Programa que identifica que tecla del código DTMF fue tocada

funciones.h

```
#ifndef __FUNCIONES_H__
#define __FUNCIONES_H__
   //Librerías de C
   #include <stdio.h>
   #include <stdlib.h>
   #include <math.h>
   //Librería que contiene los máximos y mínimos de los diferentes tipos de datos en c
   #include <limits.h>
   //Libreria para conocer tiempo de ejecución
   #include <time.h>
   //Metodos
   void leerCabeceras(char**);
   void escribirArchivo(short*,short*);
   void leerMuestras(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;
   //Archivo
   FILE* entrada;
   FILE* salida;
   //Variables para muestras
   short muestra;
   int total_muestras;
   short headers[37];
   //Métodos TDF
   #define PI acos(-1.0)//Defino la constante PI
   void calcularTDF(short*);
   void obtenerDTMF(short*,short*);
   float duracion;
   int aux1,aux2;
   int amp=1000;
   int bandera=0;
   //Arreglos de frecuencias
```

```
    int f_1[2] = {697,1209};
    int f_2[2] = {697,1336};
    int f_3[2] = {697,1477};
    int f_A[2] = {697,1633};
    int f_4[2] = {770,1209};
    int f_5[2] = {770,1336};
    int f_6[2] = {770,1477};
    int f_B[2] = {770,1633};
    int f_7[2] = {852,1209};
    int f_8[2] = {852,1336};
    int f_9[2] = {852,1477};
    int f_C[2] = {852,1633};
    int f_ASTE[2] = {941,1209};
    int f_0[2] = {941,1336};
    int f_GATO[2] = {941,1477};
    int f_D[2] = {941,1633};
#endif
```

dtmf.c

```
#include"funciones.h"
int main(int argc, char *argv[]){
    //Leo las cabeceras
    leerCabeceras(argv);
    //Defino variables
    total_muestras=subchunk2size/blockalign;
    printf("Total de muestras: %d\n", total_muestras);
    short *muestras=(short *)malloc(total_muestras * sizeof(short));
    //Leo las muestras
    leerMuestras(muestras);
    //Calculo la TDF
    calcularTDF(muestras);
}
void leerCabeceras(char ** argv){
    //Archivo de entrada
    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++;
        }else{
            fread(&headers,sizeof(short),37,entrada);
            break;
        }
    }
}
void escribirArchivo(short* muestrasRe,short* muestrasIm){
    //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(&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 calcularTDF(short* muestras){
    //Aquí va el algoritmo para la TDF
```

```
short *Xre=(short *)malloc(total_muestras * sizeof(short));
short *Xim=(short *)malloc(total_muestras * sizeof(short));
short *magnitud=(short *)malloc(total_muestras * sizeof(short));
short *fase=(short *)malloc(total_muestras * sizeof(short));
//Variables para obtener tiempo de ejecución
clock_t inicio, final;
double total;
inicio = clock();
//Algoritmo TDF
int n,k;
for (k = 0; k < total_muestras; k++)    {
   Xre[k]=0;
   Xim[k]=0;
   for (n = 0; n < total_muestras; n++)       {
      Xre[k]+=(muestras[n]/total_muestras)*cos(2*k*n*PI/total_muestras);
      Xim[k]-=(muestras[n]/total_muestras)*sin(2*k*n*PI/total_muestras);
   }
}
//Obtener tiempo e imprimir
final = clock();
total = (double)(final - inicio) / CLOCKS_PER_SEC;
printf("Tiempo de ejecucion: %f\n", total);
//Ahora calcularé la magnitud de la transformada
for (k = 0; k < total_muestras; k++){
   magnitud[k]=sqrt(pow(Xre[k],2)+pow(Xim[k],2));
}
//La fase
float valor=180.0/PI;
for (k = 0; k < total_muestras; k++){
   if (magnitud[k]>1000){
      //atan nos devuelve un valor en radianes, hay que pasarlo a grados
      if(Xre[k]==0){
         if (Xim[k]==0){
            fase[k]=0;
         }else if (Xim[k]<0){//Xim es negativa
            fase[k]=-90;
         }else{
            fase[k]=90;
         }
      }else{
         fase[k]=atan(Xim[k]/Xre[k])*valor;
      }
   }else{
      fase[k]=0;
   }
   fase[k]=(fase[k]*SHRT_MAX)/180;//Para que se pueda ver en goldwave
}
```

```
    //La salida ahora sera un archivo tipo estereo (2 canales)
    //Por lo cual hay que cambiar el numero de canales del archivo
    //y todas las demas cabeceras que dependan de esta
    chunksize-=subchunk2size;
    numchannels*=2;
    byterate*=numchannels;
    blockalign*=numchannels;
    subchunk2size*=numchannels;
    chunksize+=subchunk2size;
    escribirArchivo(magnitud,fase);
    obtenerDTMF(Xre,Xim);}
void obtenerDTMF(short *Xre,short *Xim){
    //Obtengo la duración del archivo
