Module sui::bls12381

Group operations of BLS12-381.

@param signature: A 48-bytes signature that is a point on the G1 subgroup. @param public_key: A 96-bytes public key that is a point on the G2 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

@param signature: A 96-bytes signature that is a point on the G2 subgroup. @param public_key: A 48-bytes public key that is a point on the G1 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

Returns e2/e1, fails if a is zero.

Returns e2 / e1, fails if scalar is zero.

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

Convert an Element< G1 > to uncompressed form.

Returns e2 / e1, fails if scalar is zero.

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

Returns e2 / e1, fails if scalar is zero.

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

Struct

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

@param signature: A 48-bytes signature that is a point on the G1 subgroup. @param public_key: A 96-bytes public key that is a point on the G2 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

@param signature: A 96-bytes signature that is a point on the G2 subgroup. @param public_key: A 48-bytes public key that is a point on the G1 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Returns e2 / e1, fails if scalar is zero.

```bash

```

```bash

```

```bash

```

```bash

```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash

```

```bash

```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash

```

```bash

```

Convert an Element< G1 > to uncompressed form.

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash

```

```bash

```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash

```

```bash

```

Struct

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

@param signature: A 48-bytes signature that is a point on the G1 subgroup. @param public_key: A 96-bytes public key that is a point on the G2 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

@param signature: A 96-bytes signature that is a point on the G2 subgroup. @param public_key: A 48-bytes public key that is a point on the G1 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```
```

```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Struct

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

@param signature: A 48-bytes signature that is a point on the G1 subgroup. @param public_key: A 96-bytes public key that is a point on the G2 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

@param signature: A 96-bytes signature that is a point on the G2 subgroup. @param public_key: A 48-bytes public key that is a point on the G1 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Struct

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

@param signature: A 48-bytes signature that is a point on the G1 subgroup. @param public_key: A 96-bytes public key that is a point on the G2 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

@param signature: A 96-bytes signature that is a point on the G2 subgroup. @param public_key: A 48-bytes public key that is a point on the G1 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.
```

```bash
```

```bash
```

Struct

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

@param signature: A 48-bytes signature that is a point on the G1 subgroup. @param public_key: A 96-bytes public key that is a

point on the G2 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

@param signature: A 96-bytes signature that is a point on the G2 subgroup. @param public_key: A 48-bytes public key that is a point on the G1 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash

```

```bash

```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Returns e2 / e1, fails if scalar is zero.

```bash

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Constants

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

@param signature: A 48-bytes signature that is a point on the G1 subgroup. @param public_key: A 96-bytes public key that is a point on the G2 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

@param signature: A 96-bytes signature that is a point on the G2 subgroup. @param public_key: A 48-bytes public key that is a point on the G1 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

```

## Function

@param signature: A 48-bytes signature that is a point on the G1 subgroup. @param public_key: A 96-bytes public key that is a point on the G2 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash

```

```bash

```

@param signature: A 96-bytes signature that is a point on the G2 subgroup. @param public_key: A 48-bytes public key that is a point on the G1 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

Returns e2 / e1, fails if scalar is zero.

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash

```

```bash

```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash

```

```bash

```

Function

@param signature: A 96-bytes signature that is a point on the G2 subgroup. @param public_key: A 48-bytes public key that is a point on the G1 subgroup. @param msg: The message that we test the signature against.

If the signature is a valid signature of the message and public key according to BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_, return true. Otherwise, return false.

```bash

```

```bash

```

```bash

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```
```

```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash

```

```bash

```

Function

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts
```

with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```
```

```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```
```

```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts
```

with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```
```

```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash

```

```bash

```

Function

```bash

```

```bash

```

```bash

```

```bash

```

Returns e2/e1, fails if a is zero.

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

Returns e2/e1, fails if a is zero.
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

Returns e2/e1, fails if a is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

```

## Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```
```

```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```
```

```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash

```

```bash

```

Function

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

Returns e2 / e1, fails if scalar is zero.

```bash

```
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.
```

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash

```

```bash

```

Convert an Element< G1 > to uncompressed form.

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash

```

```bash

```

Convert an Element< G1 > to uncompressed form.

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```

```bash
```

Function

Hash using DST = BLS_SIG_BLS12381G1_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```
```

```bash
```

Function

Convert an Element< G1 > to uncompressed form.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_
```

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_
```

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash
```

```bash
```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```
```

```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```
```

```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash

```

```bash

```

Function

Hash using DST = BLS_SIG_BLS12381G2_XMD:SHA-256_SSWU_RO_NUL_

```bash

```

```bash

```

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```

```bash

```
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

Let 'scalars' be the vector [s1, s2, ..., sn] and 'elements' be the vector [e1, e2, ..., en]. Returns s1 e1 + s2 e2 + ... + sn*en. Aborts with EInputTooLong if the vectors are larger than 32 (may increase in the future).

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

Returns e2 / e1, fails if scalar is zero.

```bash
```

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.
```

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

```bash
```

```bash
```

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

UncompressedG1 group operations /// Create a Element< G1 > from its uncompressed form.

```bash
```

```bash
```

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```

Function

Compute the sum of a list of uncompressed elements. This is significantly faster and cheaper than summing the elements.

```bash
```

```bash
```