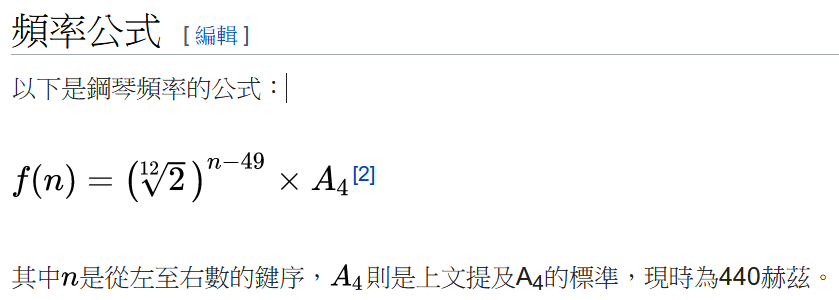
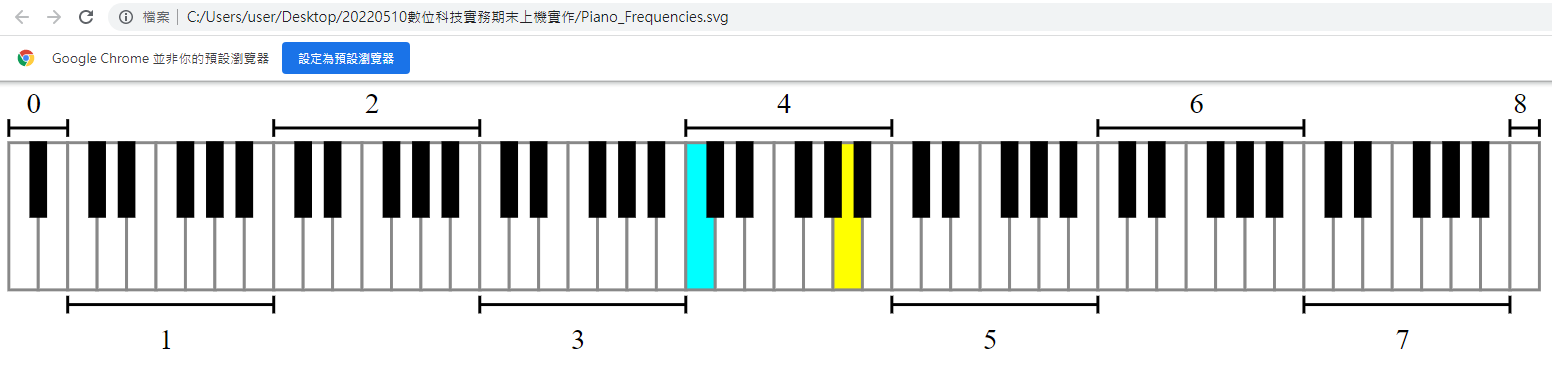
20220510數位科技實務期末上機實作用C語言製作一首簡譜歌曲的WAV檔(有標青色底處為操作處)

(1A) 鋼琴鍵頻率是指[鋼琴](https://zh.wikipedia.org/wiki/鋼琴)上每一個鍵所產生的聲音[頻率](https://zh.wikipedia.org/wiki/頻率)，以[赫茲](https://zh.wikipedia.org/wiki/赫茲)為單位。在88鍵鋼琴上，第**49**鍵（亦稱A4）是現行的調音標準440赫茲，亦稱[A440](https://zh.wikipedia.org/wiki/A440)。



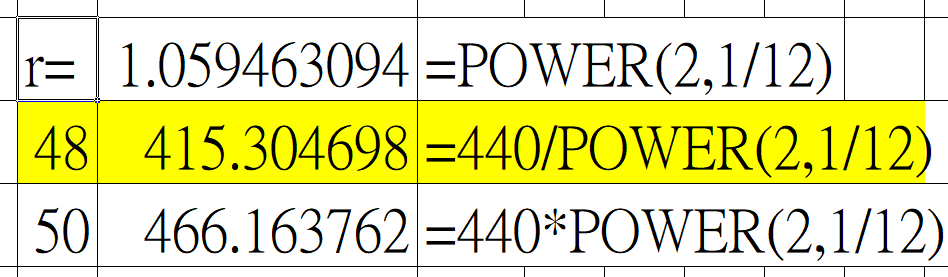
(1B)點下圖**超連結SVG檔**可觀看每一鍵的頻率，請查出**你座號數**的白色按鍵的聲音頻率(例如座號3對應第3個白色按鍵聲音頻率為32.70319566…)並截圖如下(如下88鍵鋼琴，以數字顯示八度和[中央C](https://zh.wikipedia.org/wiki/中央C) (青色) 以及 [A440](https://zh.wikipedia.org/wiki/A440) (黃色))。



