

### Programa 3 – Producto

funciones.h

```
#ifndef __FUNCIONES_H__
#define __FUNCIONES_H__

//Librerías de C
#include <stdio.h>
#include <stdlib.h>
//Librería que contiene los máximos y mínimos de los diferentes tipos de datos en c
#include <limits.h>
//Metodos
void leerCabeceras(char**);
void leerMuestras(short*,short*);
void leerMuestras2Canales(short*,short*,short*,short*);
void escribirArchivo(short*,int);
void escribirArchivo2Canales(short*,short*,int);
void calcularProducto(short*,short*);
void calcularProductoComplejos(short*,short*,short*,short*);
//Cabeceras primer archivo
int chunkid1;
int chunksize1;
int format1;
int subchunk1id1;
int subchunk1size1;
short audioformat1;
short numchannels1;
int samplerate1;
int byterate1;
short blockalign1;
short bitspersample1;
int subchunk2id1;
int subchunk2size1;
//Cabeceras segundo archivo
int chunkid2;
int chunksize2;
int format2;
int subchunk1id2;
int subchunk1size2;
short audioformat2;
short numchannels2;
int samplerate2;
int byterate2;
short blockalign2;
short bitspersample2;
int subchunk2id2;
```

```

int subchunk2size2;
//Archivo
FILE* entrada1;
FILE* entrada2;
FILE* salida;
//Variables para muestras primer archivo
int total_muestras1;
//Variables segundo archivo
int total_muestras2;
//Variables que se pueden reutilizar
short muestra;
short headers[37];
#endif

```

producto.c

```

#include"funciones.h"
int main(int argc, char *argv[])
{
    //Leo las cabeceras
    leerCabeceras(argv);
    if(numchannels1==1){
        //Defino variables para el primer archivo
        total_muestras1=subchunk2size1/blockalign1;
        short *muestras1=(short *)malloc(total_muestras1 * sizeof(short));
        //Defino variables para el segundo archivo
        total_muestras2=subchunk2size2/blockalign2;
        short *muestras2=(short *)malloc(total_muestras2 * sizeof(short));
        //Leo las muestras del primer archivos
        leerMuestras(muestras1,muestras2);
        //Calculo el producto
        calcularProducto(muestras1,muestras2);
    }else if(numchannels1==2){
        //Defino variables para el primer archivo
        total_muestras1=subchunk2size1/blockalign1;
        short *muestrasRe1=(short *)malloc(total_muestras1 * sizeof(short));
        short *muestrasIm1=(short *)malloc(total_muestras1 * sizeof(short));
        //Defino variables para el segundo archivo
        total_muestras2=subchunk2size2/blockalign2;
        short *muestrasRe2=(short *)malloc(total_muestras2 * sizeof(short));
        short *muestrasIm2=(short *)malloc(total_muestras2 * sizeof(short));
        //Leo las muestras del primer archivos
        leerMuestras2Canales(muestrasRe1,muestrasIm1,muestrasRe2,muestrasIm2);
        //Calculo el producto
        calcularProductoComplejos(muestrasRe1,muestrasIm1,muestrasRe2,muestrasIm2);
    }
}

