컴퓨터보안 RSA 과제 보고서

학과: 컴퓨터공학과

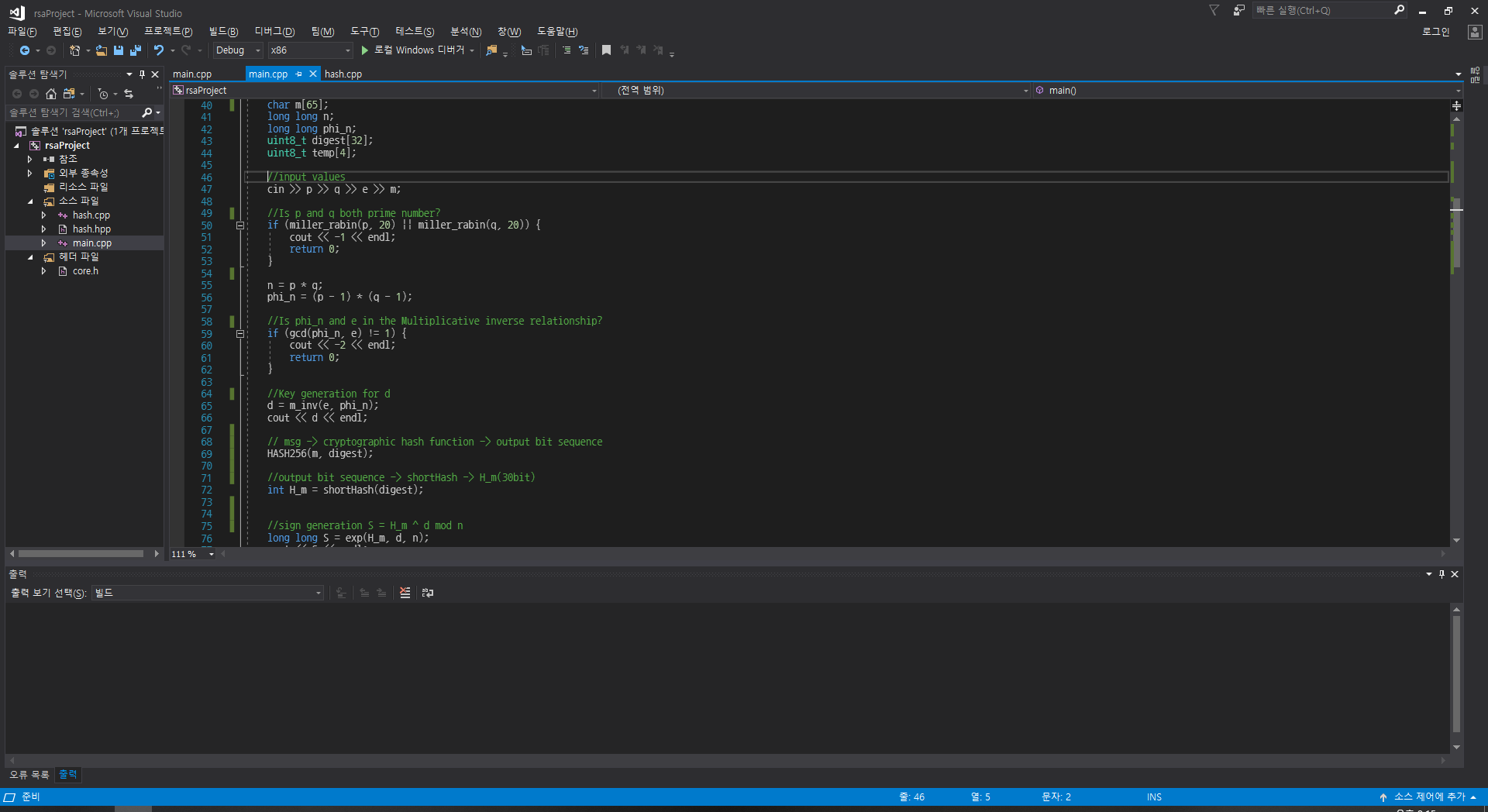
학번: 12151378

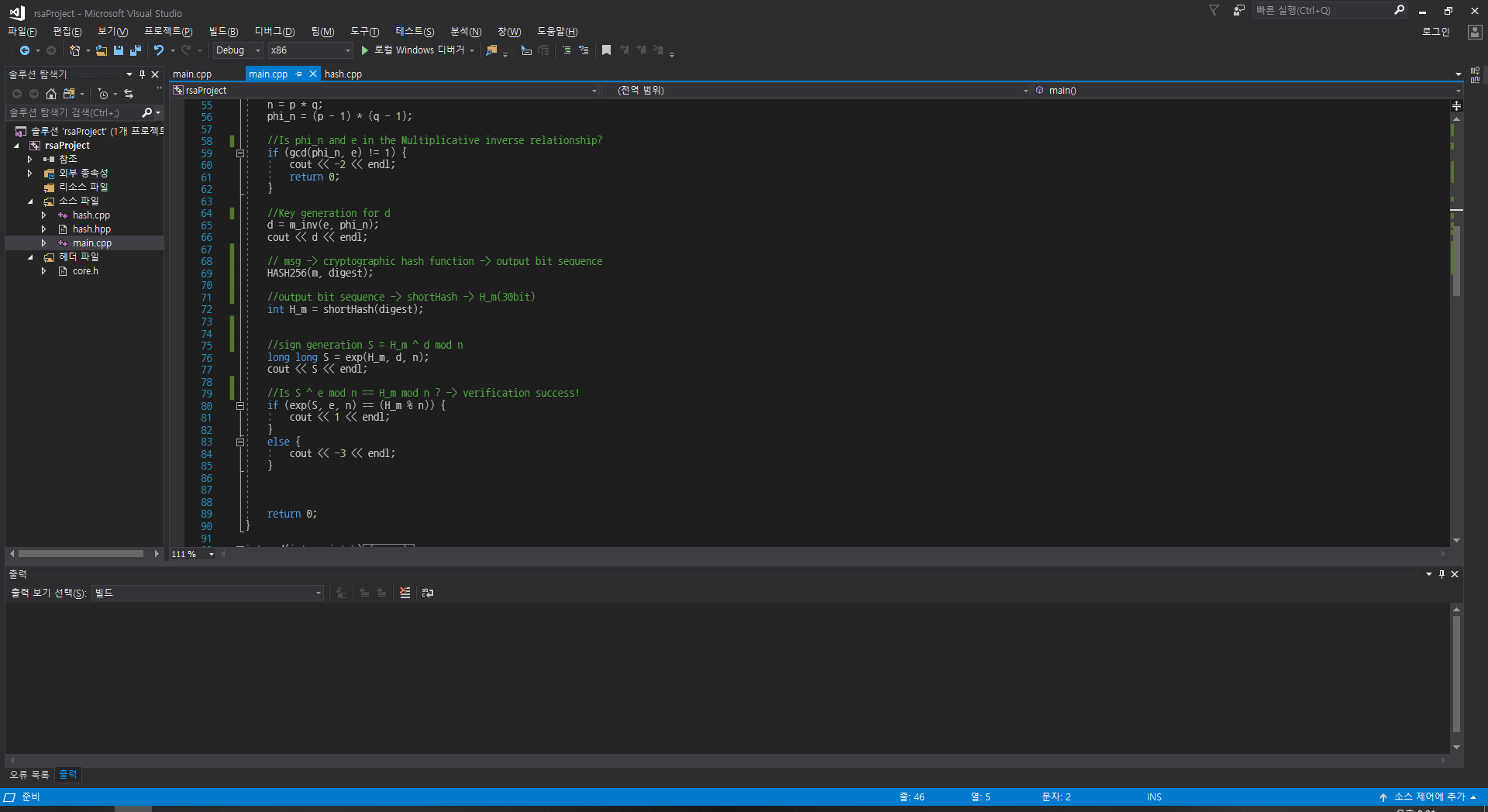
이름: 남호준

**1.프로그램 설명**

이론 수업에서 배운 RSA system을 구현. p, q와 e(공개키), m(메시지)를 입력받아 서명/확인.

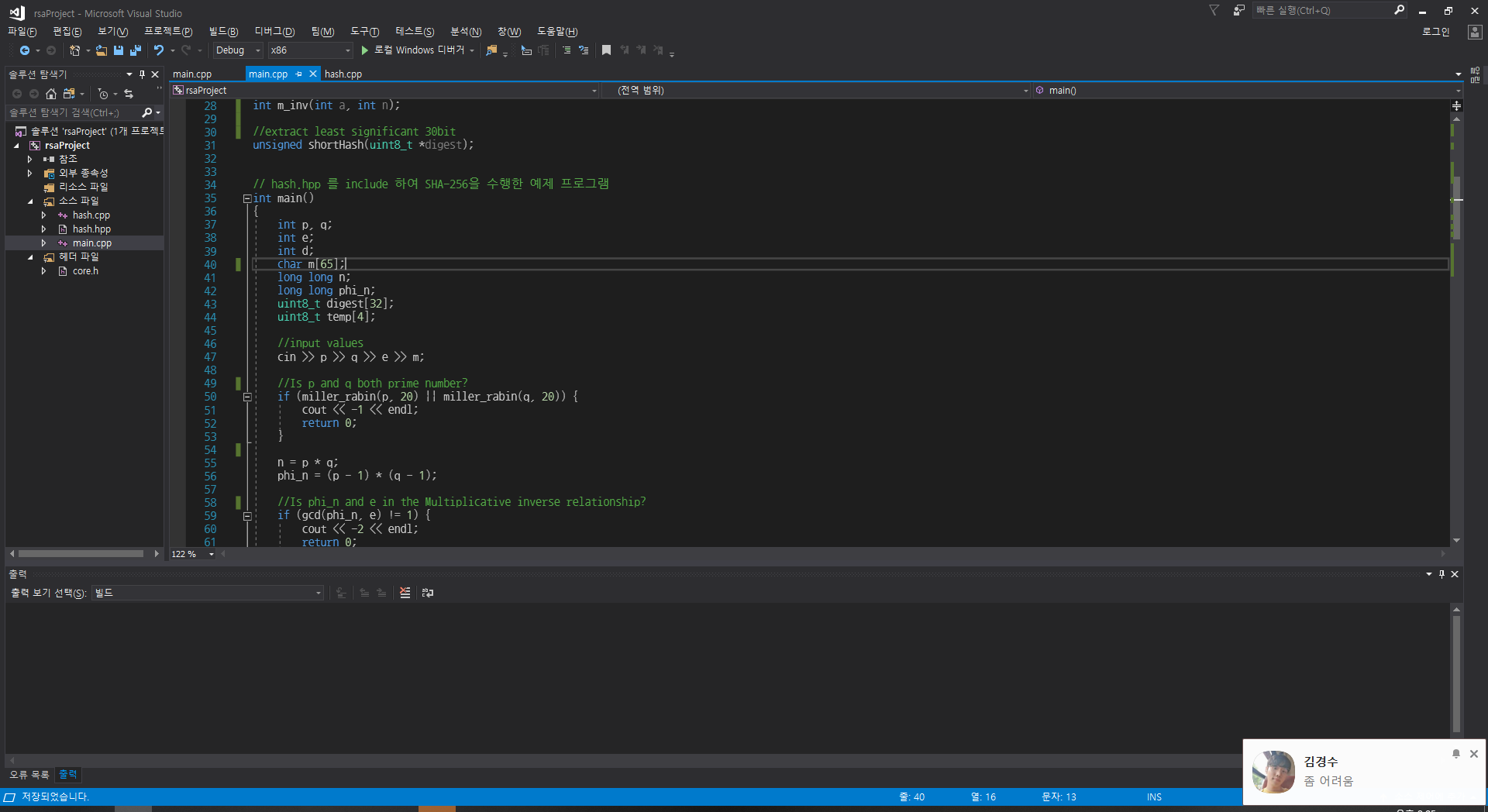
a) main 함수 설명





1. p, q, e, m을 사용자에게 입력 받습니다.
2. miller\_rabin 테스트로 입력받은 p, q가 소수인지 Test합니다. 둘중 하나라도 아니면 -1출력, 종료
3. 두 수 모두 소수로 판별되면, n과 phi\_n을 계산해 변수에 할당해줍니다.
4. gcd함수로 phi\_n과 e가 역원관계(gcd=1)가 아니면 -2를 출력, 종료
5. e와 phi\_n의 곱의 역원을 구해(extened\_euclid) d에 할당
6. HASH256을 사용해 메시지를 digest에 저장
7. shortHash함수로 digest의 최하위 30비트 추출
8. H\_m ^ d mod n -> S 생성
9. S^e mod n = H\_m mod n 확인, 같으면 인증성공(1 출력후 종료)

