```c
#include <stdio.h>

#define N       20
#define ZERO    0
#define ONE     1
#define ABS(x)  (x) * (((2 * (x)) + 1) % 2)
#define BOOLEAN unsigned short

//---------------------------------------------------------------------------------------
//                                              Equality
//                                              --------
//
// General      : Checks whether two numbers are equal in value.
//
// Parameters   :
//          number1 -  first number (int)
//          number2 -  second number (int)
//
// Return value : If there is an equality between the two numbers (BOOLEAN).
//
//---------------------------------------------------------------------------------------
// Programmer : Cohen Idan
// Student No : None
// Date       : 21.10.2019
//---------------------------------------------------------------------------------------
BOOLEAN Equality(int number1, int number2)
{
    int sub = number1 - number2;
    BOOLEAN answer = (ONE / (ABS(sub) + ONE));
           answer;
}

//---------------------------------------------------------------------------------------
//                                              Exercise 3
//                                              ----------
//
// General : The program checks the number of neighboring organs whose values are equal.
//
// Input   : None.
//
// Process : The program checks the number of neighboring organs whose values are equal.
//
// Output  : The number of neighboring organ pairs whose values
//           are equal (unsigned short).
//
//---------------------------------------------------------------------------------------
// Programmer : Cohen Idan
// Student No : 211675038
// Date       : 23.10.2019
//---------------------------------------------------------------------------------------
void main(void)
{
    unsigned short board[N] =
    {1,1,3,3,4,5,5,6,7,7,4,3,2,3,1,1,0,1,0,1};

    unsigned short sum = ZERO;
    unsigned short counter;

        (counter = ONE; counter < N; counter++)
    {
        sum += Equality(board[counter - ONE], board[counter]);
    }

    printf("Sum: %hu\n", sum);
}
```