```

```

}
void leerCabeceras(char ** argv){
    entrada1 = fopen(argv[1], "rb");
    entrada2 = fopen(argv[2], "rb");
    salida=fopen(argv[3], "wb");
    if(!entrada1 & !entrada2) {
        perror("\nFile opening failed");
        exit(0);
    }
    //Lectura de cabeceras primer archivo
    fread(&chunkid1,sizeof(int),1,entrada1);
    fread(&chunksize1,sizeof(int),1,entrada1);
    fread(&format1,sizeof(int),1,entrada1);
    fread(&subchunk1id1,sizeof(int),1,entrada1);
    fread(&subchunk1size1,sizeof(int),1,entrada1);
    fread(&audioformat1,sizeof(short),1,entrada1);
    fread(&numchannels1,sizeof(short),1,entrada1);
    fread(&samplerate1,sizeof(int),1,entrada1);
    fread(&byterate1,sizeof(int),1,entrada1);
    fread(&blockalign1,sizeof(short),1,entrada1);
    fread(&bitspersample1,sizeof(short),1,entrada1);
    fread(&subchunk2id1,sizeof(int),1,entrada1);
    fread(&subchunk2size1,sizeof(int),1,entrada1);
    //Lectura de cabeceras segundo archivo
    fread(&chunkid2,sizeof(int),1,entrada2);
    fread(&chunksize2,sizeof(int),1,entrada2);
    fread(&format2,sizeof(int),1,entrada2);
    fread(&subchunk1id2,sizeof(int),1,entrada2);
    fread(&subchunk1size2,sizeof(int),1,entrada2);
    fread(&audioformat2,sizeof(short),1,entrada2);
    fread(&numchannels2,sizeof(short),1,entrada2);
    fread(&samplerate2,sizeof(int),1,entrada2);
    fread(&byterate2,sizeof(int),1,entrada2);
    fread(&blockalign2,sizeof(short),1,entrada2);
    fread(&bitspersample2,sizeof(short),1,entrada2);
    fread(&subchunk2id2,sizeof(int),1,entrada2);
    fread(&subchunk2size2,sizeof(int),1,entrada2);
}
void leerMuestras(short *muestras1, short *muestras2){
    int i=0;
    //Lectura muestras primer archivo
    while (feof(entrada1) == 0)
    {
        if(i<total_muestras1){
            fread(&muestra,sizeof(short),1,entrada1);
            printf("Muestra %d del primer archivo:%hi\n",i+1,muestra);
            muestras1[i]=muestra;

```

```

        i++;
        //printf("Muestra %s: %d\n",i,muestras[i-1]);
    }else{
        fread(&headers,sizeof(short),37,entrada1);
        break;
    }
}
//Lectura muestras segundo archivo
i=0;
while (feof(entrada2) == 0)
{
    if(i<total_muestras2){
        fread(&muestra,sizeof(short),1,entrada2);
        printf("Muestra %d segundo archivo:%hi\n", muestra);
        muestras2[i]=muestra;
        i++;
    }else{
        fread(&headers,sizeof(short),37,entrada2);
        break;
    }
}
}
void leerMuestras2Canales(short *muestrasRe1,short* muestrasIm1,short* muestrasRe2,
short* muestrasIm2){
    int i=0;
    //Lectura de muestras del primer archivo
    while (feof(entrada1) == 0)
    {
        if(i<total_muestras1){
            fread(&muestrasRe1[i],sizeof(short),1,entrada1);
            fread(&muestrasIm1[i],sizeof(short),1,entrada1);
            i++;

        }else{
            fread(&headers,sizeof(short),37,entrada1);
            break;
        }
    }
    i=0;
    //Lectura de muestras del segundo archivo
    while (feof(entrada2) == 0)
    {
        if(i<total_muestras2){
            fread(&muestrasRe2[i],sizeof(short),1,entrada2);
            fread(&muestrasIm2[i],sizeof(short),1,entrada2);
            i++;
        }else{