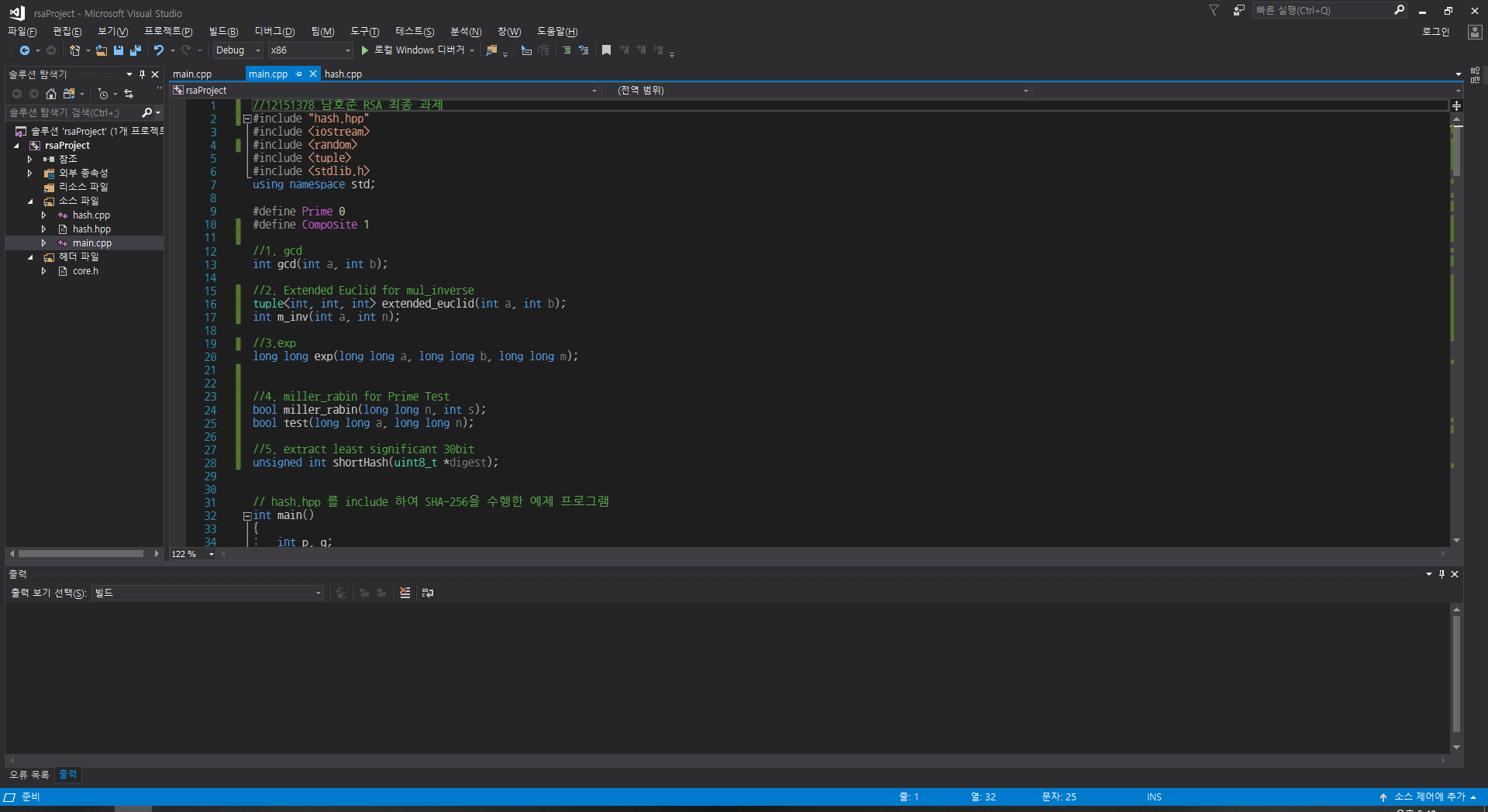
b) 사용된 변수



\* temp 배열은 Hash함수를 거친 digest에서 최하위 30비트를 추출한 output.

