

```
/*
 * Moebius_transformations.c
 */

#include "kernel.h"

CONST MoebiusTransformation Moebius_identity =
{
    {
        {{1.0, 0.0}, {0.0, 0.0}},
        {{0.0, 0.0}, {1.0, 0.0}}
    },
    orientation_preserving
};

void Moebius_copy(
    MoebiusTransformation *dest,
    MoebiusTransformation *source)
{
    sl2c_copy(dest->matrix, source->matrix);
    dest->parity = source->parity;
}

void Moebius_invert(
    MoebiusTransformation *mt,
    MoebiusTransformation *mt_inverse)
{
    sl2c_invert(mt->matrix, mt_inverse->matrix);
    if (mt->parity == orientation_reversing)
        sl2c_complex_conjugate(mt_inverse->matrix, mt_inverse->matrix);

    mt_inverse->parity = mt->parity;
}

void Moebius_product(
    MoebiusTransformation *a,
    MoebiusTransformation *b,
    MoebiusTransformation *product)
{
    SL2CMatrix factor1,
                factor2;

    sl2c_copy(factor1, a->matrix);
    sl2c_copy(factor2, b->matrix);

    if (a->parity == orientation_reversing)
        sl2c_complex_conjugate(factor2, factor2);

    sl2c_product(factor1, factor2, product->matrix);

    product->parity = (a->parity == b->parity) ?
        orientation_preserving:
        orientation_reversing;
}
```