```
    i=(int)ibcnt;                          /* save total number of bytes read */

    txLia("PAUS");                /* pause the data storage so no new points are taken */
    printOutBinaryResults();      /* format and print the results */

    printf("\n%d bytes received.\nPress <Enter> to continue.",i);
    getch(); printf("\n");

    printf("Reading Results in IEEE Binary Format\n");
    txLia("SPTS?");               /* how many points in CH1 (R) buffer? */
    ibrd(lia,tstr,20L);           /* get the answer */
    sscanf(tstr,"%d",&nPts);      /* convert from a string to an int */
    printf ("SPTS?=%d\n",nPts);

    sprintf(tstr,"TRCB?1,0,%d",nPts);      /* use TRCB? to read the points in IEEE floating point format */
    ibwrt(lia,tstr,strlen(tstr));          /* note that we cannot use txLia here because the IFC RDY bit will
                                           not be set until the transfer is complete! */
    ibrd(lia,(char *)rfBuf,(long)nPts*4L); /* read directly into a FLOAT array, 4 bytes per point */

    printf ("\nReceived %d bytes in IEEE binary format\n",ibcnt);
    printOutIEEEResults();        /* format and print results */
    printf ("Press <Enter> to continue");
    getch(); printf("\n");

    printf("Reading Results in LIA Binary Format\n");
    sprintf(tstr,"TRCL?1,0,%d",nPts);      /* use TRCL? to read the points in LIA floating point format */
    ibwrt(lia,tstr,strlen(tstr));          /* note that we cannot use txLia here because the IFC RDY bit will
                                           not be set until the transfer is complete! */
    ibrd(lia,(char *)rfBuf,(long)nPts*4L); /* read into FLOAT array but the values are NOT floats! */

    printf ("\nReceived %d bytes in LIA binary format\n",ibcnt);
    printOutLIAResults();         /* format and print results */
    printf ("End of Program");
}

void printOutBinaryResults(void)
{
    /* calculates the first 10 values of R based on the X and Y values taken in FAST mode by the SR830 */

    int i;
    float x,y,r;
    int *ptr;

    printf("\n\n");
    ptr = rxBuf;  /* ptr points to the first X,Y pair of values. X and Y are each integers. */
    for (i=0;i<10;i++)      {
    x = (float) (*ptr++) /(float) 30000.0;    /* 30000 is full scale which is 1 V in this case */
    y = (float) (*ptr++) /(float) 30000.0;    /* for other scales, multiply by the full scale voltage */
    r = (float) sqrt(x*x + y*y);      /* compute R from X and Y */
    printf("%d    %e\n",i,r);
    }
}
```