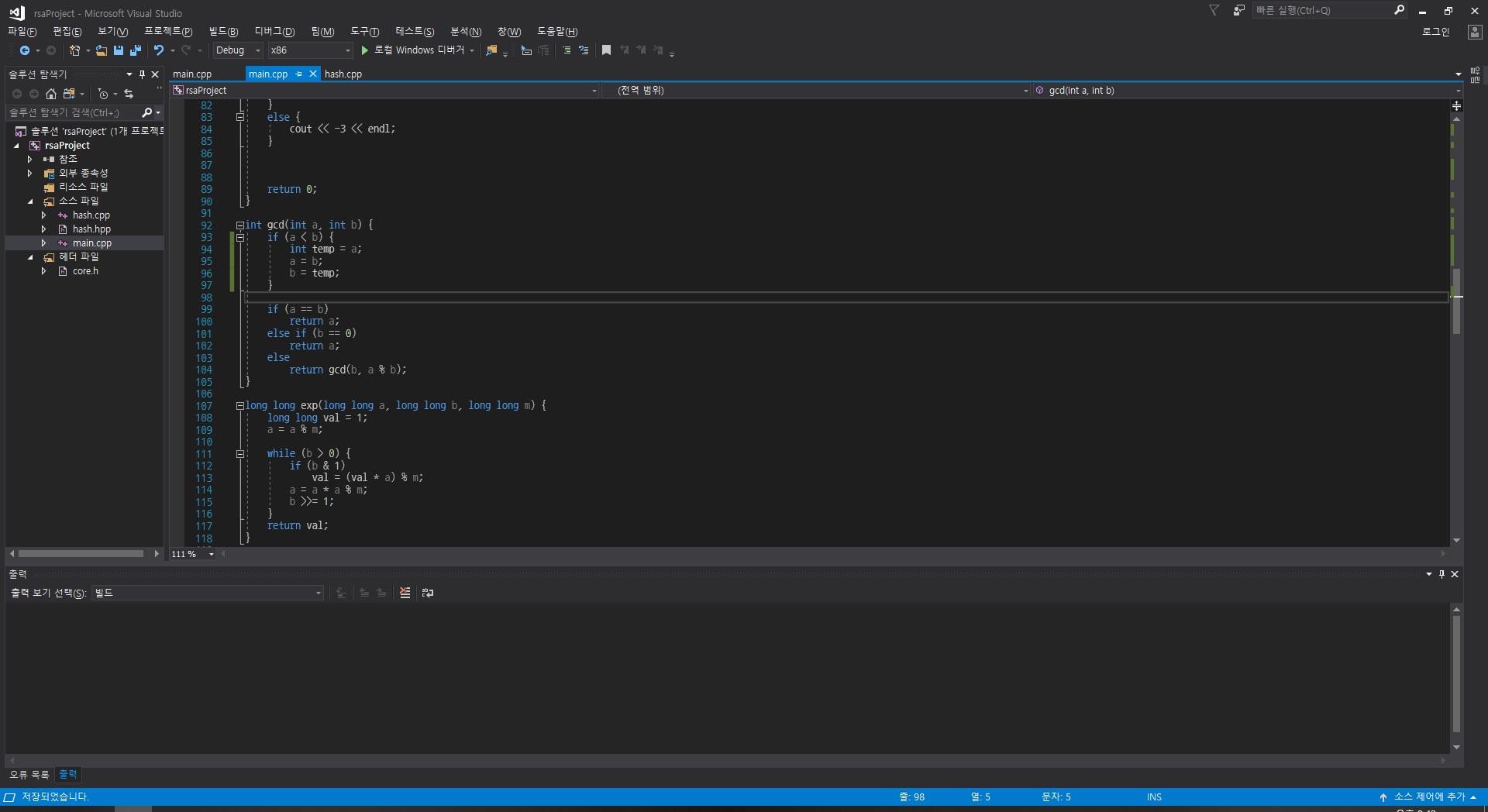
\* Prime = 0, Composite = 1 상수 선언

c) 사용된 함수



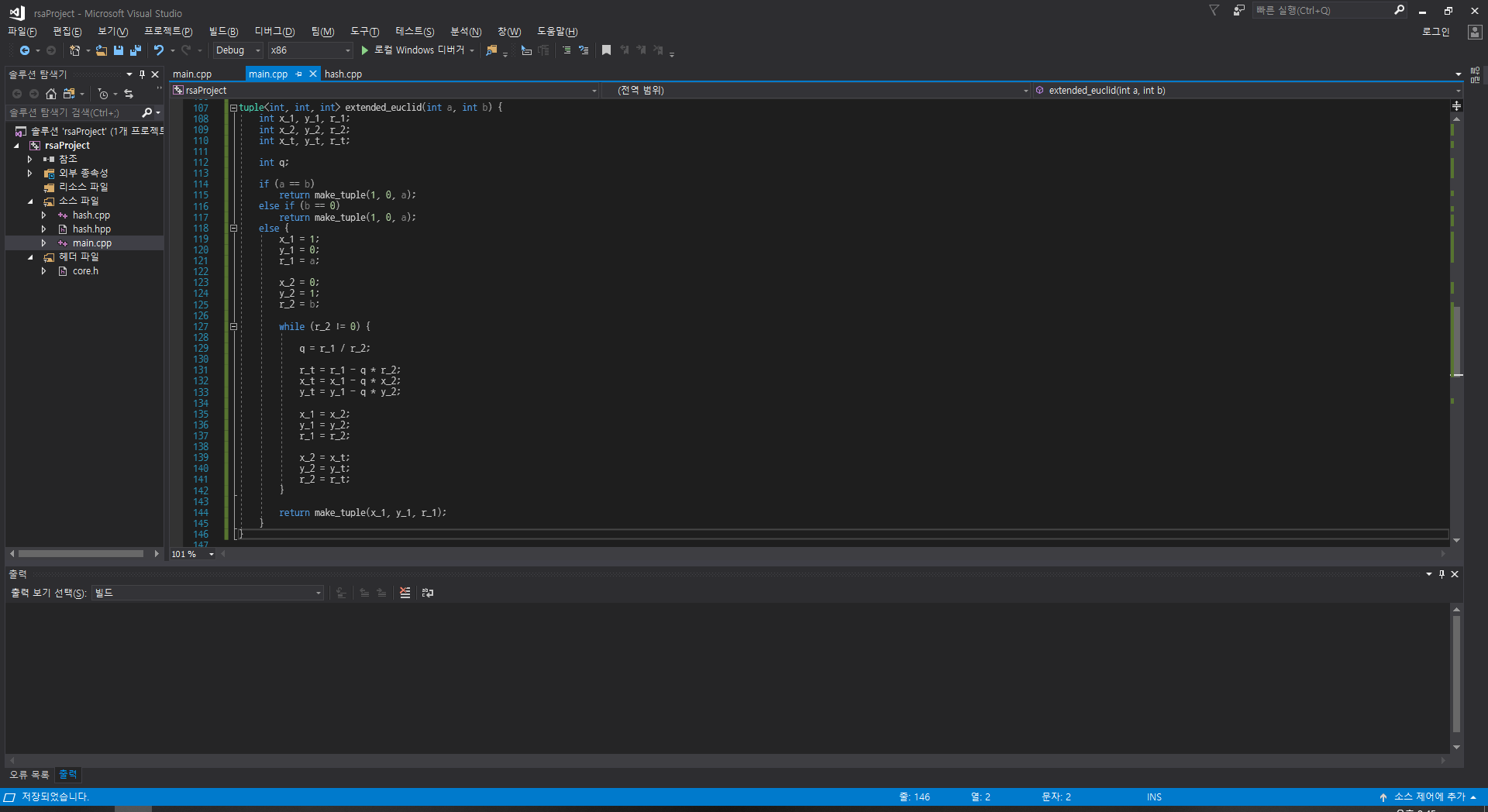
**2. 함수 설명**

1) gcd