```

```

        fread(&headers,sizeof(short),37,entrada2);
        break;
    }
}
}
}
void escribirArchivo(short* muestras,int decision){
    //Escribo las cabeceras del archivo de salida
    if (decision==0){
        //Se escriben las cabeceras del primer archivo
        fwrite(&chunkid1,sizeof(int),1,salida);
        fwrite(&chunksizel,sizeof(int),1,salida);
        fwrite(&format1,sizeof(int),1,salida);
        fwrite(&subchunk1id1,sizeof(int),1,salida);
        fwrite(&subchunk1sizel,sizeof(int),1,salida);
        fwrite(&audioformat1,sizeof(short),1,salida);
        fwrite(&numchannels1,sizeof(short),1,salida);
        fwrite(&samplerate1,sizeof(int),1,salida);
        fwrite(&byterate1,sizeof(int),1,salida);
        fwrite(&blockalign1,sizeof(short),1,salida);
        fwrite(&bitspersample1,sizeof(short),1,salida);
        fwrite(&subchunk2id1,sizeof(int),1,salida);
        fwrite(&subchunk2sizel,sizeof(int),1,salida);
        //Ahora escribo las muestras
        int i=0;
        for(i=0;i<total_muestras1;i++){
            fwrite(&muestras[i],sizeof(short),1,salida);
        }
        //Y por último los headers de goldwave
        for(i=0;i<37;i++){
            fwrite(&headers[i],sizeof(short),1,salida);
        }
    }else{
        //Se escriben las cabeceras del segundo archivo
        fwrite(&chunkid2,sizeof(int),1,salida);
        fwrite(&chunksizel,sizeof(int),1,salida);
        fwrite(&format2,sizeof(int),1,salida);
        fwrite(&subchunk1id2,sizeof(int),1,salida);
        fwrite(&subchunk1sizel,sizeof(int),1,salida);
        fwrite(&audioformat2,sizeof(short),1,salida);
        fwrite(&numchannels2,sizeof(short),1,salida);
        fwrite(&samplerate2,sizeof(int),1,salida);
        fwrite(&byterate2,sizeof(int),1,salida);
        fwrite(&blockalign2,sizeof(short),1,salida);
        fwrite(&bitspersample2,sizeof(short),1,salida);
        fwrite(&subchunk2id2,sizeof(int),1,salida);
        fwrite(&subchunk2sizel,sizeof(int),1,salida);
        //Ahora escribo las muestras
    }
}

```

```

    int i=0;
    for(i=0;i<total_muestras2;i++){
        fwrite(&muestras[i],sizeof(short),1,salida);
    }
    //Y por último los headers de goldwave
    for(i=0;i<37;i++){
        fwrite(&headers[i],sizeof(short),1,salida);
    }
}
}

void escribirArchivo2Canales(short* muestrasRe, short* muestrasIm,int decision){
    //Escribo las cabeceras del archivo de salida
    if (decision==0){
        //Se escriben las cabeceras del primer archivo
        fwrite(&chunkid1,sizeof(int),1,salida);
        fwrite(&chunksize1,sizeof(int),1,salida);
        fwrite(&format1,sizeof(int),1,salida);
        fwrite(&subchunk1id1,sizeof(int),1,salida);
        fwrite(&subchunk1size1,sizeof(int),1,salida);
        fwrite(&audioformat1,sizeof(short),1,salida);
        fwrite(&numchannels1,sizeof(short),1,salida);
        fwrite(&samplerate1,sizeof(int),1,salida);
        fwrite(&byterate1,sizeof(int),1,salida);
        fwrite(&blockalign1,sizeof(short),1,salida);
        fwrite(&bitspersample1,sizeof(short),1,salida);
        fwrite(&subchunk2id1,sizeof(int),1,salida);
        fwrite(&subchunk2size1,sizeof(int),1,salida);
        //Ahora escribo las muestras
        int i=0;
        for(i=0;i<total_muestras1;i++){
            fwrite(&muestrasRe[i],sizeof(short),1,salida);
            fwrite(&muestrasIm[i],sizeof(short),1,salida);
        }
        //Y por último los headers de goldwave
        for(i=0;i<37;i++){
            fwrite(&headers[i],sizeof(short),1,salida);
        }
    }else{
        //Se escriben las cabeceras del segundo archivo
        fwrite(&chunkid2,sizeof(int),1,salida);
        fwrite(&chunksize2,sizeof(int),1,salida);
        fwrite(&format2,sizeof(int),1,salida);
        fwrite(&subchunk1id2,sizeof(int),1,salida);
        fwrite(&subchunk1size2,sizeof(int),1,salida);
        fwrite(&audioformat2,sizeof(short),1,salida);
        fwrite(&numchannels2,sizeof(short),1,salida);
        fwrite(&samplerate2,sizeof(int),1,salida);
    }
}

