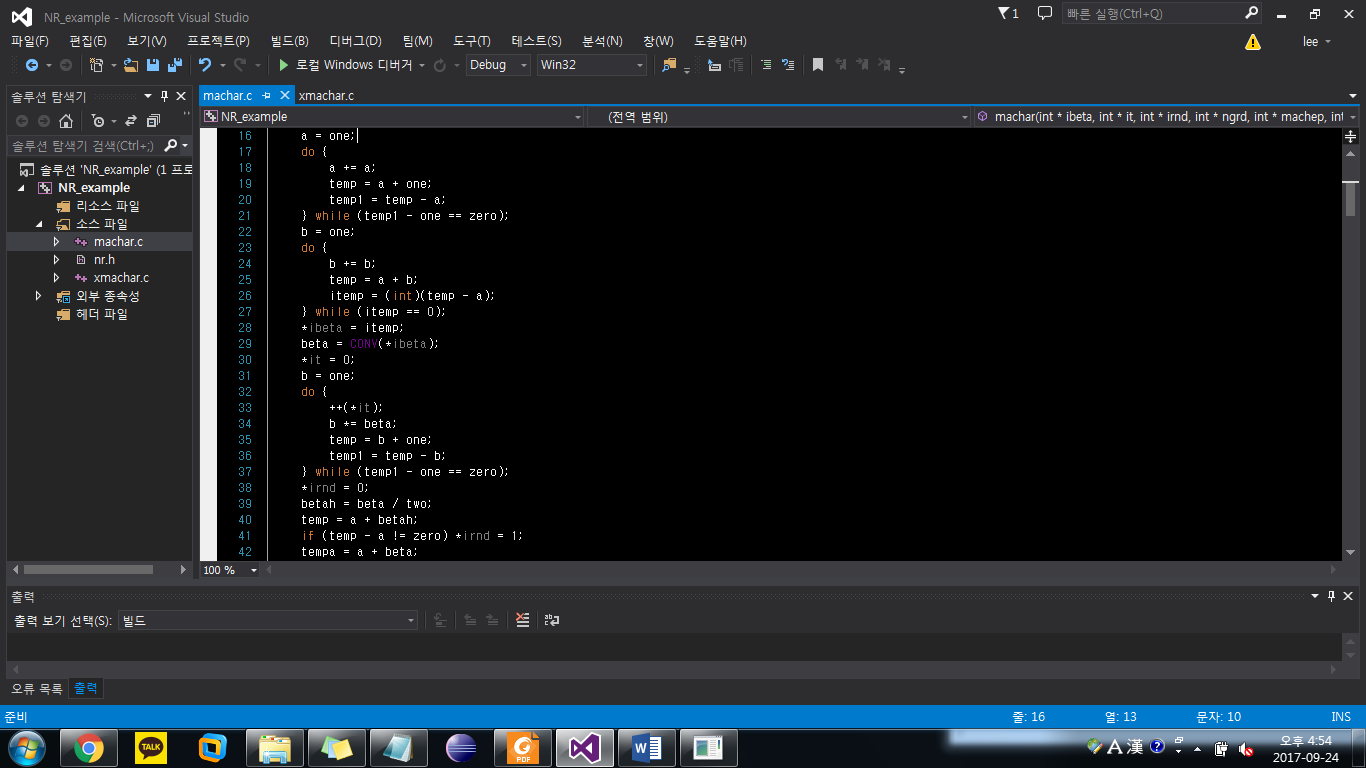
수치해석 hw1 report

2013012148 이재일

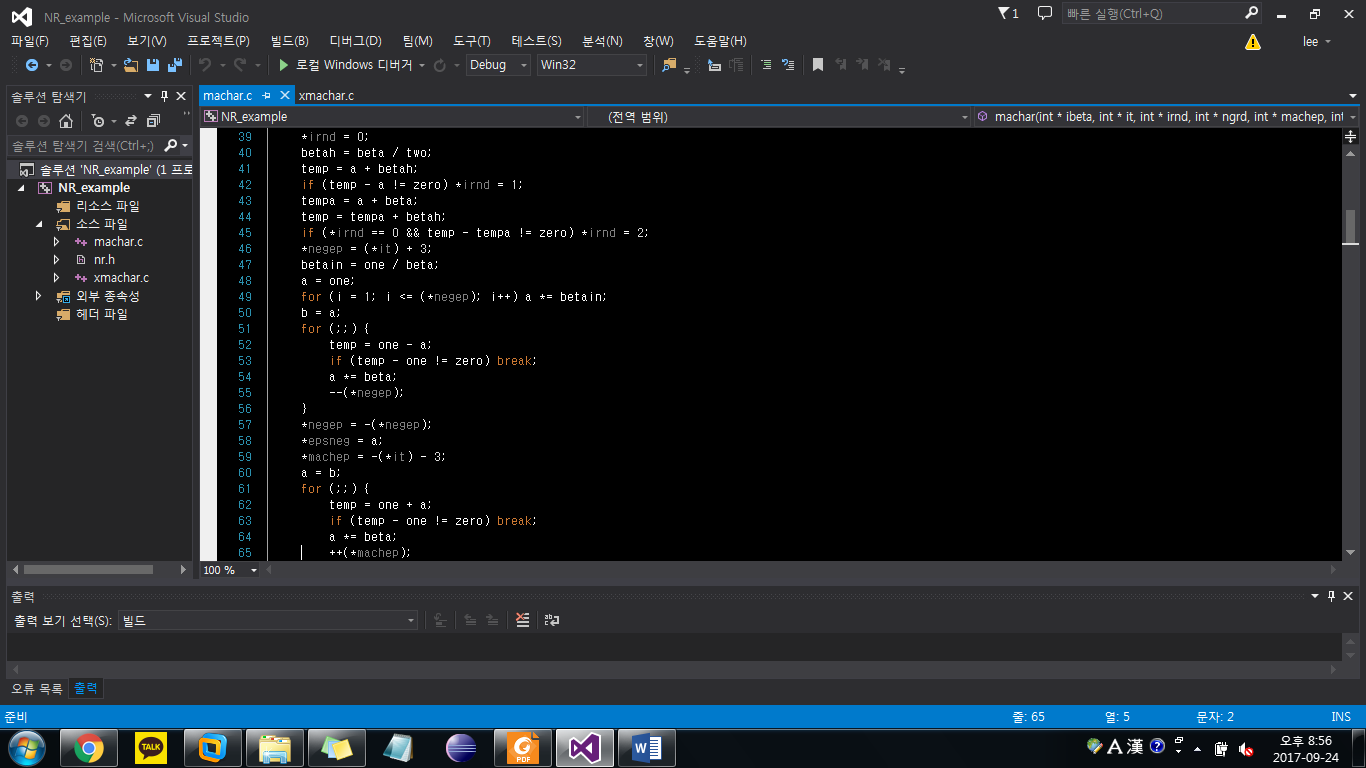
<method1>

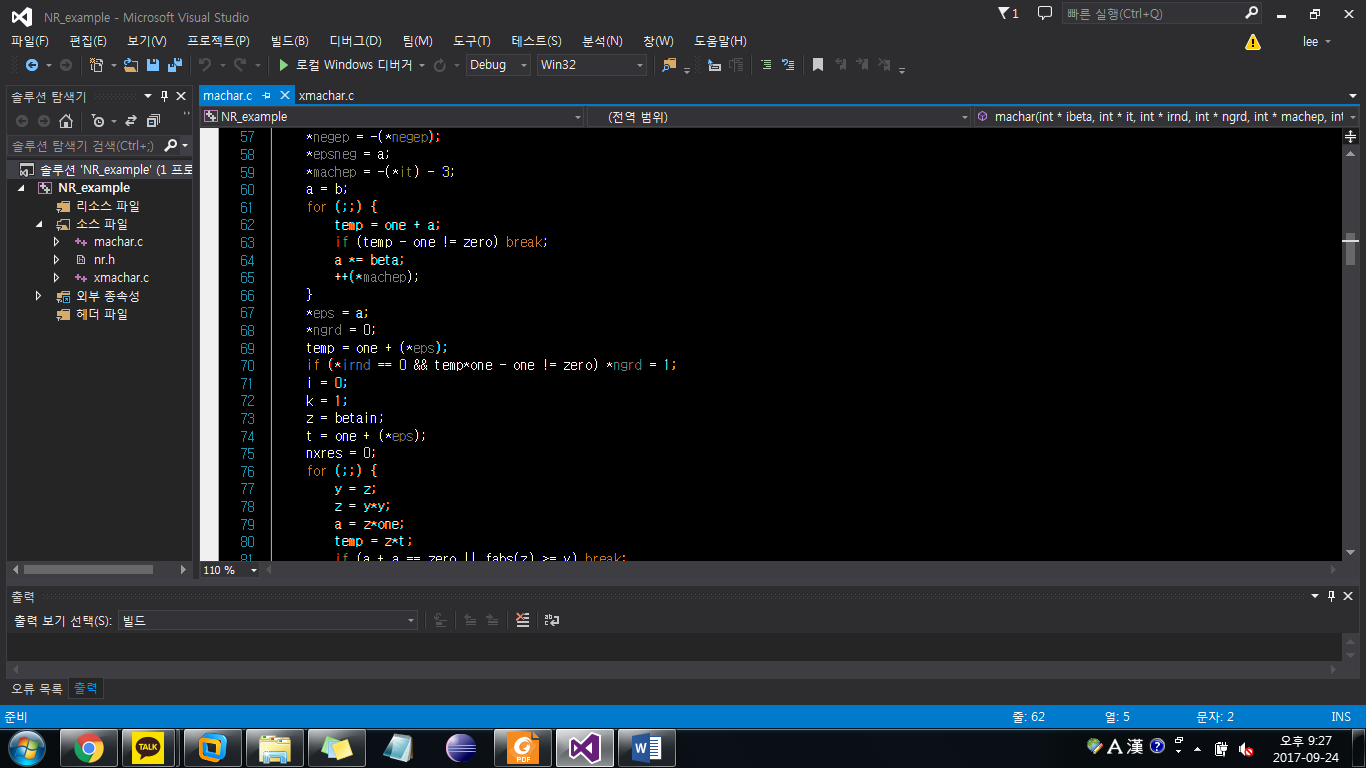


첫번째 반복문(17)으로 a+one == a가 되는 case일 때 break된다(a+one과 a의 대표값이 같은 case).

두번째 반복문으로(23)으로 a와 b를 더한 값에 a를 뺄 때 그 차이인 b(2)가 출력되는지 check한다. 