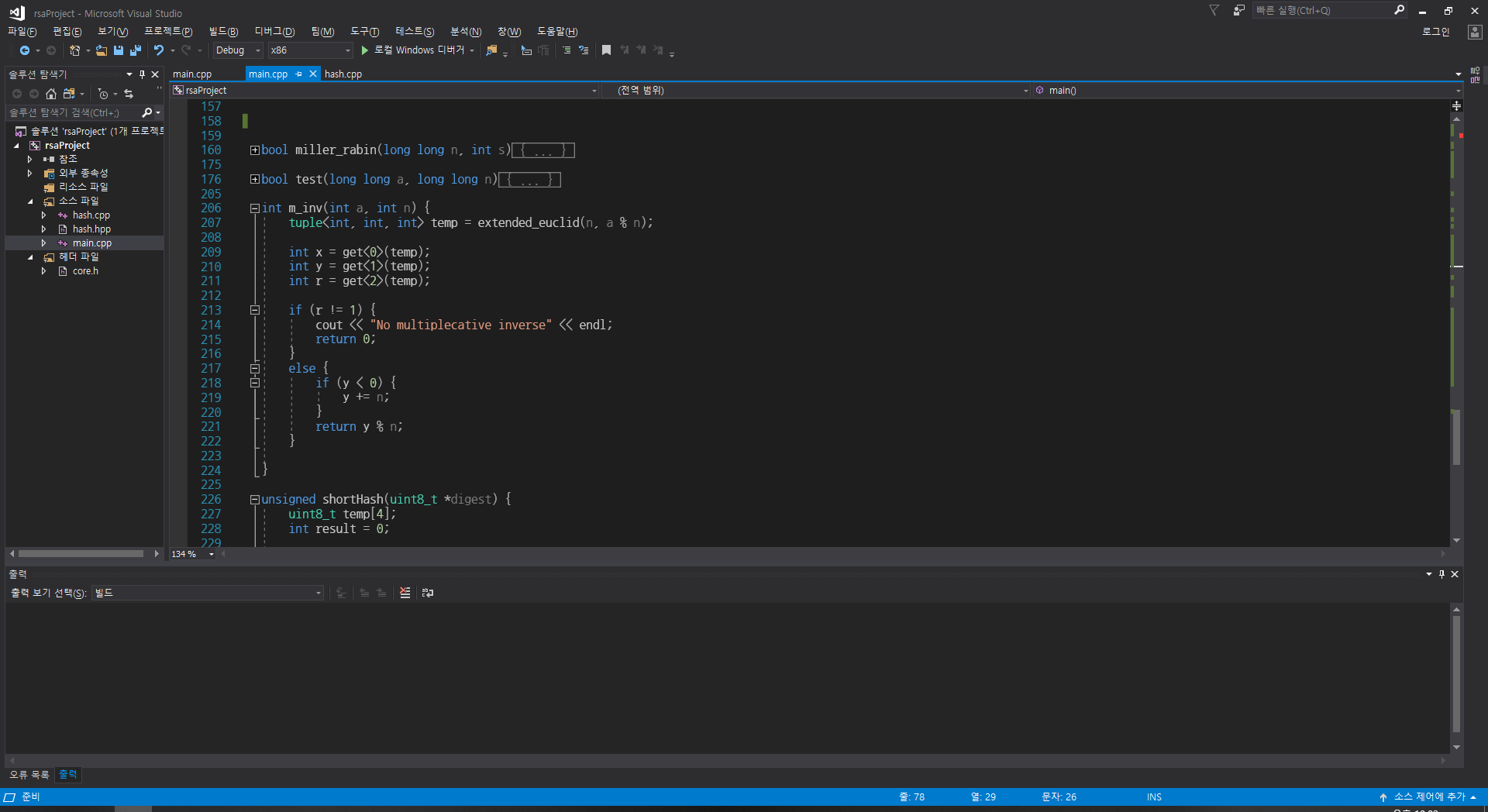


입력받은 a와 b값에 대해 최소공배수를 구합니다. a=b, b가 0이 될때까지 재귀적으로 b와, a를 b로 나눈 나머지로 함수를 호출합니다. phi\_n과 e의 gcd가 1이 아니면 곱의 역원이 존재하지 않으면 다음 step으로 넘어갈 수 없습니다.