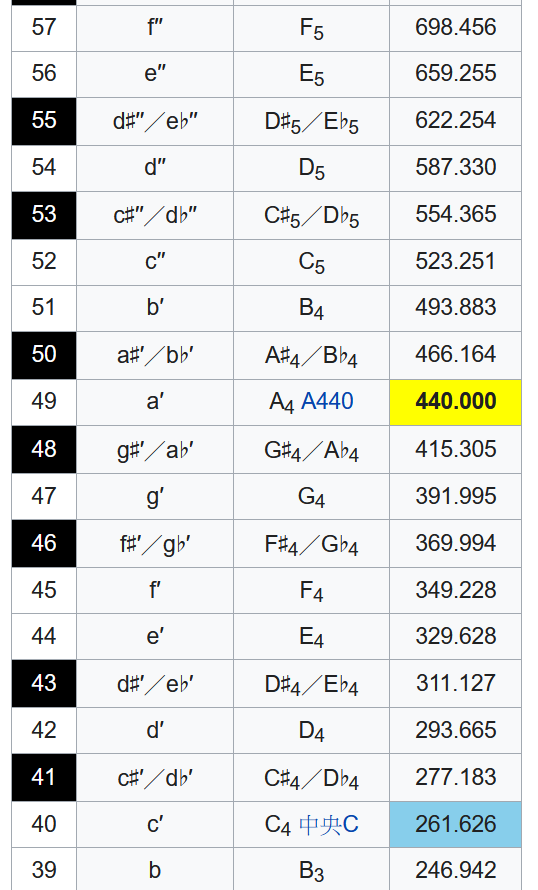
<https://upload.wikimedia.org/wikipedia/commons/2/2e/Piano_Frequencies.svg>

(1C)參考<http://lfwiki.kmvs.km.edu.tw/lftree/view_article.php?article_id=775> 中的(20220421鋼琴鍵音頻推算.xlsx) 可用EXCEL公式(=POWER(2,1/12))計算出二相鄰鋼琴鍵音頻的公比r為1.059463094

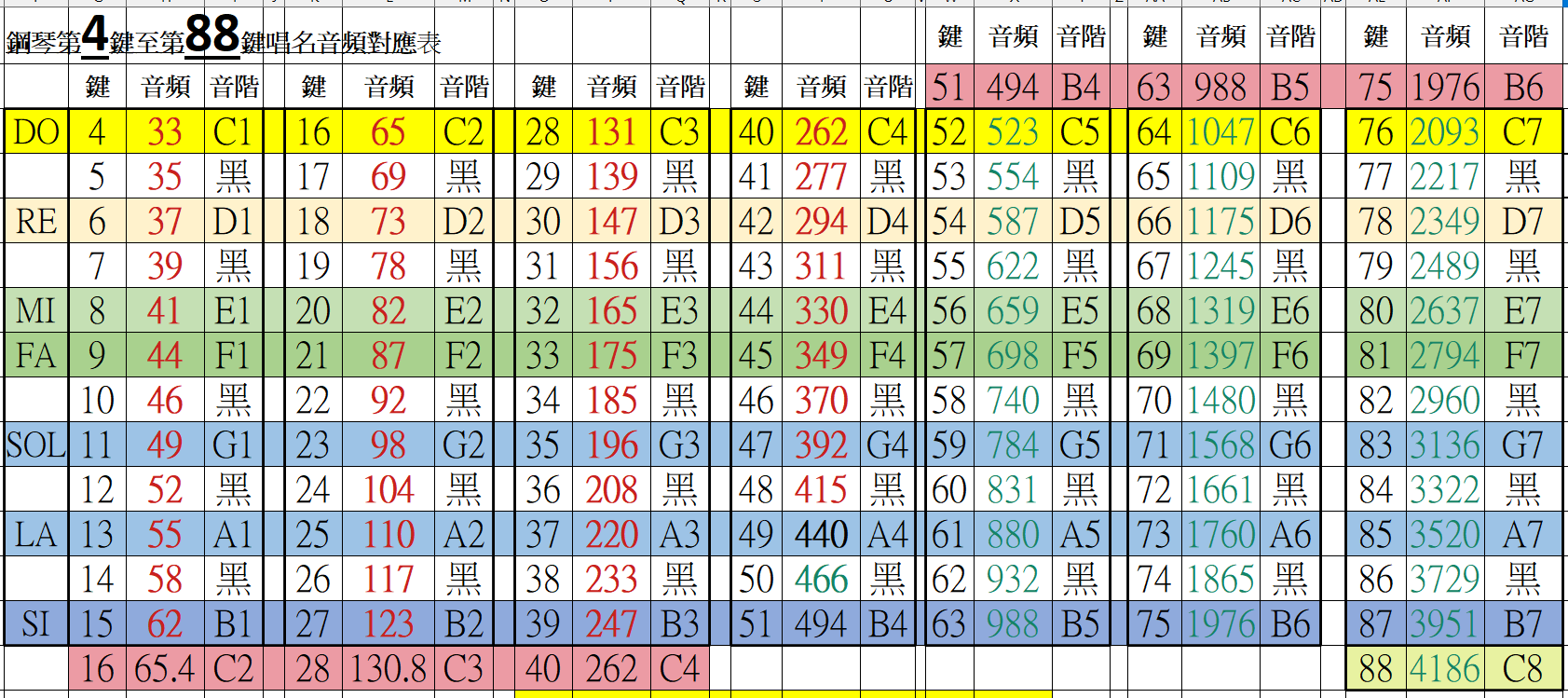
第48個琴鍵的音頻為415.304698，第50個琴鍵的音頻為466.163762如下：



