

## Programa 1- Cambiar volumen a la mitad

funciones.h

```
#ifndef __FUNCIONES_H__
#define __FUNCIONES_H__

    //Librerías de C
    #include <stdio.h>
    #include <stdlib.h>

    //Métodos
    void leerCabeceras(char**);
    void leerMuestras(short*);
    void dividirMuestras(short*,float);
    void escribirArchivo(short*);

    //Cabeceras
    int chunkid;
    int chunksize;
    int format;
    int subchunk1id;
    int subchunk1size;
    short audioformat;
    short numchannels;
    int samplerate;
    int byterate;
    short blockalign;
    short bitspersample;
    int subchunk2id;
    int subchunk2size;

    //Archivos
    FILE* entrada;
    FILE* salida;

    //Variables para muestras
    short muestra;
    int total_muestras;
    short headers[37];
#endif
```

half.c

```
#include "funciones.h"
int main(int argc, char *argv[]){
    //Leo las cabeceras
    leerCabeceras(argv);
    //Defino variables
    total_muestras=subchunk2size/blockalign;
    short *muestras=(short *)malloc(total_muestras * sizeof(short));
    //Leo las muestras
    leerMuestras(muestras);
    //Divido el volumen a la mitad
    dividirMuestras(muestras,.5);
    //Escribo el archivo
    escribirArchivo(muestras);
}
void leerCabeceras(char ** argv){
    //Abrir archivos
    entrada = fopen(argv[1], "rb");
    salida=fopen(argv[2], "wb");
    if(!entrada) {
        perror("\nFile opening failed");
        exit(0);
    }
    fread(&chunkid,sizeof(int),1,entrada);
    fread(&chunksize,sizeof(int),1,entrada);
    fread(&format,sizeof(int),1,entrada);
    fread(&subchunk1id,sizeof(int),1,entrada);
    fread(&subchunk1size,sizeof(int),1,entrada);
    fread(&audioformat,sizeof(short),1,entrada);
    fread(&numchannels,sizeof(short),1,entrada);
    fread(&samplerate,sizeof(int),1,entrada);
    fread(&byterate,sizeof(int),1,entrada);
    fread(&blockalign,sizeof(short),1,entrada);
    fread(&bitspersample,sizeof(short),1,entrada);
    fread(&subchunk2id,sizeof(int),1,entrada);
    fread(&subchunk2size,sizeof(int),1,entrada);
}
void leerMuestras(short *muestras){
    int i=0;
    while (feof(entrada) == 0)
    {
        if(i<total_muestras){
            fread(&muestra,sizeof(short),1,entrada);
            muestras[i]=muestra;
            i++;
            //printf("Muestra %s: %d\n",i,muestras[i-1]);
        }
    }
}
```

```

    }else{
        fread(&headers,sizeof(short),37,entrada);
        break;
    }
}
}
}
void dividirMuestras(short *muestras, float factor){
    int i;
    for (i = 0; i < total_muestras; i++)
    {
        muestras[i]*=factor;
    }
}
void escribirArchivo(short* muestras){
    //Escribo el archivo
    fwrite(&chunkid,sizeof(int),1,salida);
    fwrite(&chunksize,sizeof(int),1,salida);
    fwrite(&format,sizeof(int),1,salida);
    fwrite(&subchunk1id,sizeof(int),1,salida);
    fwrite(&subchunk1size,sizeof(int),1,salida);
    fwrite(&audioformat,sizeof(short),1,salida);
    fwrite(&numchannels,sizeof(short),1,salida);
    fwrite(&samplerate,sizeof(int),1,salida);
    fwrite(&byterate,sizeof(int),1,salida);
    fwrite(&blockalign,sizeof(short),1,salida);
    fwrite(&bitspersample,sizeof(short),1,salida);
    fwrite(&subchunk2id,sizeof(int),1,salida);
    fwrite(&subchunk2size,sizeof(int),1,salida);
    //Ahora escribo las muestras
    int i=0;
    for(i=0;i<total_muestras;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);
    }
}

```