**컴퓨터과학과 C 프로그래밍 2 실습2**

- 실습 번호: lab2

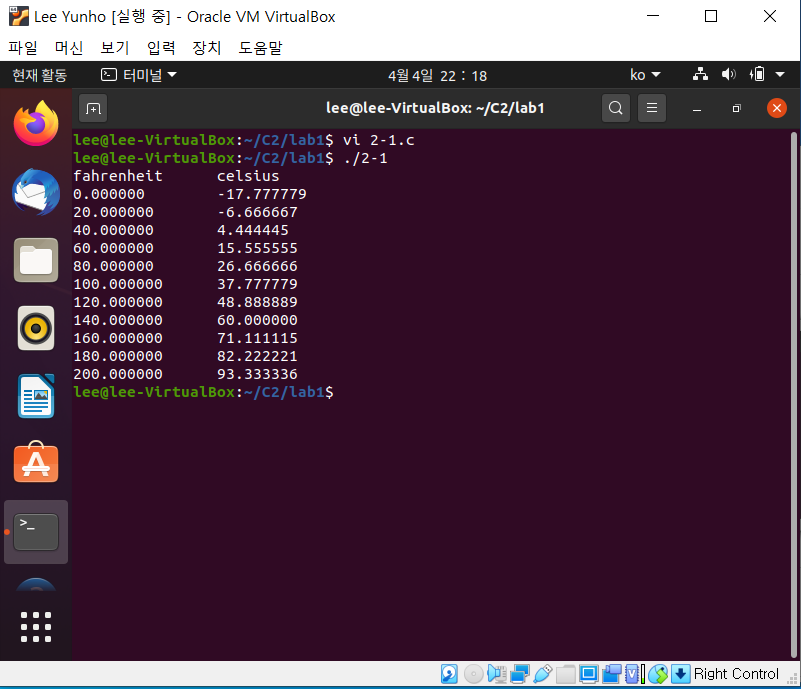
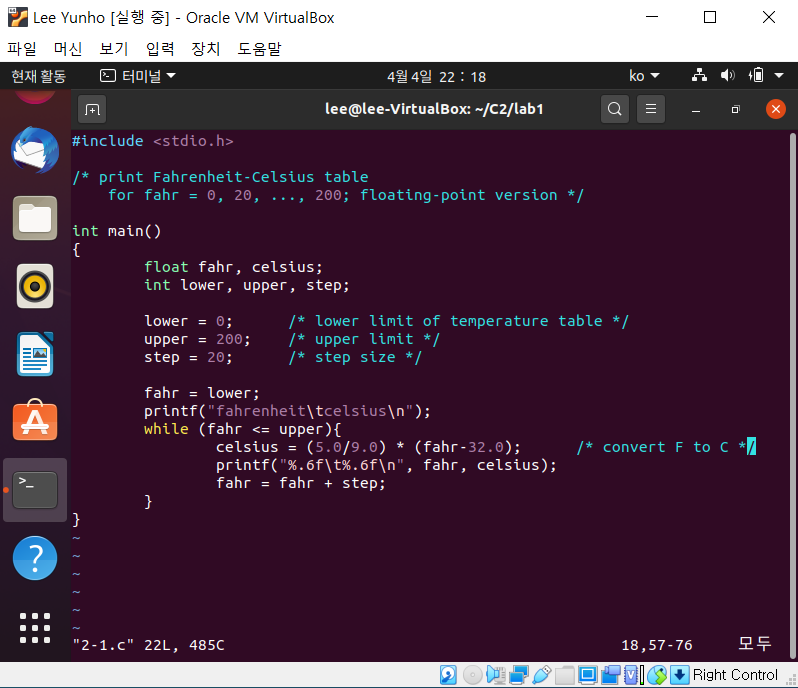
- 실습 디렉토리: ~CP2/lab2/ 이름: 이윤호\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- 실습 날짜: 2023 년 4 월 4 일 학번:202210957\_\_\_\_ 분반:3\_\_\_\_\_\_