(1D)如下為(1B)中的第39個至第57個鋼琴鍵的聲音頻率及符號列表：



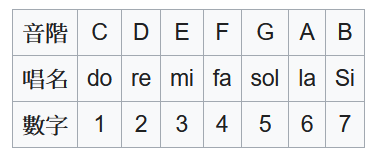
(1E)[**鋼琴第4鍵至第88鍵唱名音頻對應**表](http://lfwiki.kmvs.km.edu.tw/lftree/uploads/20220421鋼琴鍵音頻推算.ods)



(2A) [維基百科簡譜條目](https://zh.wikipedia.org/wiki/簡譜)摘要：

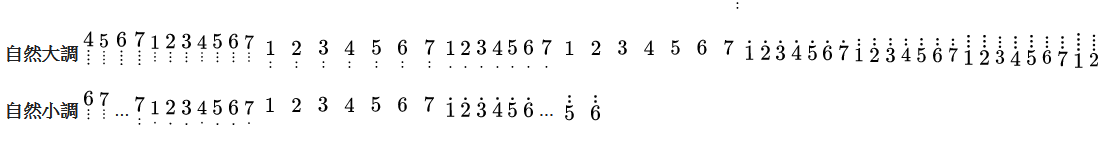
**簡譜**（英語：**numbered musical notation**）為[記譜法](https://zh.wikipedia.org/wiki/记谱法)之一，主要以[數字](https://zh.wikipedia.org/wiki/數字)作表達，亦稱為**數字譜**。其起源於[18世紀](https://zh.wikipedia.org/wiki/18世紀)的[法國](https://zh.wikipedia.org/wiki/法國)[[1]](https://zh.wikipedia.org/wiki/簡譜" \l "cite_note-南明離火-1)，後經[德國人](https://zh.wikipedia.org/wiki/德國人)改良，遂成今日之貌。 另外，在德文裡，其名為「*Ziffernsystem*」，意若「數字系統」（number system）

### (2B)音符：[音符](https://zh.wikipedia.org/wiki/音符)用數字1至7表示，請參考如下音階、唱名與音符數字的對應表：



### (2C) 八度：

### 如果是高一個[八度](https://zh.wikipedia.org/wiki/八度)，就會在數字上方加上一點。如果是低一個八度，就會數字下方加上一點。在中間的那一個[八度](https://zh.wikipedia.org/wiki/八度)就什麼也不用加。如果要再高一個八度，就在上方垂直加上兩點；要再低一個八度，就在下方垂直加上兩點，如此類推。見下表所示：



