

```

/*
 * This file contains the functions
 *
 * Boolean all_Dehn_coefficients_are_integers(Triangulation *manifold);
 * Boolean Dehn_coefficients_are_integers(Cusp *cusp);
 *
 * Boolean all_Dehn_coefficients_are_relatively_prime_integers(Triangulation *
manifold);
 * Boolean Dehn_coefficients_are_relatively_prime_integers(Cusp *cusp);
 *
 * Boolean all_cusps_are_complete(Triangulation *manifold);
 * Boolean all_cusps_are_filled(Triangulation *manifold);
 *
 * which are used within the kernel to test whether Dehn filling coefficients
 * are (relatively prime) integers.
 */

```

```

#include "kernel.h"

```

```

Boolean all_Dehn_coefficients_are_integers(
    Triangulation *manifold)
{
    Cusp *cusp;

    for (cusp = manifold->cusp_list_begin.next;
         cusp != &manifold->cusp_list_end;
         cusp = cusp->next)

        if (Dehn_coefficients_are_integers(cusp) == FALSE)

            return FALSE;

    return TRUE;
}

```

```

Boolean all_Dehn_coefficients_are_relatively_prime_integers(
    Triangulation *manifold)
{
    Cusp *cusp;

    for (cusp = manifold->cusp_list_begin.next;
         cusp != &manifold->cusp_list_end;
         cusp = cusp->next)

        if (Dehn_coefficients_are_relatively_prime_integers(cusp) == FALSE)

            return FALSE;

    return TRUE;
}

```

```

Boolean Dehn_coefficients_are_integers(
    Cusp *cusp)
{
    return
    (
        cusp->is_complete == TRUE
        ||
        (
            cusp->m == (double)(int)cusp->m
            && cusp->l == (double)(int)cusp->l
        )
    );
}

```

```

Boolean Dehn_coefficients_are_relatively_prime_integers(
    Cusp *cusp)
{
    return
    (

```

```
    cusp->is_complete == TRUE
||
    (
        cusp->m == (double)(int)cusp->m
        && cusp->l == (double)(int)cusp->l
        && gcd((long int)cusp->m, (long int)cusp->l) == 1
    )
);
}
```

```
Boolean all_cusps_are_complete(
    Triangulation *manifold)
{
    Cusp *cusp;

    for (cusp = manifold->cusp_list_begin.next;
         cusp != &manifold->cusp_list_end;
         cusp = cusp->next)

        if (cusp->is_complete == FALSE)

            return FALSE;

    return TRUE;
}
```

```
Boolean all_cusps_are_filled(
    Triangulation *manifold)
{
    Cusp *cusp;

    for (cusp = manifold->cusp_list_begin.next;
         cusp != &manifold->cusp_list_end;
         cusp = cusp->next)

        if (cusp->is_complete == TRUE)

            return FALSE;

    return TRUE;
}
```