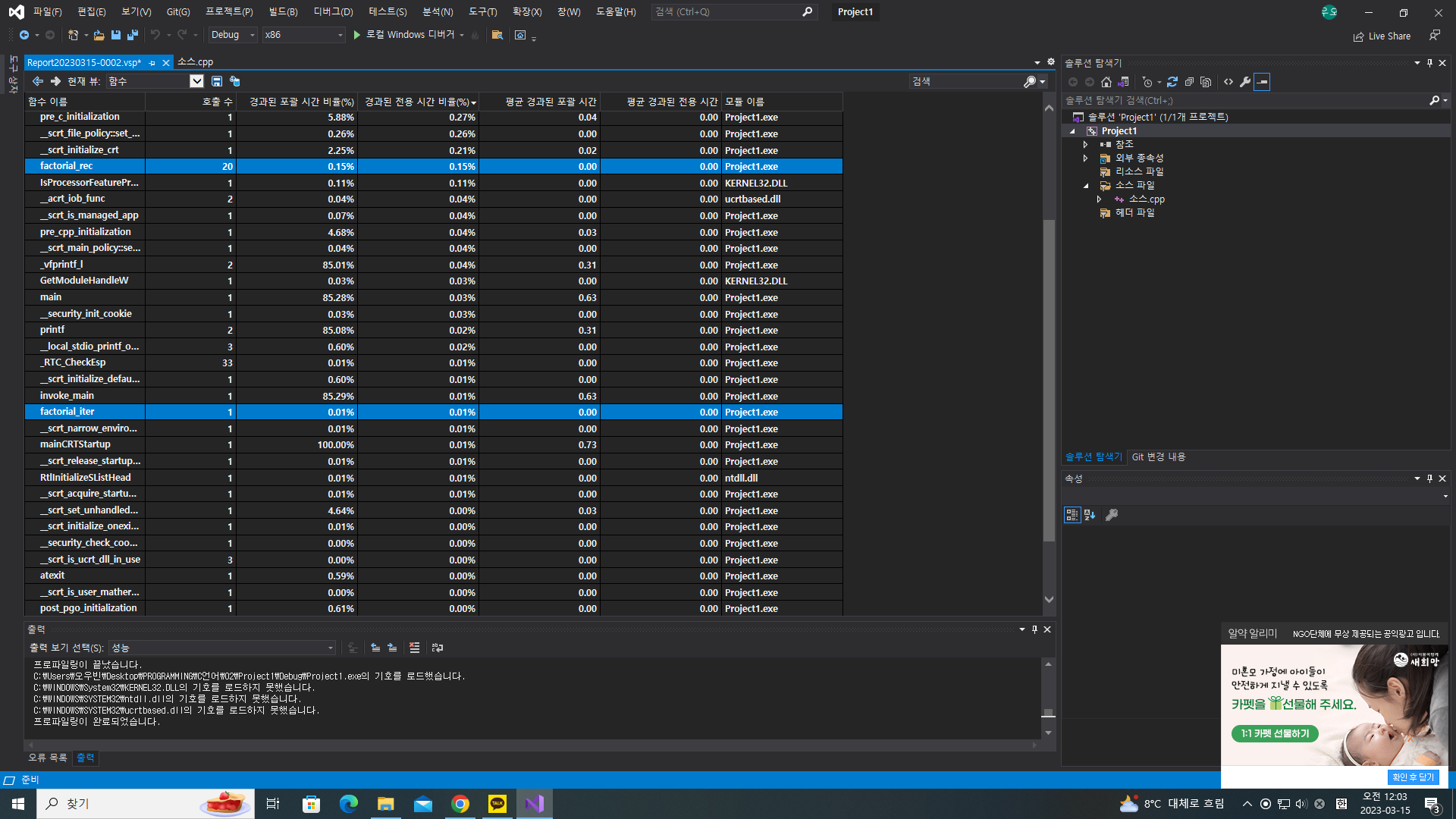
텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명



실행 시 위와 같은 결과 값이 나왔습니다.

같은 값이지만 ‘Iterative factorial’에서는 ‘factorial\_iter’의 함수 자체는 한번만 돌면서 안에 있는 for문 안에서 1부터 20까지의 수가 내림차순으로 반복되어 곱해져 20!의 값이 나왔습니다.

이 말을 식으로 풀어쓰면 20 \* 19 \* 18 \* … 2 \* 1이 됩니다.

반면 ‘Recursive factorial’에서는 ‘factorial\_rec’의 함수 자기 자신이 n이 1이 될때까지 n이 1씩 줄면서 총 20바퀴를 돌아 결과 값을 만들었습니다. n이 1이 되면 계속되어 반복되는 것을 방지하기 위해 1의 return값을 주었습니다.

이 말을 식으로 풀어쓰면 20 \* factorial\_rec(19)

19 \* factorial\_rec(18)

18 …

2 \* 1이 됩니다.

이 두 방법을 비교하여 봄으로써 호출 수 및 경과된 전용 시간 비율이 함수 안의 반복문으로 반복하는 ‘Iterative factorial’에 비해 함수 자신을 반복하는 ‘Recursive factorial’ 방법이 상당히 높다는 것을 알 수 있었고 ‘Iterative factorial’의 방법이 더욱 효율적이라는 것을 알게 되었습니다.