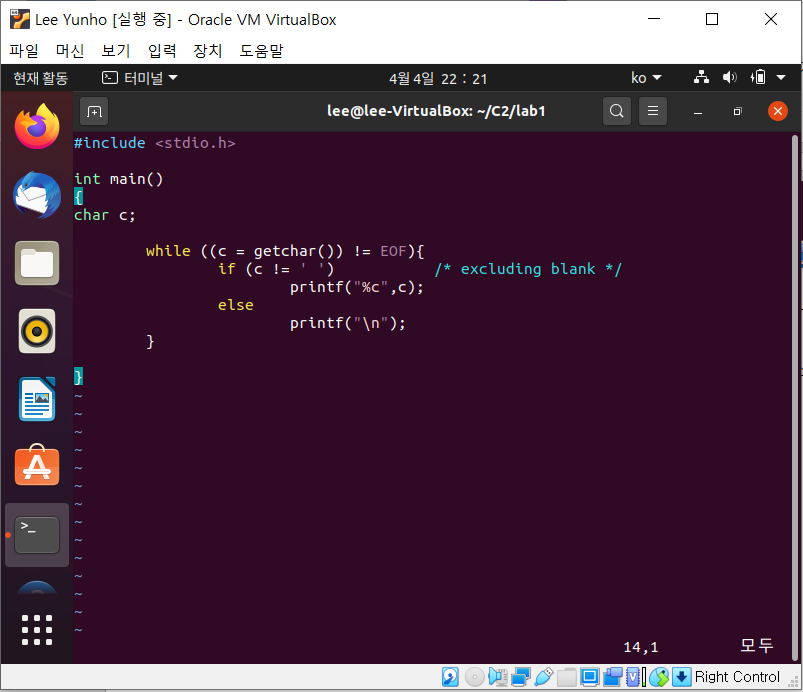
- 실습 제목: Tutorial, Types, Operators and Expressions

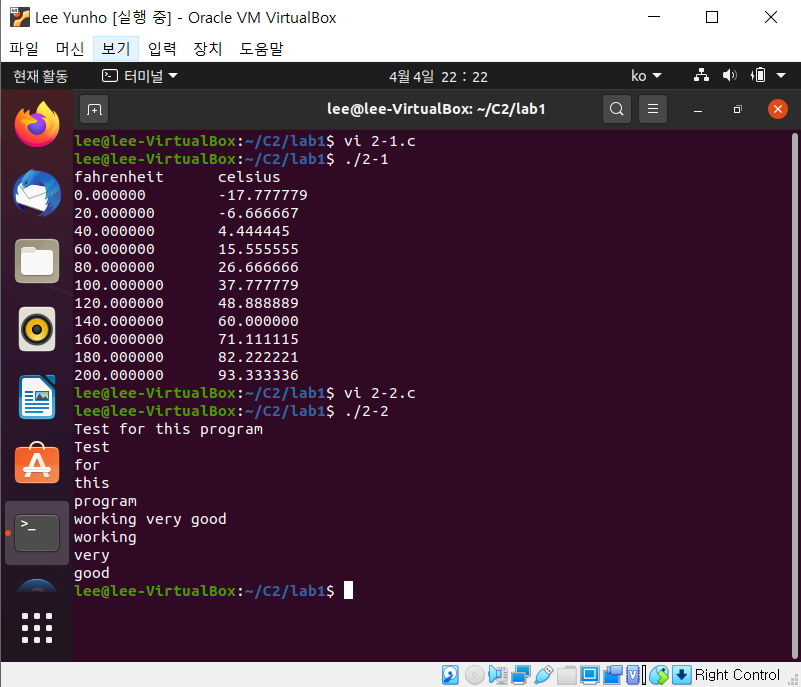
- 실습 내용:

1. (화씨-섭씨) CH1 slide 9쪽의 프로그램을 참고하여 다음과 같이 화씨 온도를 섭씨 온도로 바꾸어 출력하는 프로그램 2-1.c를 만들고 이를 $gcc -o 2-1 2-1.c로 컴파일하고 수행하여 그 결과를 보이시오. (소스코드에 적당한 comments를 적으시오, 소스코드 캡처, 결과 캡처를 넣으시오.) (2)