2.extended\_euclid, m\_inv

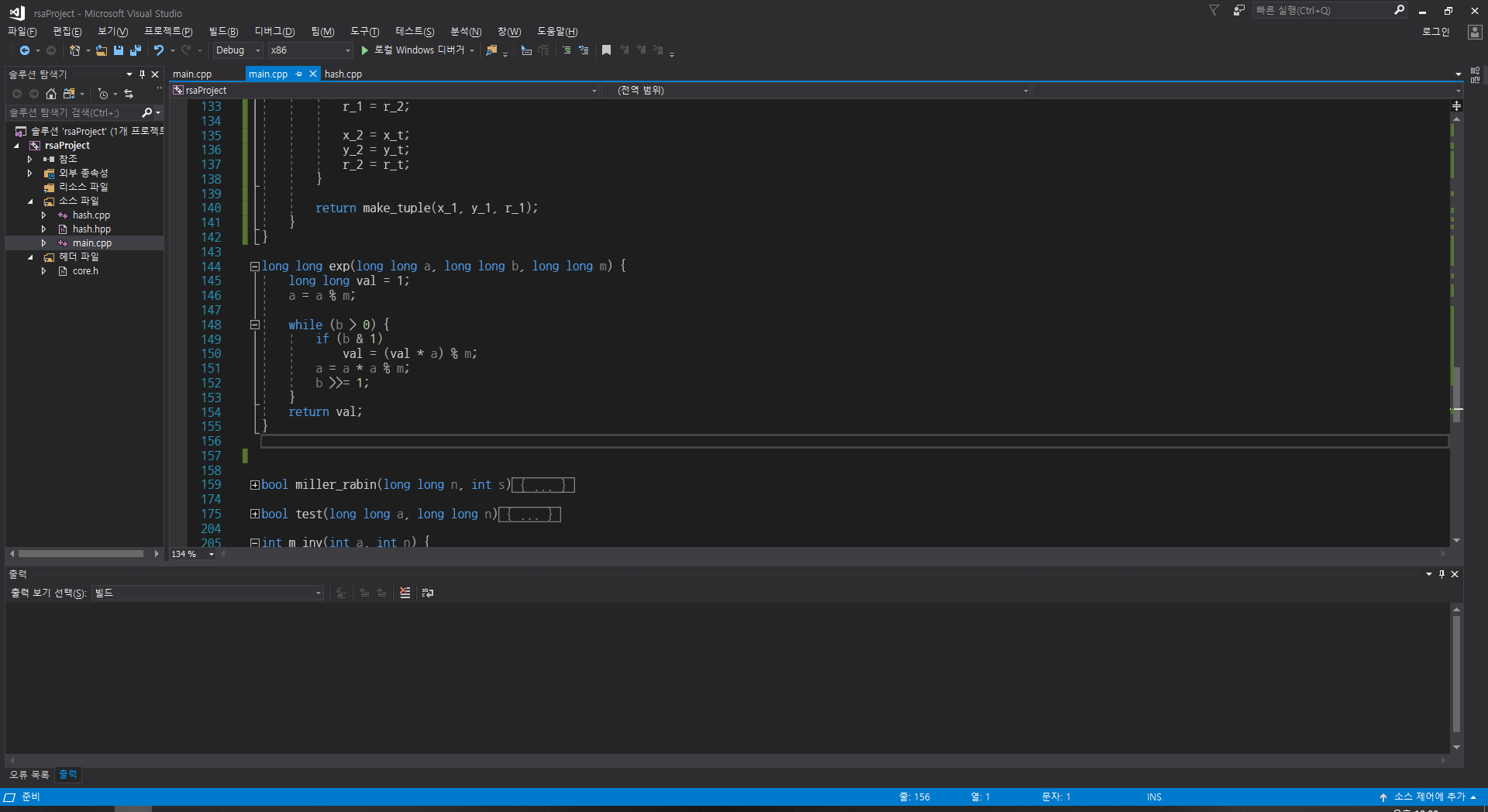


입력받은 a,b 에서 반복적으로 연산을 수행하며 a\*x + b\*y + r -> x, y, r을 리턴합니다.



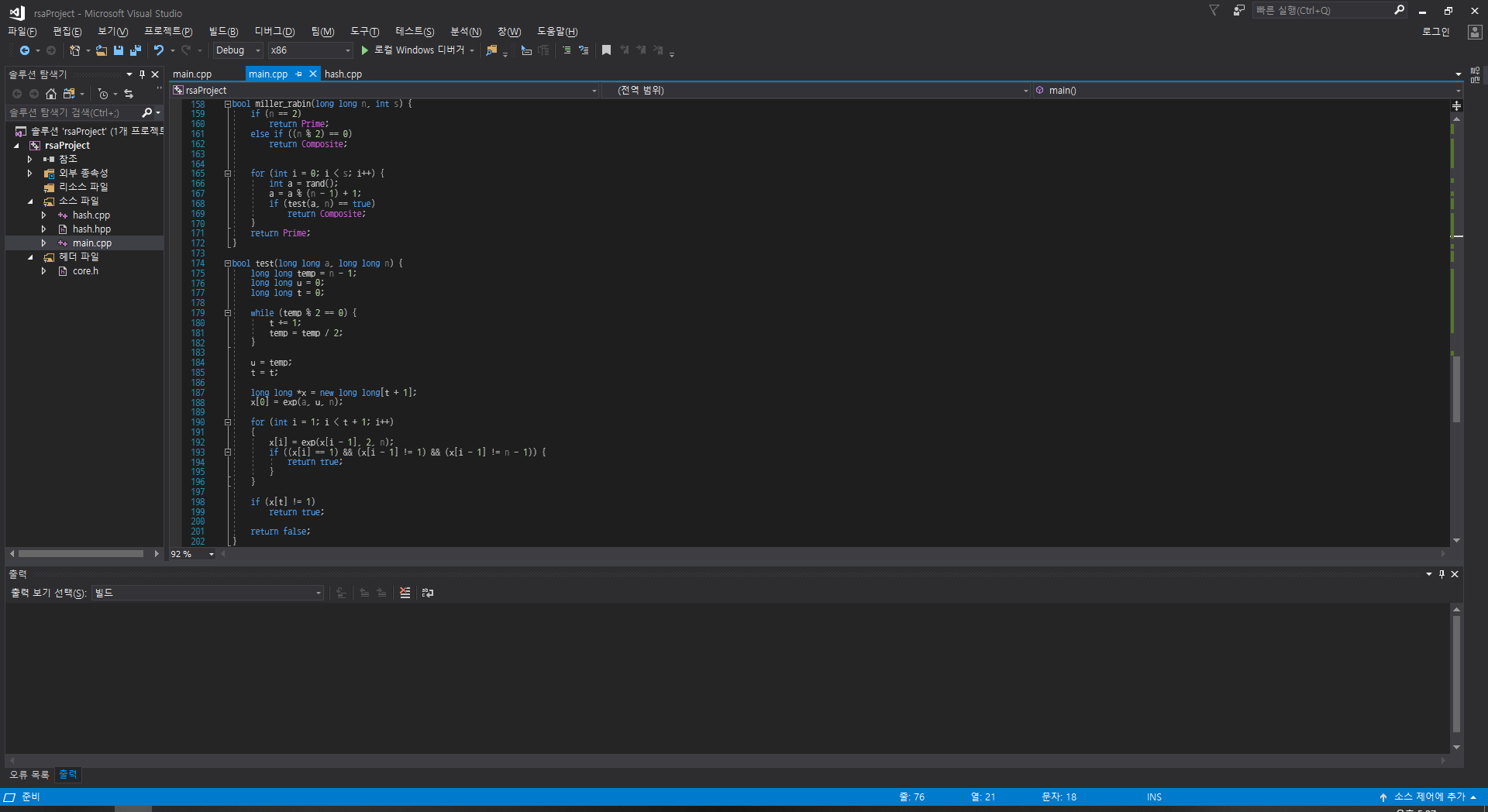
extended\_euclid함수로 받은 값을 받아 r이 1이 아니면 (gcd가 1이 아니면) 종료하고, 그렇지 않으면 (y % n) 곱의 역원을 리턴해주는데 만약 y가 음수이면 modular n을 한번 더해줘 양수를 리턴하게 해줍니다.