### (2D) 音長

通常**只有數字的是**[**四分音符**](https://zh.wikipedia.org/wiki/四分音符)。**數字下加一條橫線，就可令四分音符的長度減半，即成為**[**八分音符**](https://zh.wikipedia.org/wiki/八分音符)；**兩條橫線可令八分音符的長度減半，即成為**[**十六分音符**](https://zh.wikipedia.org/wiki/十六分音符)，以此類推；簡單來説，下加橫線數目與五線譜的符尾數目相對應。**數字後方的橫線延長音符，每加一條橫線延長一個**[**四分音符**](https://zh.wikipedia.org/wiki/四分音符)**的長度**。

正如五線譜的[附點](https://zh.wikipedia.org/wiki/附点)一樣，數字後方加一點會將音符長度增加一半。如果低八度的下加點與音長的下加線同時使用，則點在線的下面。

### (2E) 休止符:[休止符](https://zh.wikipedia.org/wiki/休止符)用「0」來表示。

(3)下載[Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe](http://lfwiki.kmvs.km.edu.tw/tree/Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe)(約49MB，**如系統防護軟體啟用中可能無法下載**)並執行，以裝妥DEV-C++ C語言編譯開發環境(SDK)。

(4A)WIN鍵/執行DEV-C++/按CTRL+N編寫如下程式碼並存為 d:\WAV-1.cpp

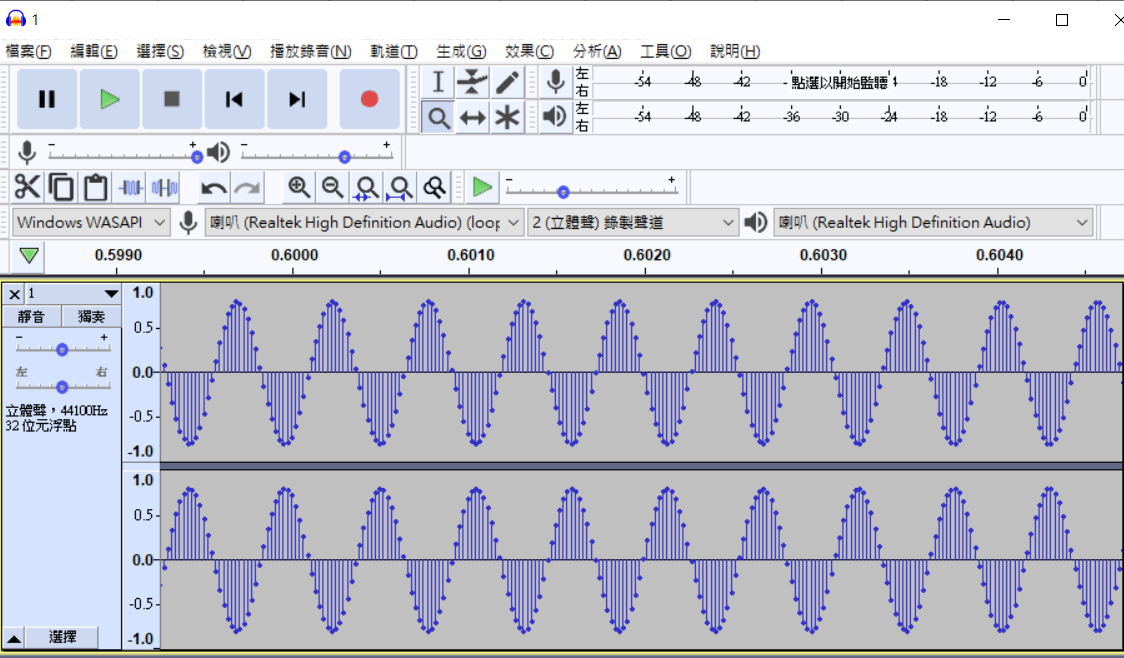
|  |
| --- |
| **#include <stdio.h>**  **#include <string.h>**  **#include <iostream>**  **#include <queue>**  **#include <math.h>**  **#define PI 3.14159265 //圓周率PI**  **using namespace std;**  **//宣告定義一個有44個位元組的WAV音訊檔的檔頭結構wavH如下：**  **typedef struct{ //wav Header**  **char ChunkID[4]; //00-03 區塊描述**  **unsigned int ChunkSize; //04-07區塊長度**  **char FormatTag[4]; //08-11 wave音訊檔格式**  **char SubChunk1ID[4]; //12-15 子區塊1描述**  **unsigned int SubChunk1Size; //16-19子區塊1長度**  **unsigned short AudioFormat; //20-21 音訊編碼格式，值1為PCM格式**  **unsigned short NumChannels; //22-23 聲道數**  **unsigned int SampleRate; //24-27 取樣頻率**  **unsigned int ByteRate; //28-31 每秒位元組率**  **unsigned short BlockAlign; //32-33對齊位元組數**  **unsigned short BitsPerSample; //34-35 每個樣本點的位元數**  **char SubChunk2ID[4]; //36-39子區塊2描述**  **unsigned int SubChunk2Size; //40-43子區塊2長度**  **} wavH;**  **wavH h; //宣告1個為wavH結構的變數h**  **queue <unsigned char> q; //宣告資料型態為無正負號的字元佇列q**  **//定義可將ts秒的第n個鋼琴鍵的聲音取樣點加入佇列q中的函數AddPitchSymbol2Q如下：**  **void AddPitchSymbol2Q(double ts, int n){**  **int c=0; //宣告初值為0的計數變數c**  **unsigned char x1,x2; //以佔1位元組unsigned char變數x1及x2分別來記錄第1、2聲道取樣點值**  **double dt=6.28/44100,t=0; // 以dt值記二個相鄰取樣點時間差，取樣時間t的初值為0**  **double r= pow(2,1.0/12); //宣告變數r, 其值為由低而高二相鄰鋼琴鍵音頻的公比**  **double bf=440\*pow(r,n-49); //宣告變數bf, 其值為依公式算出的第n個鋼琴鍵的音頻**  **double w=2\*PI\*bf; //宣告變數w, ,其值為第n個鋼琴鍵音頻bf的2PI倍**  **while (c < 44100\*(ts-0.01)){ //取樣(ts-0.01)秒的點：**  **x1=128\*sin(w\*t)\*4/5 +128; //計算頻率bf的聲音振動正弦波在第1聲道的取樣點值x1**  **x2=-x1; //讓第2聲道的取樣點值x2為-x1**  **q.push(x1); //將x1推入佇列q**  **q.push(x2); //將x2推入佇列q**  **t=t+dt; //將取樣點時間t調增dt**  **c=c+1; //將c值調增1**  **} //當迴圈下界**    while (c < 44100\*ts){ //取樣最後0.01秒的點：  q.push(128); //128為8位元聲音振幅取樣值的中間數，即靜默值  q.push(128); //第2聲道值也為128  t+=dt;  c=c+1;  }  **h.SubChunk2Size+=h.ByteRate\*ts; //調整檔頭變數h的聲音資料區(子區塊2)的大小。