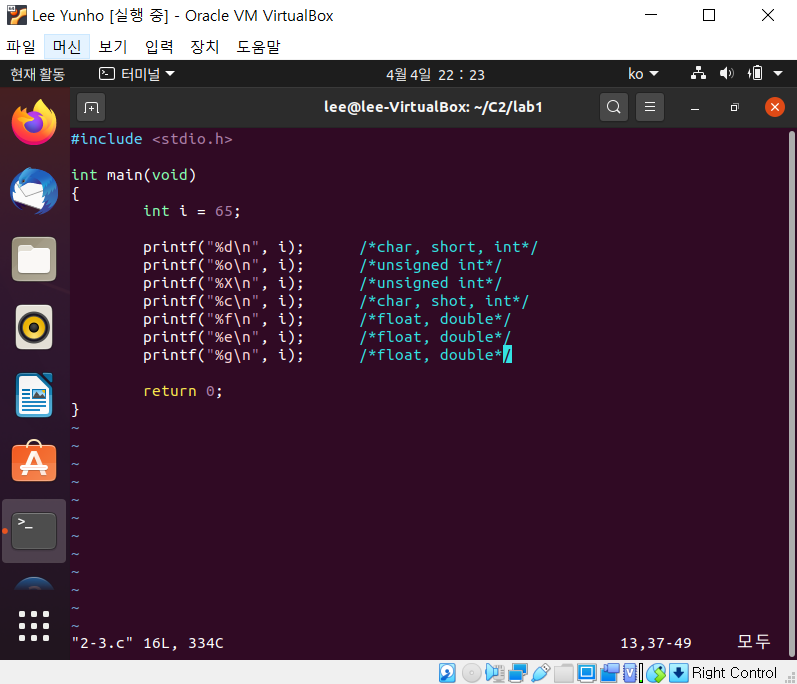


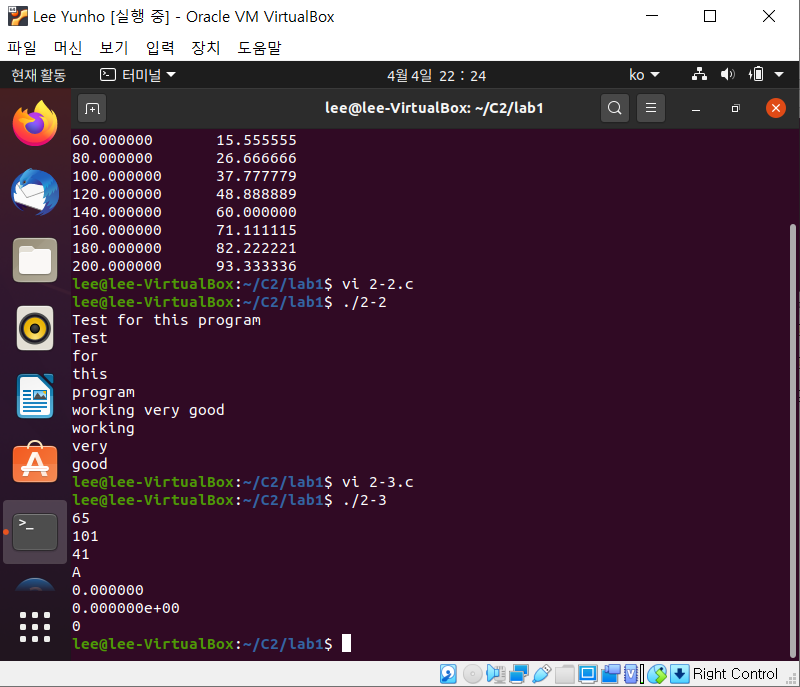
2. (word count program의 변형) CH1 slide 16족의 프로그램을 참고하여 그림과 같이 입력된 문자열들을 한 줄에 한 단어 씩 출력하는 프로그램 2-2.c를 만들고 이를 컴파일하고 수행하여 그 결과를 보이시오. (소스코드에 적당한 comments를 적으시오, 소스코드 캡처, 결과 캡처를 넣으시오.) (2)





3. (printf() conversion) 다음과 같은 프로그램 2-3.c를 작성, 컴파일, 수행하여 결과를 학인하시오. 일단 이 프로그램을 컴파일 하면 warning이 발생하는데, 그대로 수행하여 결과를 보이고, 왜 이러한 결과가 나오는지 이유를 적으시오. (소스코드에 적당한 comments를 적으시오, 소스코드 캡처, 결과 캡처를 넣으시오.) (2)



