```c
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <string.h>
#include <process.h>
#include "lctypdef.h"
#include "fixfunc.h"
#include "crtdbg.h"              /* debug tools                             */

#define RING_SIZE 1251 /* length of the delay line in DetectFixation() -- */ /* should be greater than */ iMinimumFixSamples */
#define NEW_FIX 0
#define PRES_FIX 1
#define PREV_FIX 2

/* STRUCTURES DEFINITIONS: */

struct _stFix /* FIXATION DATA */
{
    int iStartCount; /* call count that the fixation starts */
    int iEndCount; /* call count that the fixation ends */
    int iNSamples; /* number of samples in the present fix */
    float fXSum; /* summations for calculation of average */
    float fYSum; /* fixation position */
    float fx; /* average coordinate of the eye fixation */
    float fy; /* point (user selected units) */
};

struct _stRingBuf /* RING BUFFER STORING PAST VALUES: */
{
    float fXGaze; /* gazepoint coordinate */
    float fYGaze; /* */
    BOOL bGazeFound; /* gazepoint found flag */
    int iEyeMotionState; /* state of the eye motion: */
        /* MOVING */
        /* FIXATING */
        /* FIXATION_COMPLETED */
    float fXFix; /* current fixation coordinate */
    float fYFix; /* */
    float fGazeDeviation; /* gazepoint deviation from the fixation */
};
```
int iSacDuration; /* saccade duration */
int iFixDuration; /* fixation duration */

/* GLOBAL FIXFUNC VARIABLES: */
int iCallCount; /* number of times this function has been called since it was initialized */
/* (30ths or 60ths of a second, depending on eyetracking sample rate) */
int iNPresOut; /* number of successive gazepoint samples outside the present fixation circle */

int iNSamplesSinceLastGoodFixPoint; /* number of samples since the last known good gazepoint in the present fixation */
/* Note: During continuous tracking of a normal fixation, this value is 1. */
float fPresDr; /* difference between gazepoint and fixation (x, y, and radius) */
float fNewDr; /* difference between gazepoint and fixation (x, y, and radius) */
int iMaxMissedSamples; /* maximum number of successive gazepoint samples that may go untracked before a fixation is allowed to continue */
int iMaxOutSamples; /* maximum number of successive gazepoint samples that may go outside the fixation circle */

struct _stFix stFix[3]; /* prior, present and new fixations */
/* indexed by: NEW_FIX 0 */
/* PRES_FIX 1 */
/* PREV_FIX 2 */

struct _stRingBuf stRingBuf[RING_SIZE];
int iRingIndex; /* ring index of the present gaze sample */
int iRingIndexDelay; /* ring index of the gaze sample taken iMinimumFixSamples ago */

#endif /* DIAG_FIXFUNC */

void InitFixation(int iMinimumFixSamples) {
/* minimum number of gaze samples that can be considered a fixation */
/* Note: if the input value is less than 3, the function sets it to 3 */
/* This function clears any previous, present and new fixations, and it initializes DetectFixation()'s internal ring buffers of prior gazepoint data. InitFixation() should be called prior to a sequence of calls to DetectFixation(). */
{
/* Set the maximum allowable number of consecutive samples that may go untracked within a fixation. */
iMaxMissedSamples = 3;
/* Set the maximum allowable number of consecutive samples that may go outside the fixation circle. */
iMaxOutSamples = 1;
/* Initialize the internal ring buffer. */
for (iRingIndex = 0; iRingIndex < RING_SIZE; iRingIndex++)
{
    stRingBuf[iRingIndex].fXGaze = 0.0F;
    stRingBuf[iRingIndex].fYGaze = 0.0F;
}
}