

```

#include<stdio.h>
#include<stdlib.h>

int fnRightShift(unsigned int , unsigned int);

/
*****
*****
*Function      :      main
*Input parameters:      no parameters
*RETURNS       :      0 on success
*****
*****/

int main(void)
{
    unsigned int iVal, iNewVal, iNum;
    int iChoice;

    printf("\n*****")
);
    printf("\n*\tPROGRAM TO IMPLEMENT CIRCULAR RIGHT SHIFT\t*
\n");

    printf("*****");

    do
    {
        printf("\nEnter the value to be rotated\n");
        scanf("%u",&iVal);

        printf("\nEnter the number of positions by which the
value has to be rotated\n");
        scanf("%u",&iNum);

        iNewVal = fnRightShift(iVal, iNum);

        printf("\nThe value %u after right rotation by %u
bits = %u\n",iVal,iNum,iNewVal);

        printf("\nPress 1 to continue or any other key to
exit....\n");
        scanf("%d",&iChoice);
    }while(1 == iChoice);

    return 0;
}

/
*****
*****

```

```

*Function          : fnRightRot
*Description       : Function to perform Right Circular shift
by the number of bits specified
*Input parameters  :
*      unsigned int iX - value to be rotated
*      int iN         - no of positions by which rotation has to
be performed
*RETURNS          : resultant value after rotation has been
performed
*****
*****/

```

```

int fnRightShift(unsigned int iX , unsigned int iN)
{
    unsigned int iShift;
    int i;
    iShift = 1 << 31;

    for(i = 0; i < iN ; i++)
    {
        /*IF iX HAS AN 1 AT THE RIGHTMOST BIT IT HAS TO
        APPEAR AT THE LEFT END AFTER ROTATION*/

        if( iX % 2 )
        {
            iX = iX >> 1;
            iX |= iShift;
        }
        else /*OTHERWISE SIMPLY APPLY RIGHT SHIFT*/
        {
            iX = iX >> 1;
        }
    }
    return iX;
}

```