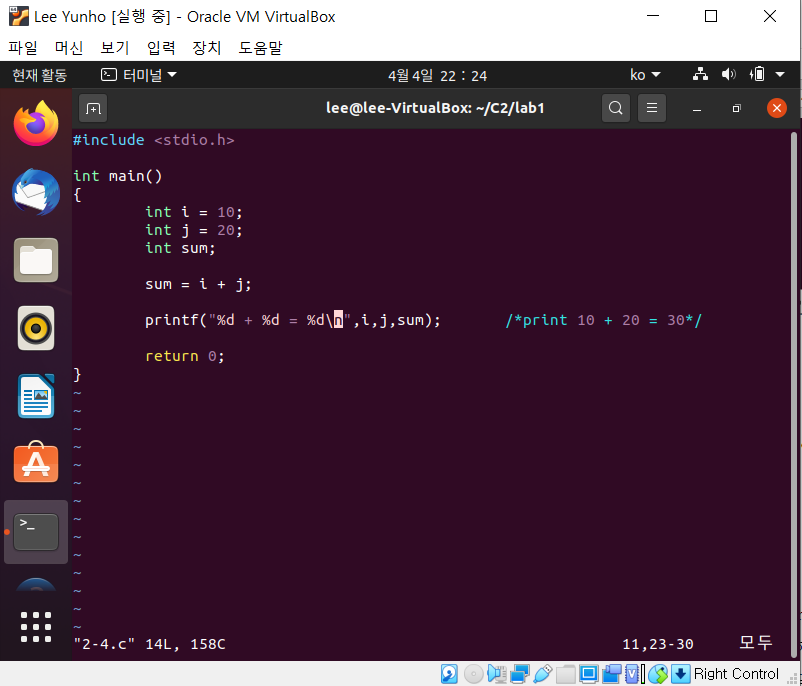


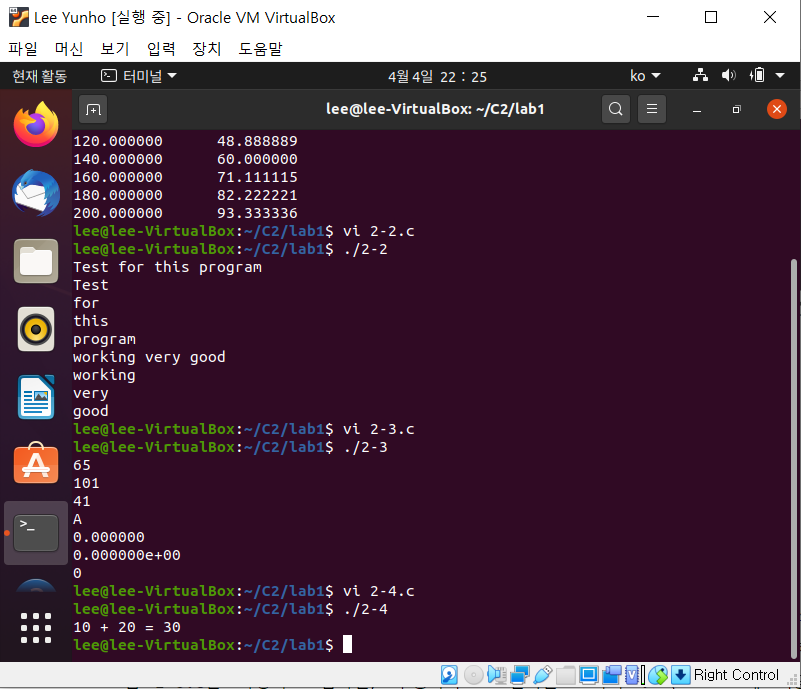
서식지정자 %f(float, double형)에 int형을 집어넣음

서식지정자 %e(float, double형)에 int형을 집어넣음

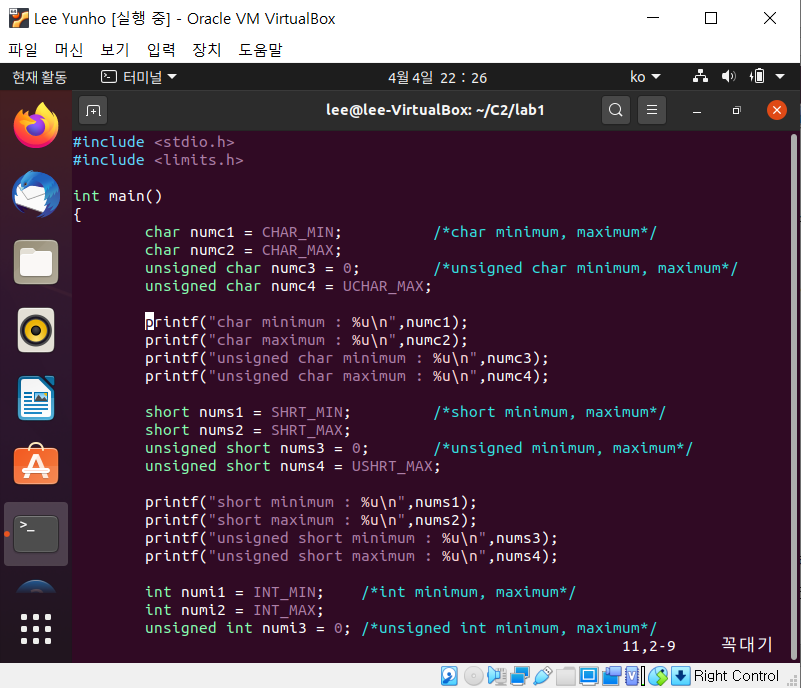
서식지정자 %g(float, double형)에 int형을 집어넣음