    duracion=(float)total_muestras/(float)samplerate;
    printf("Duracion del archivo: %f\n", duracion);
    int i;
    for(i=0;i<total_muestras/2;i++){
          if(Xre[i]>amp)
          printf("Xre[%d] = %d\n", i, Xre[i]);
    }
    //Aquí reviso que tonos fueron identificados
    //Tecla: 1, frecuencia baja=697Hz, alta=1209Hz
    aux1=(int)floor((float)f_1[0]*duracion);
    aux2=(int)floor((float)f_1[1]*duracion);
    if(Xre[aux1]>amp && Xre[aux2]>amp){
       puts("El tono corresponde a un 1");
       bandera=1;
    }
    aux1=(int)floor((float)f_2[0]*duracion);
    aux2=(int)floor((float)f_2[1]*duracion);
    //Tecla: 2, frecuencia baja=697Hz, alta=1336Hz
    if(Xre[aux1]>amp && Xre[aux2]>amp){
       puts("El tono corresponde a un 2");
       bandera=1;
    }
    aux1=(int)floor((float)f_3[0]*duracion);
    aux2=(int)floor((float)f_3[1]*duracion);
    //Tecla: 3, frecuencia baja=697Hz, alta=1477Hz
    if(Xre[aux1]>amp && Xre[aux2]>amp){
       puts("El tono corresponde a un 3");
       bandera=1;
    }
    aux1=(int)floor((float)f_A[0]*duracion);
    aux2=(int)floor((float)f_A[1]*duracion);
    //Tecla: A, frecuencia baja=697Hz, alta=1633Hz
    if(Xre[aux1]>amp && Xre[aux2]>amp){
       puts("El tono corresponde a una A");
```

```
    bandera=1;
  }
  aux1=(int)floor((float)f_4[0]*duracion);
  aux2=(int)floor((float)f_4[1]*duracion);
  //Tecla: 4, frecuencia baja=770Hz, alta=1209Hz
  if(Xre[aux1]>amp && Xre[aux2]>amp){
    puts("El tono corresponde a un 4");
    bandera=1;
  }
  aux1=(int)floor((float)f_5[0]*duracion);
  aux2=(int)floor((float)f_5[1]*duracion);
  //Tecla: 5, frecuencia baja=770Hz, alta=1336Hz
  if(Xre[aux1]>amp && Xre[aux2]>amp){
    puts("El tono corresponde a un 5");
    bandera=1;
  }
  aux1=(int)floor((float)f_6[0]*duracion);
  aux2=(int)floor((float)f_6[1]*duracion);
  //Tecla: 6, frecuencia baja=770Hz, alta=1477Hz
  if(Xre[aux1]>amp && Xre[aux2]>amp){
    puts("El tono corresponde a un 6");
    bandera=1;
  }
  aux1=(int)floor((float)f_B[0]*duracion);
  aux2=(int)floor((float)f_B[1]*duracion);
  //Tecla: B, frecuencia baja=770Hz, alta=1633Hz
  if(Xre[aux1]>amp && Xre[aux2]>amp){
    puts("El tono corresponde a una B");
    bandera=1;
  }
  aux1=(int)floor((float)f_7[0]*duracion);
  aux2=(int)floor((float)f_7[1]*duracion);
  //Tecla: 7, frecuencia baja=852Hz, alta=1209Hz
  if(Xre[aux1]>amp && Xre[aux2]>amp){
    puts("El tono corresponde a un 7");
    bandera=1;
  }
  aux1=(int)floor((float)f_8[0]*duracion);
  aux2=(int)floor((float)f_8[1]*duracion);
  //Tecla: 8, frecuencia baja=852Hz, alta=1336Hz
  if(Xre[aux1]>amp && Xre[aux2]>amp){
    puts("El tono corresponde a un 8");
    bandera=1;
  }
  aux1=(int)floor((float)f_9[0]*duracion);
  aux2=(int)floor((float)f_9[1]*duracion);
  //Tecla: 9, frecuencia baja=852Hz, alta=1477Hz
```

```
    if(Xre[aux1]>amp && Xre[aux2]>amp){
        puts("El tono corresponde a un 9");
        bandera=1;
    }
    aux1=(int)floor((float)f_C[0]*duracion);
    aux2=(int)floor((float)f_C[1]*duracion);
    //Tecla: C, frecuencia baja=852Hz, alta=1633Hz
    if(Xre[aux1]>amp && Xre[aux2]>amp){
        puts("El tono corresponde a una C");
        bandera=1;
    }
    aux1=(int)floor((float)f_ASTE[0]*duracion);
    aux2=(int)floor((float)f_ASTE[1]*duracion);
    //Tecla: *, frecuencia baja=941Hz, alta=1209Hz
    if(Xre[aux1]>amp && Xre[aux2]>amp){
        puts("El tono corresponde a un *");
        bandera=1;
    }
    aux1=(int)floor((float)f_0[0]*duracion);
    aux2=(int)floor((float)f_0[1]*duracion);
    //Tecla: 0, frecuencia baja=941Hz, alta=1336Hz
    if(Xre[aux1]>amp && Xre[aux2]>amp){
        puts("El tono corresponde a un 0");
        bandera=1;
    }
    aux1=(int)floor((float)f_GATO[0]*duracion);
    aux2=(int)floor((float)f_GATO[1]*duracion);
    //Tecla: #, frecuencia baja=941Hz, alta=1477Hz
    if(Xre[aux1]>amp && Xre[aux2]>amp){
        puts("El tono corresponde a un #");
        bandera=1;
    }
    aux1=(int)floor((float)f_D[0]*duracion);
    aux2=(int)floor((float)f_D[1]*duracion);
    //Tecla: D, frecuencia baja=941Hz, alta=1633Hz
    if(Xre[aux1]>amp && Xre[aux2]>amp){
        puts("El tono corresponde a una D");
        bandera=1;
    }
    if(!bandera){
          puts("Ningun tono detectado\n");
    }
}
```