/*main.c file*/

```
#include <stdio.h>
#include <conio.h>
#include "macro1.h"

int main (void)
{
FILE *fpr,*fpi,*fpr1,*fpi1;
short var=0; //used for TOFLT
double var1; //used for TOFLT

short var2; //used for TOFIX
double var3;//used for TOFIX

int q,q2,type;
char ch;
long i=0;
q2=12; //Q format if u need to convert to and store in the file
printf("Pl.enter Q format:");
scanf("%d",&q);
printf("Enter 1 for FLT-FIX and 2 for FIX-FLT conversion:");
scanf("%d",&type);

fpr=fopen("real_in.txt","r");
fpi=fopen("imag_in.txt","r");

fpr1=fopen("real_out.txt","w");
fpi1=fopen("imag_out.txt","w");

while(1)
{
        ch=fgetc(fpr);
        if(type==1)
        {
                fscanf(fpr,"%lf",&var3);
                var2=TOFIX(var3,q);
                //FCONV(var2, q, q2)
                fprintf(fpr1,"%d\n",var2);
                if(ch==EOF)
                break;
                i=i+1;
        }
        else if(type==2)
```

```
        {
                fscanf(fpr,"%d",&var);
                var1=TOFLT(var,q);
                fprintf(fpr1,"%lf\n",var1);
                if(ch==EOF)
                break;
                i=i+1;
        }
        else
                printf("wrong type");
}

while(1)
{
        ch=fgetc(fpi);
        if(type==1)
        {
                fscanf(fpi,"%lf",&var3);
                var2=TOFIX(var3,q);
                //FCONV(var2, q, q2)
                fprintf(fpi1,"%d\n",var2);
                if(ch==EOF)
                break;
        }

        else if(type==2)
        {
                fscanf(fpi,"%d",&var);
                var1=TOFLT(var,q);
                fprintf(fpi1,"%lf\n",var1);
                if(ch==EOF)
                break;
        }
        else
                printf("wrong type");
        }

printf("I/Q packet length=%d",i);
return 0;
}
```

/macro1.h header file/

```
/* convert a from q1 format to q2 format */
#define FCONV(a, q1, q2) (((q2)>(q1)) ? (a)<<((q2)-(q1)) : (a)>>((q1)-(q2)))
```

```
/* convert to and from floating point */
#define TOFIX (d, q) ((int) ((d)*(double) (1<< (q))))
#define TOFLT (a, q) ((double) (a) / (double) (1<< (q)))
```