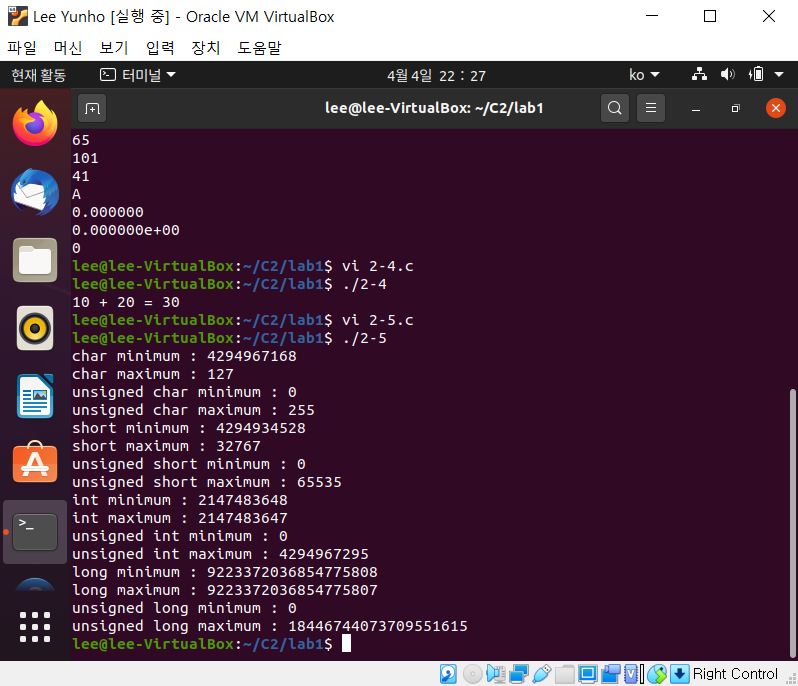
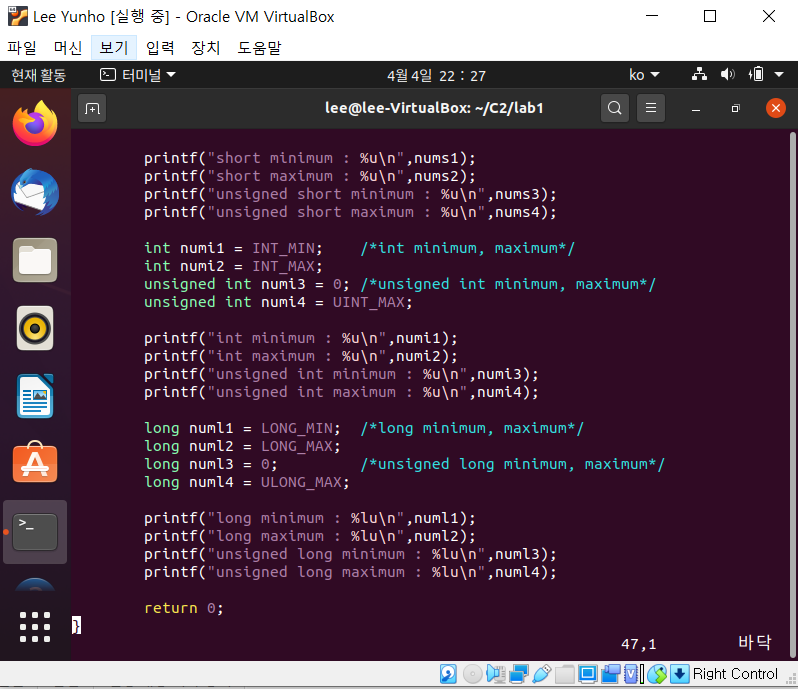
4. (integer add) 두 개의 int 변수 i와 j에 각각 10과 20을 저장하고 이를 더하여 int 변수 sum에 저장한 후 10 + 20 = 30의 형태로 출력해 내는 프로그램 2-4.c를 작성하고 컴파일, 수행하여 그 결과를 보이시오. (소스코드에 적당한 comments를 적으시오, 소스코드 캡처, 결과 캡처를 넣으시오.) (2)