해당 연산은 정상적으로 진행되어서 itemp값이 2가 되어서 반복문을 빠져나온다. -> ibeta값 2가 된다. 즉 float형에서 a크기일 때 1의 차이는 구분하지 않지만, 2의 차이는 구분이 된다.

세번째 반복문(32)으로 b와 b+one의 차이를 구분하지 않는 b값을 찾는다. 이때 it(곱셈 횟수)의 출력이 24가 나온다. 즉 b가 2^24일 때 2^24와 2^24+1을 구분하지 않고 같은 값으로 인식한다





negep의 결과값이 -24, machep의 결과값이 -23이 나왔다. 즉 float형은 value의 최소 단위로 음의 방향으로는 2^-24부터 양의 방향으로는 2^-23부터 구분을 한다는 뜻이다.

iexp가 8, minexp로 -126, maxexp로 128이 나왔다. float형의 지수부가 8비트이고 -126~128까지 표현됨을 알 수 있다.

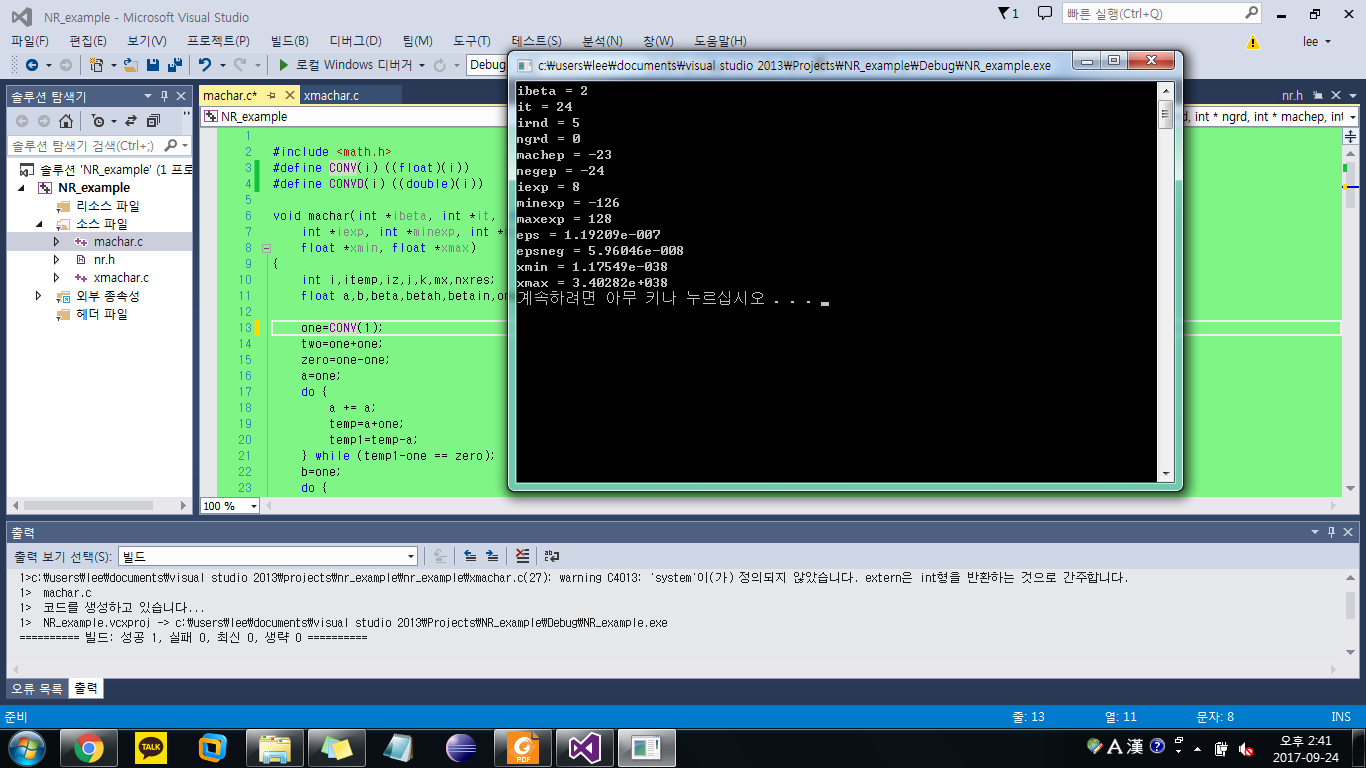
eps가 1.19209e-007, espneg가 5.96046e-008이 나왔다. 양수쪽 epsilon(간격, machine epsilon)이 eps이고 음수 방향 epsilon이 espneg임을 알 수 있다.