3. exp

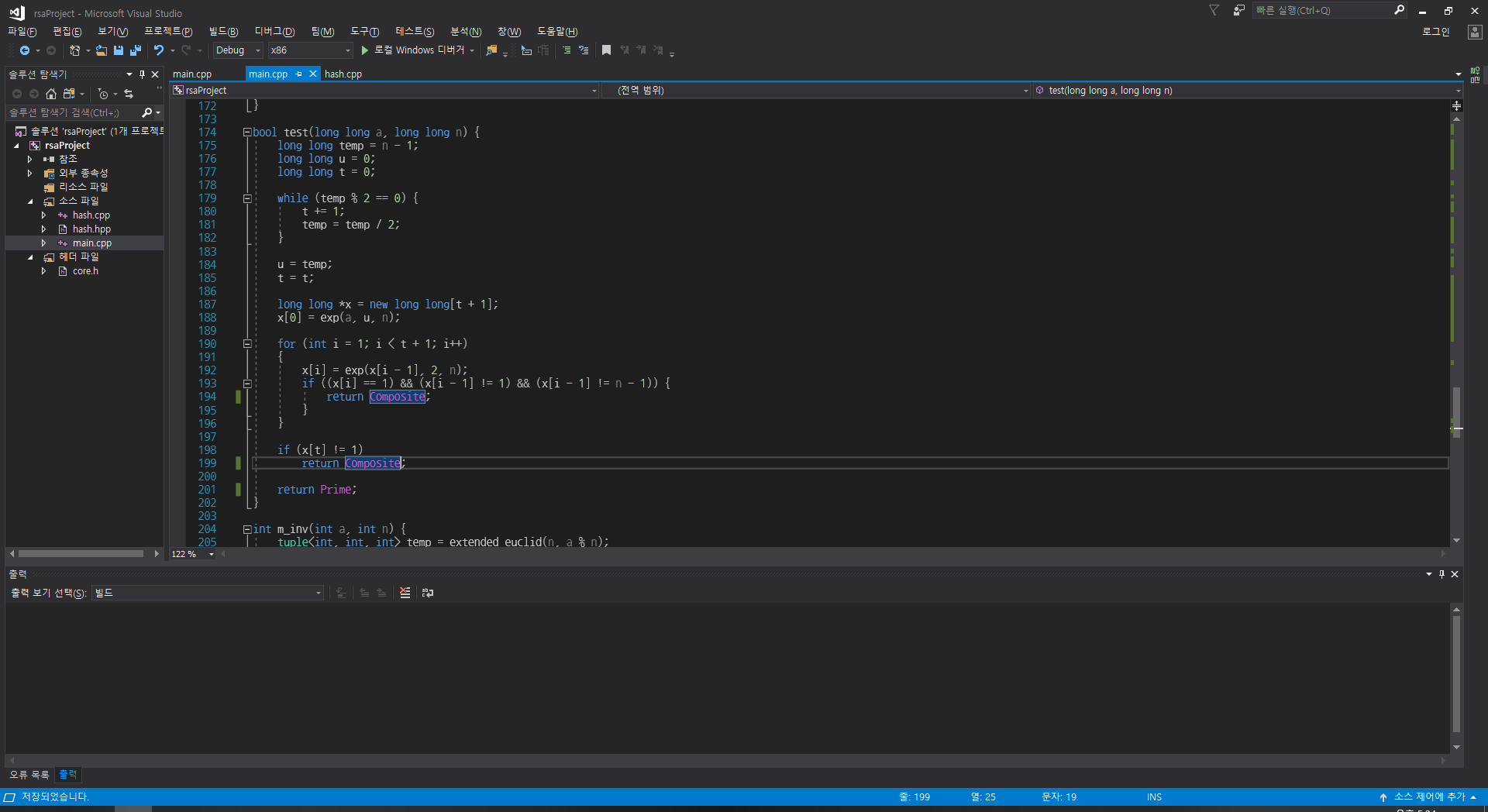


a^b mod m 을 위한 함수입니다. modular 성질-> a = a % m 후 expo 연산 수행

4.miller\_rabin, test



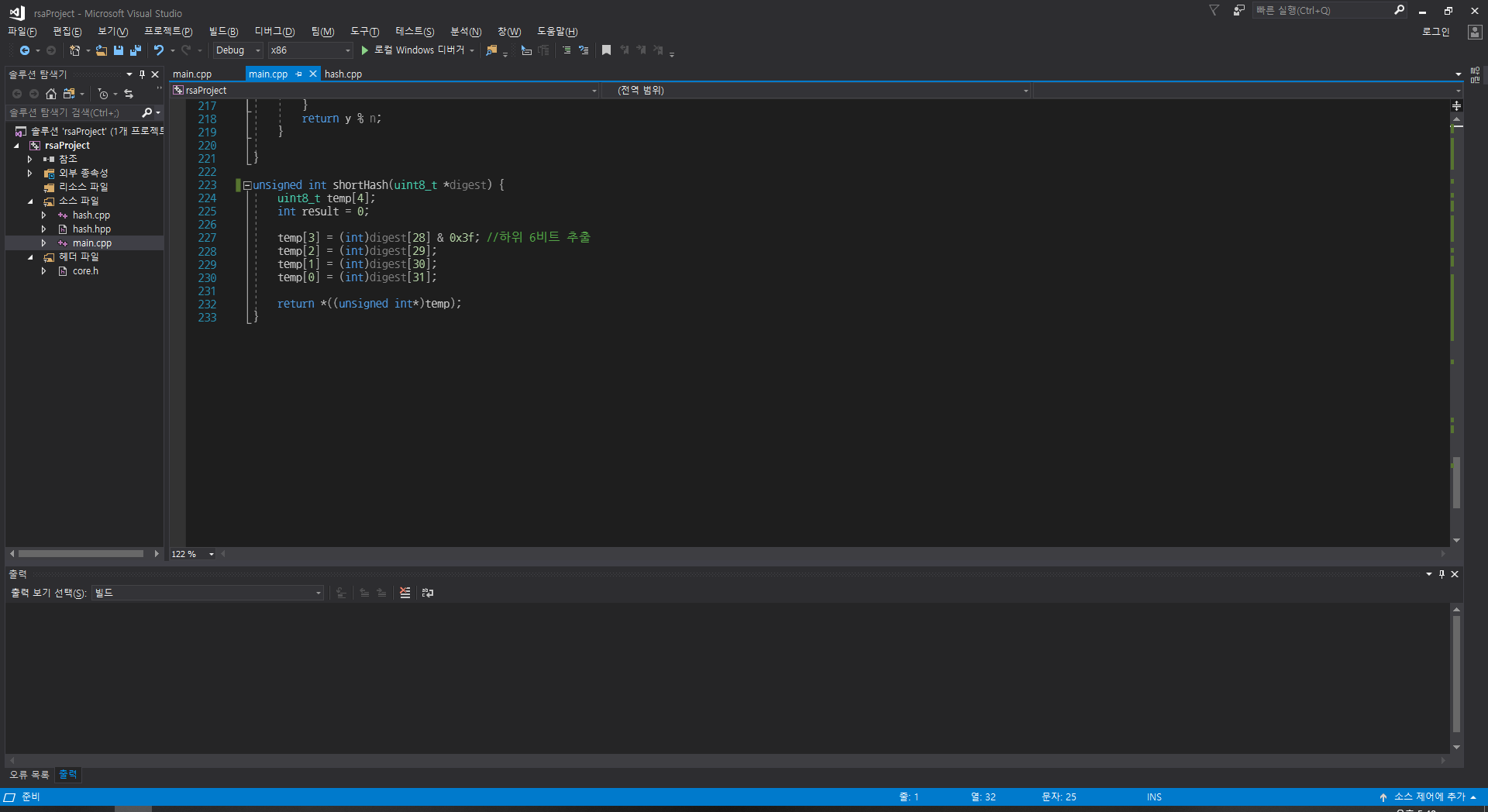
주어진 수 n에 대한 소수판별 함수로 랜덤한 수를 뽑아 NSR test, Fermat test를 s번 실시합니다.



NSR test – 랜덤한 값 a에 대해 a^mod n = 1 && a!=+1 or -1 인 a가 있는지 검사합니다.

Fermat test – a^(n-1) mod n != 1인지 검사합니다. 둘다 통과하게 되면 false(prime)을 리턴하게 됩니다.