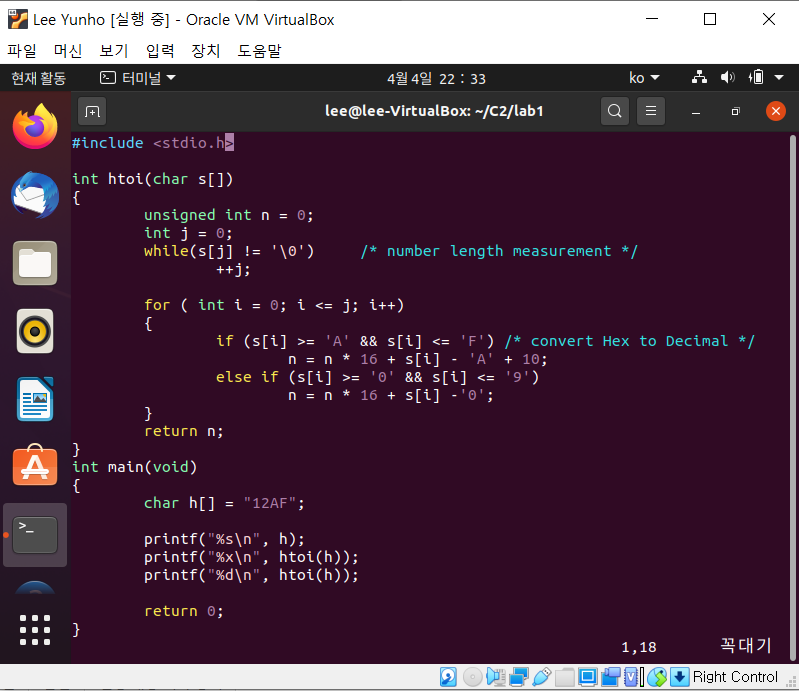


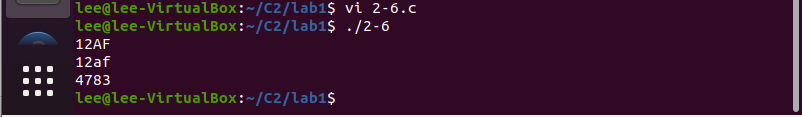
5. (limits.h) <limits.h> 헤더파일에 정의된 매크로를 이용하면 char, short, int, long이 각각 signed와 unsigned인 경우 최소값과 최대값을 쉽게 알 수 있다. 이를 출력하는 프로그램 2-5.c를 작성하고 컴파일, 수행하여 그 결과를 보이시오. (소스코드에 적당한 comments를 적으시오, 소스코드 캡처, 결과 캡처를 넣으시오.)(google에서 limits.h를 검색하여 참고하시오.)(2)