xmin이 1.17549e-038, xmax 3.40282e+038이 나왔다. Min Normal이 xmin이고 Max Normal이 xmax임을 알 수 있다.

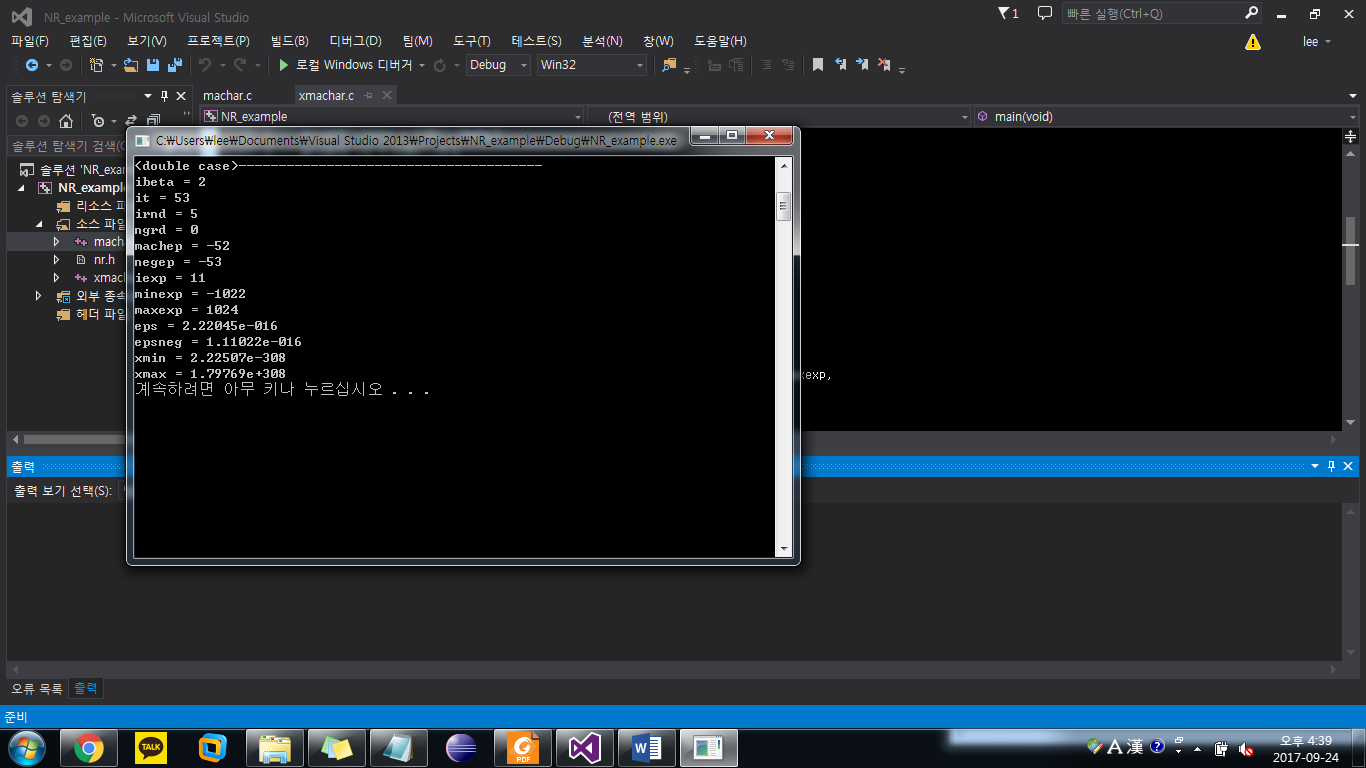
->double형도 마찬가지로 check한다.

