Stack	Project	Comments
<Cafe Signature> 15, 105	<Cafe Signature=??,???> <Cafe Public Key=??> DUP HASH160 <Cafe Public Key Hash=?> EQUALVERIFY CHECKSIG ^ | Value <sig> is pushed to the top of the stack --- Cafe Signature = Bob's signature ○ Step 1: Create a hash of the Alice's transaction (43) concatenated with Gopesh's public key. ○ Gopesh's public key = k*G = 7*5 = 35 ==> HASH = 4335 % 3 = 0 ○ ○ Step 2: Encrypt the hash with Bob's private key (3) ○ ○ ==> ○ ○ C1 = PR(Bob)*G =3*5=15 ○ C2 = Hash + PR(Bob) * PU(Gopesh) = 0+3*35=105	<sig> | <sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG ^ | Value <sig> is pushed to the top of the stack
<Cafe Public Key> 15 <Cafe Signature> 15, 105	<Cafe Signature=??,???> <Cafe Public Key=??> DUP HASH160 <Cafe Public Key Hash=?> EQUALVERIFY CHECKSIG ^ | Value <Cafe Public Key> is pushed to the stack, on top of <sig> --- Cafe Public Key = Bob's Public Key = 15	<PubK> <sig> | <sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG ^ | Value <PubK> is pushed to the stack, on top of <sig>

<Cafe Public Key> 15 <Cafe Public Key> 15 <Cafe Signature> 15, 105	<Cafe Signature=??,???> <Cafe Public Key=??> DUP HASH160 <Cafe Public Key Hash=?> EQUALVERIFY CHECKSIG ^ | DUP operator duplicates the top item in the stack, the resulting value is pushed to the top of the stack	<PubK> <PubK> <sig> | <sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG ^ | DUP operator duplicates the top item in the stack, the resulting value is pushed to the top of the stack
<Cafe Public Key Hash> 0 <Cafe Public Key> 15 <Cafe Signature>15, 105	<Cafe Signature=??,???> <Cafe Public Key=??> DUP HASH160 <Cafe Public Key Hash=?> EQUALVERIFY CHECKSIG ^ | HASH160 operator hashes the top item in the stack with RIPEMD160(SHA256(PubK)), the resulting value (Cafe Public Key Hash) is pushed to the top of the stack --- Cafe Public Key Hash = Bob Public Key Hash = 15 % 3 = 0	<PubKHash> <PubK> <sig> | <sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG ^ | HASH160 operator hashes the top item in the stack with RIPEMD160(SHA256(PubK)), the resulting value (PubKHash) is pushed to the top of the stack
<Cafe Public Key Hash> 0 <Cafe Public Key Hash> 0 <Cafe Public Key> 15 <Cafe Signature> 15, 105	<Cafe Signature=??,???> <Cafe Public Key=??> DUP HASH160 <Cafe Public Key Hash=?> EQUALVERIFY CHECKSIG ^ | The value <Cafe Public Key Hash> from the script is pushed on the top of the value <Cafe Public Key Hash> calculated previously from the HASH160 of the <Cafe Public Key>.	<PubKHash> <PubKHash> <PubK> <sig> | <sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG ^ | The value PubKHash from the script is pushed on the top of the value PubKHash calculated previously from the HASH160 of the PubK.

<Cafe Public Key> 15 <Cafe Signature> 15, 105	<Cafe Signature=??,???> <Cafe Public Key=??> DUP HASH160 <Cafe Public Key Hash=?> EQUALVERIFY CHECKSIG ^ | The EQAULVERIFY operator compares the PubKHash encumbering the transaction with the PubKHash calculated from the user's PubK. If they match, both are removed and execution contines	<PubK> <sig> | <sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG ^ | The EQAULVERIFY operator compares the PubKHash encumbering the transaction with the PubKHash calculated from the user's PubK. If they match, both are removed and execution contines
TRUE	<Cafe Signature=??,???> <Cafe Public Key=??> DUP HASH160 <Cafe Public Key Hash=?> EQUALVERIFY CHECKSIG ^ | The CHECKSIG operator checks that the signature <Cafe Signature> matches the public key &t;PubK> and pushes TRUE to the top of the stack if true. --- Refer Elliptic Curve Digital Signature Algorithm (ECDSA) --- HASH = C2 - d * C1 = C2 - PR_G * PU_B = 105 - 7 * 15 = 0	TRUE | <sig> <PubK> DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG ^ | The CHECKSIG operator checks that the signature <sig> matches the public key &t;PubK> and pushes TRUE to the top of the stack if true.