**  **h.ChunkSize+=h.ByteRate\*ts; //調整檔頭變數h的聲音區塊長度的大小。**  printf("Add ts=%.3lf %dth piano key f=%.2lf pitch symbol to Q\n",ts,n,bf);  **//顯示加入q的音及秒數**  **} // AddPitchSymbol2Q函數結束**  **//將佇列q的聲音取樣點連同WAV檔頭結構h寫入outfile**  **void writeQ(wavH h,queue <unsigned char> q,FILE \*outfile) {**  **unsigned char x1,x2; //宣告各佔1位元組的區域變數 x1及x2，用來暫存1位元組的聲音取樣點值**  **fwrite(&h, sizeof( wavH ), 1, outfile); //將wavH結構變數h的內容寫入到outfile**  **while (not q.empty()) { //當迴路上邊界，當佇列q還沒有空則執行當迴路內部**  **x1=q.front(); q.pop(); //將佇列q前端元素值拷給x1並移除之。**  **fwrite(&x1, sizeof(unsigned char), 1, outfile); //將x1寫入outfile**  **x2=q.front(); q.pop(); //將佇列q前端元素值拷給x2並移除之。**  **fwrite(&x2, sizeof(unsigned char), 1, outfile); //將x2寫入outfile**  **} //當迴路下邊界**  **} //writeQ函數結束**  **int main ( ) { //主程式開始**  **//如下ut值請用「0.3 + 你的座號/100，例如1號為0.3 +0.01」**  **double ut = 0.3 +0.01; // 用ut值表單位時間0.3秒( 1個**[**四分音符**](https://zh.wikipedia.org/wiki/四分音符)**的音長)**  **double bf; // 宣告bf來記音頻值**  **unsigned char x1,x2; //宣告主程式中各佔1位元組的區域變數 x1及x2**  **//宣告outfile檔案指標變數並用二元輸出模式開啟1.wav**  **FILE \*outfile = fopen("1.wav", "wb");**  //開始設定檔頭h的內容  **char st[5]="RIFF"; //宣告1個可放5個字元(char)的字元陣列變數st，其初值為"RIFF"**  **memcpy(&h.ChunkID,st,4); //00—03 //將st字元陣列的前4個字元拷給h.ChunkID**  **sprintf(st,"WAVE"); //08—11 設定st字串值為" WAVE "**  **memcpy(&h.FormatTag,st,4); //將st字元陣列的前4個字元拷給h.FormatTag**  **sprintf(st,"fmt "); //12—15 //設定st字串值為"fmt "**  **memcpy(&h.SubChunk1ID,st,4); //將st字元陣列的前4個字元拷給h. SubChunk1ID**  **h.SubChunk1Size=16; //16-19 設定檔頭第1區塊長度為16**  **h.AudioFormat =1; //20-21 設定音訊編碼格式，值1為PCM格式**  **h.NumChannels=2; //22-23 設定聲道數為2**  **h.SampleRate=44100; //24-27 設定取樣頻率為每秒44100點**  **h.ByteRate=h.SampleRate\*h.NumChannels; //28-31 設定每秒位元組率為聲道數\*取樣頻率**  **h.BlockAlign=2; //32-33 設定對齊位元組數為2個BYTE，每一取樣點佔2個位元組**  **h.BitsPerSample=8; //34-35 設定每一聲道取樣點的位元數**  **sprintf(st,"data"); //36—39 設定st字串值為" data "**  **memcpy(&h.SubChunk2ID,st,4); //將st字元陣列的前4個字元拷給h. SubChunk2ID**  **h.SubChunk2Size=0; //40-43 設定子區塊2(取樣點資料區塊)長度的初值為0**  **h.ChunkSize=h.SubChunk2Size+36; //04–07 設定取樣點資料區塊為區塊2長度+36**  //初步完成設定檔頭h的內容  **AddPitchSymbol2Q(ut,40); //Do:C4 加入ut秒(1個**[**四分音符**](https://zh.wikipedia.org/wiki/四分音符)**的音長)的第40鍵Do的音到佇列Q中**  **AddPitchSymbol2Q(ut,42); //Re:D4 同上再加入ut秒的第42鍵Re到Q**  **AddPitchSymbol2Q(ut,44); //Mi:E4 同上再加入ut秒的第44鍵Mi到Q**  **AddPitchSymbol2Q(ut,45); //Fa:F4 同上再加入ut秒的第45鍵Fa到Q**  **writeQ(h,q,outfile); //將準備妥當的wav檔頭h及佇列Q中的取樣點寫入outfile**  **printf("SAVE %.3lf second Music Q to 1.wav ok!\n", h.SubChunk2Size/(double)h.ByteRate);**  **fclose(outfile); //關閉輸出檔outfile**  **return (0);**  **} //主程式結束** |

