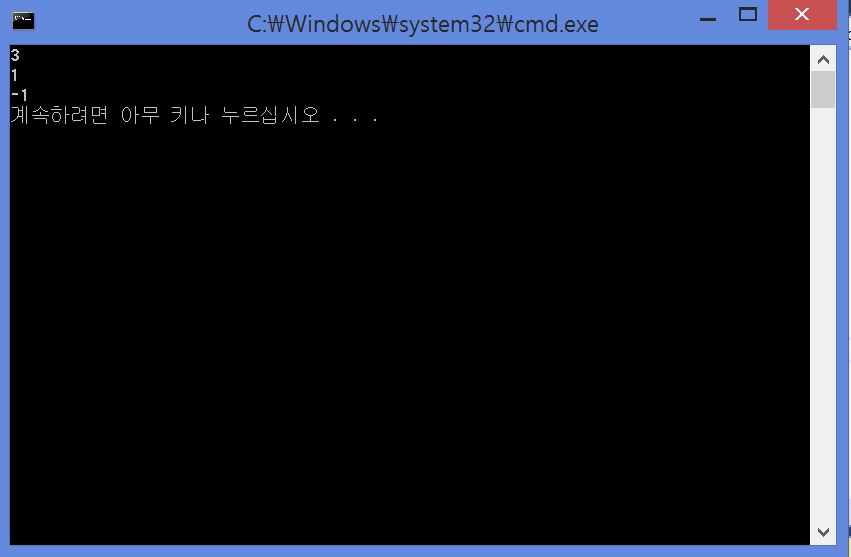
변수 I, j, k 세 개를 선언한다. (i는 s 문자열 개수, j는 c 문자열 개수이다. k는 문자 삭제 시, 뒤의 것을 앞으로 끌어올 때 쓰이며 총 개수는 i(최종 삭제된 결과)와 같게 된다.)

중첩 for문을 이용한다. i, j=0이고 s[i]와 c[j]가 null을 만나지 않을 때까지 I, j의 개수를 증가시킨다. 이 속에서 s[i]=c[j]인 경우(문자가 똑같은 경우)가 되면 삭제 해야 하므로, k=i이고 s[k]가 null을 만나지 않을 동안 증가시키는 for문을 만든다. 이 때 확인 순서가 한 칸 뒤로 밀려나므로 s[k]는 s[k+1](다음순서문자)이 된다. 이 for문을 빠져 나온 후 s[k]를 null문자로 비운다. 이 때 i의 총 개수는 문자 삭제로 인해 줄어든다.

**(숙제) Exercise 2-5**

**Write the function any(s1,s2), which returns the first location in a string s1 where any character from the string s2 occurs, or -1 if s1 contains no characters from s2. (The standard library function strpbrk does the same job but returns a pointer to the location.)**

#실행 결과 (capture) 및 설명



하나의 문자열(s1)이 있고, 또 다른 특정 문자열(s2)이 주어졌을 때, s2의 문자열 중 하나라도 s1에 존재할 시 존재하는 위치를 출력하고, 없을 시 -1을 출력하는 프로그램이다.

우선 이러한 계산을 할 함수 any를 선언한다. S1 s2는 각 각 문자를 비교해야 하므로 char로 선언한다. I, j를 선언하고, s1[i]가 null을 만나기 전까지 i를 증가시키는 for문을 만들고, j도 s2[j]가 위와 같은 조건일 때 증가하는 for문을 중첩하여 만든다. (조건 동시에 만족하도록) s1[i]=s2[j]가 같을 경우 그 위치를 출력하고, 같은 경우가 없으면 -1을 출력하도록 한다.

마지막으로 main에서 printf를 사용하여 이를 출력한다.