

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

```
    // Name: Call Component Macro, Type: Call Component Macro: LCDI2C1::Start()
    FCD_0be11_LCDI2C1__Start();
```

```
    // Name: Set secret code, Type: Calculation:
    // codice[0] = 5
    // codice[1] = 5
    // codice[2] = 7
    // codice[3] = 8
    FCV_CODICE[0] = 5;
    FCV_CODICE[1] = 5;
    FCV_CODICE[2] = 7;
    FCV_CODICE[3] = 8;
```

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

```
        // Name: See which key has been pressed, Type: Call Component Macro: KeyPremuto=v5__keypad0::GetAscii()
        FCV_KEYPREMUTO = FCD_0c871_v5__keypad0__GetAscii();
```

```
        // Name: Any key at all, Type: Decision: KeyPremuto <> 255?
        if (FCV_KEYPREMUTO != 255)
        {
```

```
            // Name: Key #, Type: Decision: KeyPremuto = 35?
            if (FCV_KEYPREMUTO == 35)
            {
```

```
                // Name: Call Component Macro, Type: Call Component Macro: LCDI2C1::Clear()
                FCD_0be11_LCDI2C1__Clear();
```

```
            } else {
```

```
                // Name: Call Component Macro, Type: Call Component Macro: LCDI2C1::PrintAscii(KeyPremuto)
                FCD_0be11_LCDI2C1__PrintAscii(FCV_KEYPREMUTO);
```

```
                // Name: Shift doorbel, Type: Calculation:
                // KeyPremuto = KeyPremuto - 48
                // corretto = 0
                FCV_KEYPREMUTO = FCV_KEYPREMUTO - 48;
                FCV_CORRETTO = 0;
```

```
                // Name: Calculation, Type: Calculation:
                // combinazione[0] = combinazione[1]
                FCV_COMBINAZIONE[0] = FCV_COMBINAZIONE[1];
```

```
                // Name: Correct number, Type: Decision: combinazione[0] = codice[0]?
                if (FCV_COMBINAZIONE[0] == FCV_CODICE[0])
                {
```

```
                    // Name: Keep score, Type: Calculation:
                    // corretto = corretto + 1
                    FCV_CORRETTO = FCV_CORRETTO + 1;
```

```
                // } else {
                }
            }
```

```
                // Name: passa al codice successivo, Type: Calculation:
                // combinazione[1] = combinazione[2]
                FCV_COMBINAZIONE[1] = FCV_COMBINAZIONE[2];
```

```
                // Name: Correct number, Type: Decision: combinazione[1] = codice[1]?
                if (FCV_COMBINAZIONE[1] == FCV_CODICE[1])
                {
```

```

// Name: Keep score, Type: Calculation:
// corretto = corretto + 1
FCV_CORRETTO = FCV_CORRETTO + 1;

// } else {
}

// Name: passa al codice successivo, Type: Calculation:
// combinazione[2] = combinazione[3]
FCV_COMBINAZIONE[2] = FCV_COMBINAZIONE[3];

// Name: Correct number, Type: Decision: combinazione[2] = codice[2]?
if (FCV_COMBINAZIONE[2] == FCV_CODICE[2])
{

// Name: Keep score, Type: Calculation:
// corretto = corretto + 1
FCV_CORRETTO = FCV_CORRETTO + 1;

// } else {
}

// Name: passa al codice successivo, Type: Calculation:
// combinazione[3] = KeyPremuto
FCV_COMBINAZIONE[3] = FCV_KEYPREMUTO;

// Name: Correct number, Type: Decision: combinazione[3] = codice[3]?
if (FCV_COMBINAZIONE[3] == FCV_CODICE[3])
{

// Name: Keep score, Type: Calculation:
// corretto = corretto + 1
FCV_CORRETTO = FCV_CORRETTO + 1;

// } else {
}
}

// Name: aspetta che il tasto sia rilasciato, Type: Loop: While KeyPremuto <> 255
while (FCV_KEYPREMUTO != 255)
{

// Name: , Type: Call Component Macro: KeyPremuto=v5_keypad0::GetAscii()
FCV_KEYPREMUTO = FCD_0c871_v5_keypad0_GetAscii();

}

// Name: tutti i codici corrispondono, Type: Decision: corretto = 4?
if (FCV_CORRETTO == 4)
{

// Name: Call Component Macro, Type: Call Component Macro: LCDI2C1::Cursor(0, 1)
FCD_0be11_LCDI2C1_Cursor(0, 1);

// Name: Call Component Macro, Type: Call Component Macro: LCDI2C1::PrintString("codice esatto")
FCD_0be11_LCDI2C1_PrintString("codice esatto", 14);

// Name: Output, Type: Output: 1 -> D0
SET_PORT_PIN(D,0,(1));

// Name: Delay, Type: Delay: 3 s
FCI_DELAYBYTE_S(3);

// Name: Output, Type: Output: 0 -> D0
SET_PORT_PIN(D,0,(0));

// Name: Call Component Macro, Type: Call Component Macro: LCDI2C1::Clear()
FCD_0be11_LCDI2C1_Clear();

// } else {
}

// } else {
}

}
}

```