註1：一個**unsigned char**型態的**無正負字元變數**佔用1個位元組的空間。

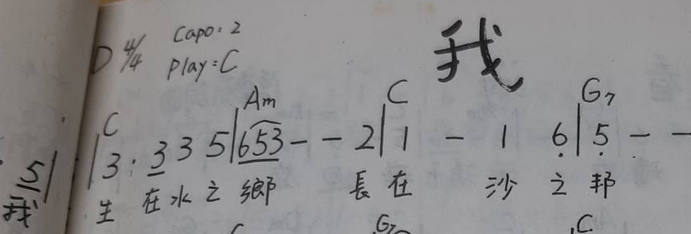
(4b)按F11編譯並執行wav-1.cpp結果如下：

|  |
| --- |
|  |

(5a)請確認已裝妥AudaCity聲音訊號編輯軟體  
(5b)用AudaCity打開(4B)所生成的WAV音樂檔1.wav並用上方放大鏡鈕放大音檔波形如下：



(6A) 將(4A)的 d:\WAV-1.cpp另存為d:\WAV-2.cpp，

並調整AddPitchSymbol2Q函數中的x1頻率產生公式如綠色底色處，並在如下**writeQ(h,q,outfile);指令**之前增加可產生如下簡譜第1句的音樂訊號的紫色底色處指令(請參考1E:**鋼琴第4鍵至第88鍵唱名音頻對應**表)，並修改輸出的WAV檔名為2.wav，並回存WAV-2.cpp。