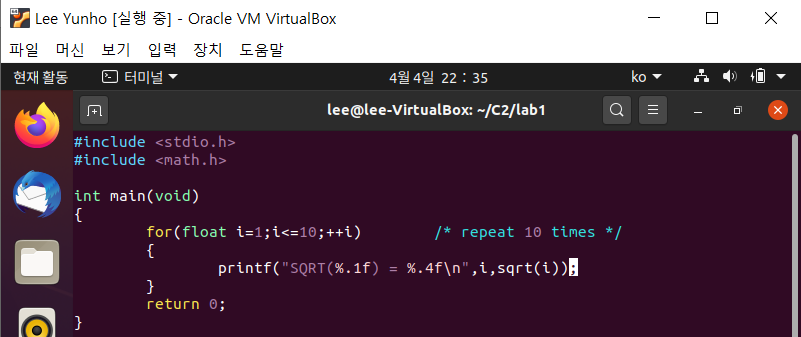


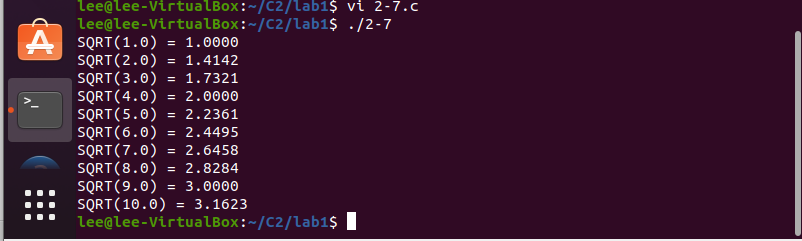
6. (htoi() 함수 작성) 16진 표기의 문자열을 integer로 바꾸어 주는 함수 int htoi(char s[]); 를 만들고 아래의 main() 과 연결되는 프로그램 2-6.c를 작성하고 컴파일, 수행하여 그 결과를 보이시오. (소스코드에 적당한 comments를 적으시오, 소스코드 캡처, 결과 캡처를 넣으시오.) (2)



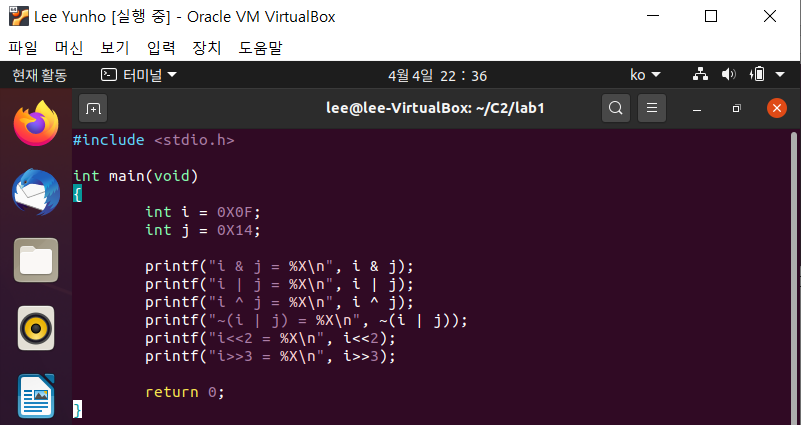


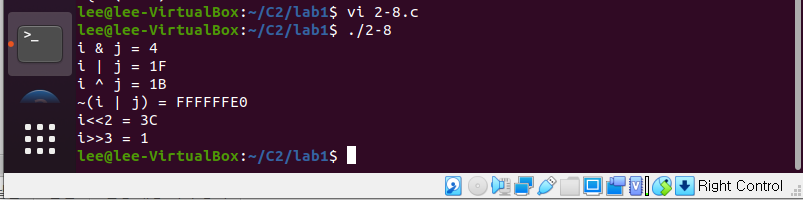
7. (sqrt() 활용) <math.h> 에서 정의된 sqrt() 함수와 반복문을 이용하여 다음과 같은 출력을 내는 프로그램 2-7.c를 작성하고 컴파일, 수행하여 그 결과를 보이시오. (소스코드에 적당한 comments를 적으시오, 소스코드 캡처, 결과 캡처를 넣으시오.) (2)





8. (비트연산자) 다음과 같은 프로그램 2-8.c를 만들고 컴파일, 수행하여 결과를 보이시오.





그리고 왜 그런 결과가 나오는지 설명하시오.

0X0F를 2진수로 변환하면 00001111,

0X14를 2진수로 변환하면 00010100이다.

&는 비트 연산자 AND를 나타내는데, 두 개의 비트가 모두 1일 때 1을 반환하므로 00000100,16진수로 4이다.

|는 비트 연산자 OR을 나타내는데, 두 개의 비트 중 하나라도 1이면 1을 반환하므로 00011111, 16진수로 1F이다.

^는 비트 연산자 XOR을 나타내는데, 두 개의 비트가 서로 다른 경우에 1을 반환하므로 000011011, 16진수로 1B이다.

~는 비트 연산자 NOT을 나타내는데, 비트를 0에서 1로, 1에서 0으로 반전시킨다.

i | j의 값을 반전시키면 11100000이므로, FFFFFFE0이 나온다.

<<는 <<의 비트 열을 << 우측에 있는 수만큼 왼쪽으로 이동시키므로 00111100, 16진수로 3C이다.

>>는 >>의 비트 열을 >> 우측에 있는 수만큼 오른쪽으로 이동시키므로 00000001, 16진수로 1이다.

(결과 캡처를 넣으시오.) (2)

끝