```

```

fwrite(&byterate2,sizeof(int),1,salida);
fwrite(&blockalign2,sizeof(short),1,salida);
fwrite(&bitspersample2,sizeof(short),1,salida);
fwrite(&subchunk2id2,sizeof(int),1,salida);
fwrite(&subchunk2size2,sizeof(int),1,salida);
//Ahora escribo las muestras
int i=0;
for(i=0;i<total_muestras2;i++){
    fwrite(&muestrasRe[i],sizeof(short),1,salida);
    fwrite(&muestrasIm[i],sizeof(short),1,salida);
}
//Y por último los headers de goldwave
for(i=0;i<37;i++){
    fwrite(&headers[i],sizeof(short),1,salida);
}
}
}

void calcularProducto(short* muestras1, short* muestras2){
    int i;
    //Reviso que archivo tiene más muestras
    if(total_muestras1<=total_muestras2){
        //El segundo archivo tiene más
        puts("El segundo archivo tiene mas muestras");
        short *resultado=(short *)malloc(total_muestras2 * sizeof(short));
        for (i = 0; i < total_muestras2; i++)
        {
            if(i<total_muestras1){
                resultado[i]=(muestras1[i]*muestras2[i])/(SHRT_MAX+2);
                //printf("Muestra resultado %d: %hi\n",i+1,resultado[i]);
            }else{
                resultado[i]=0;
            }
        }
        //Escribo el resultado y cabeceras segundo archivo
        escribirArchivo(resultado,1);
    }else{
        //El primer archivo tiene más
        puts("El primer archivo tiene mas muestras");
        short *resultado=(short *)malloc(total_muestras1 * sizeof(short));
        for (i = 0; i < total_muestras1; i++)
        {
            if(i<total_muestras2){
                resultado[i]=(muestras1[i]*muestras2[i])/(SHRT_MAX+2);
            }else{
                resultado[i]=0;
            }
        }
    }
}

```

```

    //Escribo el resultado y cabeceras primer archivo
    escribirArchivo(resultado,0);
}

}

void calcularProductoComplejos(short* muestrasRe1, short* muestrasIm1, short*
muestrasRe2, short* muestrasIm2){
    int i;
    //Reviso que archivo tiene más muestras
    if(total_muestras1<=total_muestras2){
        //El segundo archivo tiene más
        puts("El segundo archivo tiene mas muestras o igual numero de muestras");
        short *resultadoRe=(short *)malloc(total_muestras2 * sizeof(short));
        short *resultadoIm=(short *)malloc(total_muestras2 * sizeof(short));
        //short resultadoRe[total_muestras2];
        //short resultadoIm[total_muestras2];
        for (i = 0; i < total_muestras2; i++)
        {
            if(i<total_muestras1){
                //Tenemos que hacer producto de complejos
                resultadoRe[i]=((muestrasRe1[i]*muestrasRe2[i])-
(muestrasIm1[i]*muestrasIm2[i]))/(SHRT_MAX*2);
                resultadoIm[i]=((muestrasRe1[i]*muestrasIm2[i])+(muestrasRe2[i]*muestrasIm1[i]))/(S
HRT_MAX*2);
                printf("Real:%d \n",resultadoRe[i]);
                printf("Imaginaria: %d\n", resultadoIm[i]);
            }else{
                resultadoRe[i]=0;
                resultadoIm[i]=0;
            }
        }
        //Escribo el resultado y cabeceras segundo archivo
        escribirArchivo2Canales(resultadoRe,resultadoIm,1);
    }else{
        //El primer archivo tiene más
        puts("El primer archivo tiene mas muestras");
        short *resultadoRe=(short *)malloc(total_muestras1 * sizeof(short));
        short *resultadoIm=(short *)malloc(total_muestras1 * sizeof(short));
        for (i = 0; i < total_muestras1; i++){
            if(i<total_muestras2){
                //Tenemos que hacer producto de complejos
                resultadoRe[i]=((muestrasRe1[i]*muestrasRe2[i])-
(muestrasIm1[i]*muestrasIm2[i]))/(SHRT_MAX*2);
                resultadoIm[i]=((muestrasRe1[i]*muestrasIm2[i])+(muestrasRe2[i]*muestrasIm1[i]))/(S
HRT_MAX*2);
            }else{
                resultadoRe[i]=0; resultadoIm[i]=0;
            }
        }
        escribirArchivo2Canales(resultadoRe,resultadoIm,0);
    }
}

```