註：如上簡譜解譯如下： 47(1)SOL:G4我、47(3/2)ME:E4 生、 44(1/2)ME:E4 在、 44(1)ME:E4 水、 47(1)SOL:G4 之、 49(1/3)La:A4 鄉、47(1/3)SOL:G4、44(7/3)ME:E4。

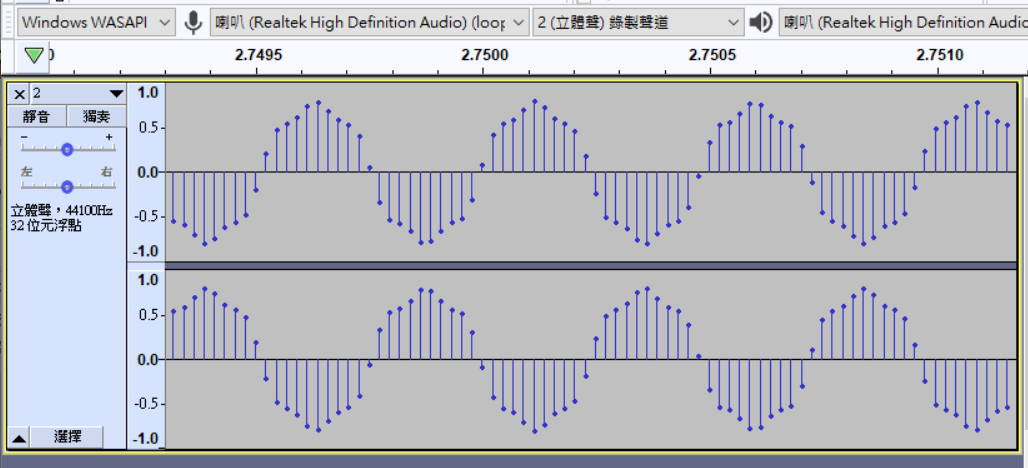
|  |
| --- |
| :  //如下為基頻bf及3倍基頻及5倍基頻組成的正弦聲波在時刻t的取樣點  x1=128\*(sin(w\*t)+0.1\*sin(3\*w\*t)+0.1\*sin(5\*w\*t))\*0.8 +128;  AddPitchSymbol2Q(ut/2,47); //SOL:G4 我，對第47號鍵音波取樣ut/2秒的點(每秒44100點)並加入q  AddPitchSymbol2Q(ut+ut/2,44); //ME:E4 生 // ut為1個[**四分音符**](https://zh.wikipedia.org/wiki/四分音符)**的音長**  AddPitchSymbol2Q(ut/2,44); //ME:E4 在  AddPitchSymbol2Q(ut,44); //ME:E4 水  AddPitchSymbol2Q(ut,47); //SOL:G4 之  AddPitchSymbol2Q(ut/3,49); //La:A4 鄉。  AddPitchSymbol2Q(ut/3,47); //SOL:G4  AddPitchSymbol2Q(ut/3+2\*ut,44); //ME:E4  **writeQ(h,q,outfile);**  : |