5. shortHash

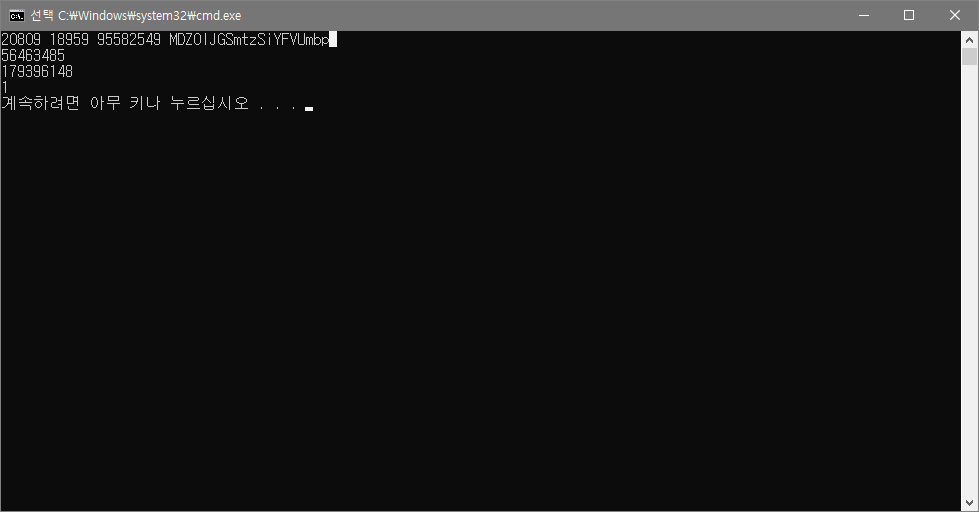


해쉬함수를 거친 digest 배열을 받아 digest[28]의 6비트, digest[29], digest[30], digest[31](6bit + 3byte = 30bit)를 추출해 임의의 temp 배열에 담아 정수형으로 반환합니다.

**3. 실행결과**

ex1) input: 20809 18959 95582549 MDZOlJGSmtzSiYFVUmbp

output:



ex2) input: 20809 18959 95582549 mDGeoryvOAIp

output:

