```c
/* Lab EXAMPLE - Name:  I. M. Student */

/* headers */
#include "stdio.h"
#include "T1.h"
#include <string.h>
#include <stdarg.h>

/* prototypes */
double  omegad (double m, double b, double k);  /* d. nat. frequency */
double  double_in( char *prompt );      /* user integer input */

/* definitions */
#define     M   1.0
#define     B   2.0

/* directives */

/* globals */

/*--------------------------------------------------------------------
    Function main
    Purpose: Calculate four values of the damped natural Frequency.
--------------------------------------------------------------------*/
int main(int argc, char **argv) {
    NiFpga_Status status;
    int     i;  /* index */
    double  fd; /* damped natural frquency */
    double  ks; /* spring constant */

    status = MyRio_Open();          /*Open the myRIO NiFpga Session.*/
    if (MyRio_IsNotSuccess(status)) return status;

    for(i=1; i<5; i++) {
        ks=double_in("Enter Spring Constant");
        fd=omegad(M, B, ks)/2/3.14159;
        printf("\fThis is the damped natural frequency:\n");
        printf("%g Hz",fd);
    }

    status = MyRio_Close();  /*Close the myRIO NiFpga Session. */
    return status;
}
/*--------------------------------------------------------------
Function omegad
    Purpose:    adds a square to the argument.
    Parameters:  (in) - m, mass
                 (in) - b, damping
                 (in) - k, spring constant
    Returns:     wd - damped natural frequncy (r/s)
*--------------------------------------------------------------*/
double  omegad(double m, double b, double k)  {
    double  zeta;   /* damping ratio */
    double  wn;     /* natural frequency */
    double  wd;     /* damped natural frequency */

/* Compute the damping ratio & natural frequency */
    zeta=b/2/sqrt(m*k);
    wn=sqrt(k/m);
    wd=wn*sqrt(1-zeta*zeta);
    return wd;
}
```