```
/*---------------------------------------------------------------------=*\
  Use :
\*---------------------------------------------------------------------=*/
void FCM_Main()
{

    // Name: Loop, Type: Loop: While 1
    while (1)
    {

        // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(3, unita, 0)
        FCD_0fca1_led_7seg_quad1__ShowDigit(3, FCV_UNITA, 0);

        // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(2, decine, 0)
        FCD_0fca1_led_7seg_quad1__ShowDigit(2, FCV_DECINE, 0);

        // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(1, centinaia, 0)
        FCD_0fca1_led_7seg_quad1__ShowDigit(1, FCV_CENTINAIA, 0);

        // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad1::ShowDigit(0, migliaia, 1)
        FCD_0fca1_led_7seg_quad1__ShowDigit(0, FCV_MIGLIAIA, 1);

        // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad2::ShowDigit(3, diecimila, 0)
        FCD_0fca2_led_7seg_quad2__ShowDigit(3, FCV_DIECIMILA, 0);

        // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad2::ShowDigit(2, centomila, 0)
        FCD_0fca2_led_7seg_quad2__ShowDigit(2, FCV_CENTOMILA, 0);

        // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad2::ShowDigit(1, mega, 1)
        FCD_0fca2_led_7seg_quad2__ShowDigit(1, FCV_MEGA, 1);

        // Name: Call Component Macro, Type: Call Component Macro: led_7seg_quad2::ShowDigit(0, diecimega, 0)
        FCD_0fca2_led_7seg_quad2__ShowDigit(0, FCV_DIECIMEGA, 0);

        // Name: Calculation, Type: Calculation:
        // unita = unita + 1
        // unita = unita % 10
        FCV_UNITA = FCV_UNITA + 1;
        FCV_UNITA = FCV_UNITA % 10;

        // Name: Decision, Type: Decision: unita = 0?
        if (FCV_UNITA == 0)
        {

            // Name: Calculation, Type: Calculation:
            // decine = decine + 1
            // decine = decine % 10
            FCV_DECINE = FCV_DECINE + 1;
            FCV_DECINE = FCV_DECINE % 10;

            // Name: Decision, Type: Decision: decine = 0?
            if (FCV_DECINE == 0)
            {

                // Name: Calculation, Type: Calculation:
                // centinaia = centinaia + 1
                // centinaia = centinaia % 10
                FCV_CENTINAIA = FCV_CENTINAIA + 1;
                FCV_CENTINAIA = FCV_CENTINAIA % 10;

                // Name: Decision, Type: Decision: centinaia = 0?
                if (FCV_CENTINAIA == 0)
                {

                    // Name: Calculation, Type: Calculation:
                    // migliaia = migliaia + 1
                    // migliaia = migliaia % 10
                    FCV_MIGLIAIA = FCV_MIGLIAIA + 1;
                    FCV_MIGLIAIA = FCV_MIGLIAIA % 10;

                    // Name: Decision, Type: Decision: migliaia = 0?
                    if (FCV_MIGLIAIA == 0)
                    {

                        // Name: Calculation, Type: Calculation:
                        // diecimila = diecimila + 1
                        // diecimila = diecimila % 10
                        FCV_DIECIMILA = FCV_DIECIMILA + 1;
                        FCV_DIECIMILA = FCV_DIECIMILA % 10;

                        // Name: Decision, Type: Decision: diecimila = 0?
                        if (FCV_DIECIMILA == 0)
                        {

                            // Name: Calculation, Type: Calculation:
                            // centomila = centomila + 1
                            // centomila = centomila % 10
                            FCV_CENTOMILA = FCV_CENTOMILA + 1;
                            FCV_CENTOMILA = FCV_CENTOMILA % 10;

                            // Name: Decision, Type: Decision: centomila = 0?
                            if (FCV_CENTOMILA == 0)
                            {

                                // Name: Calculation, Type: Calculation:
                                // mega = mega + 1
                                // mega = mega % 10
                                FCV_MEGA = FCV_MEGA + 1;
                                FCV_MEGA = FCV_MEGA % 10;

                                // Name: Decision, Type: Decision: mega = 0?
                                if (FCV_MEGA == 0)
                                {

                                    // Name: Calculation, Type: Calculation:
                                    // diecimega = diecimega + 1
                                    // diecimega = diecimega % 10
                                    FCV_DIECIMEGA = FCV_DIECIMEGA + 1;
                                    FCV_DIECIMEGA = FCV_DIECIMEGA % 10;

                                // } else {

                                }

                            // } else {

                            }

                        // } else {

                        }

                    // } else {

                    }

                // } else {

                }

            // } else {

            }

        // } else {

        }

    }

}
```