<실행결과>

float case



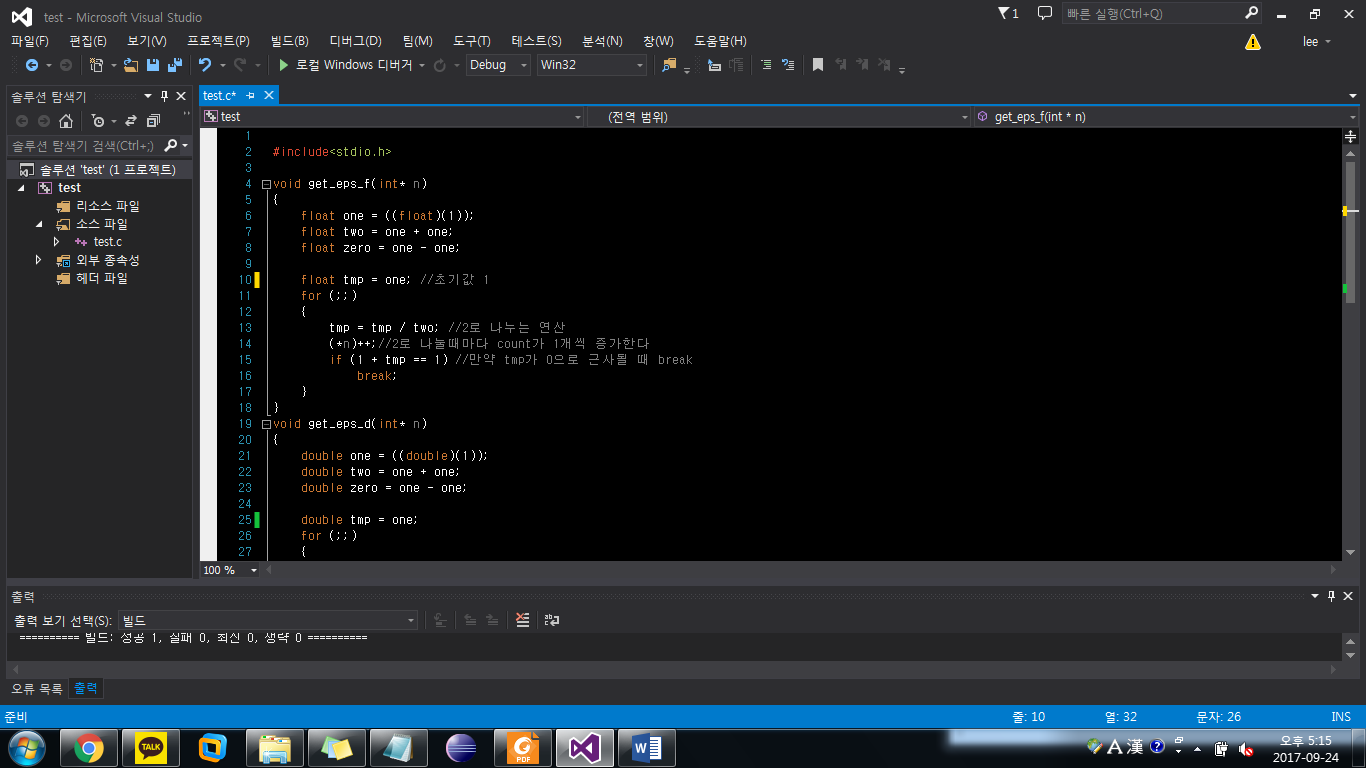
double case



<method2>

get\_eps\_f함수와 get\_eps\_d함수를 생성한다.

get\_eps\_f함수 내부.



get\_eps\_d함수는 get\_eps\_f에서 float형만 double로 바꾼 형태이다

<실행결과>

float형은 n이 24일 때, double형은 n이 53일 때

1 + 2^-n = 1을 만족한다