(6b)按F11編譯並執行wav-2.cpp結果如下：

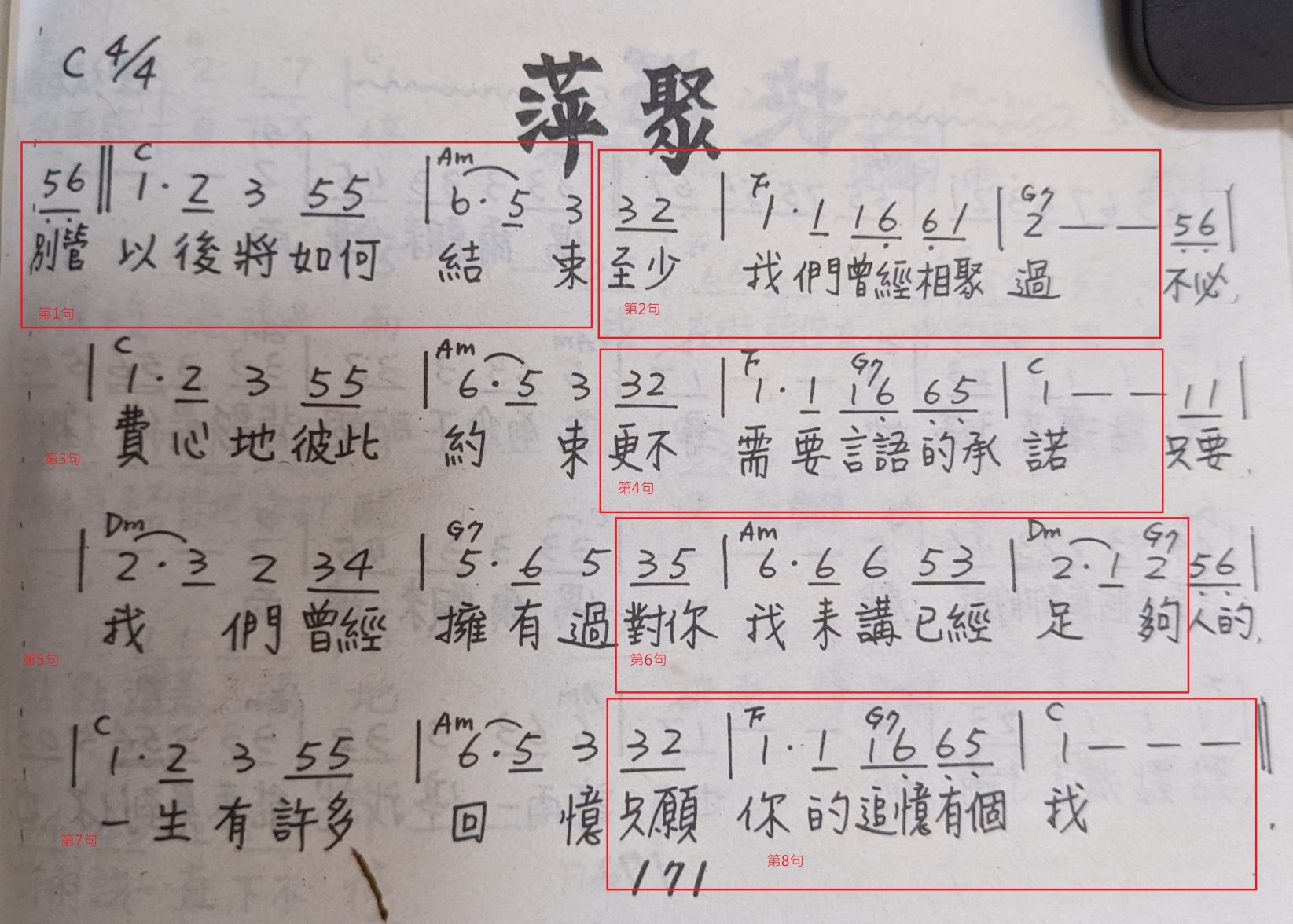
|  |
| --- |
|  |

(6c)用AUDACITY打開(6b)所製作的2.wav並用上方放大鏡鈕放大音檔波形

並查看0.275秒後的波形如下：



(7A) 將(6A)的 d:\WAV-2.cpp另存為d:\WAV-3.cpp並修改程式來產生如下簡譜8句中任一句歌詞音樂的WAV檔(檔名請存為3.wav)。



註：如上簡譜開頭二句解譯如下：

第1句：35(1/2)、37(1/2)、40(3/2)、42(1/2)、44(1)、47(1/2) 、47(1/2)、49(3/2)、47(1/2)、44(1)   
第2句：44(1/2)、42(1/2)、40(3/2)、40(1)、40(1/2)、37(1/2) 、37(1/2)、40(1/2)、42(3)

例如第1句的前半句：

**「ut/2 35 Sol別」、「 ut/2 37 La 管」、「 ut+ ut/2 40 Do以」、「 ut/2 42 RE後」**

**、「 ut 44 Mi 將」**的音訊取樣程式碼應該如下：

AddPitchSymbol2Q(ut/2,35);

AddPitchSymbol2Q(ut/2,37);

AddPitchSymbol2Q(3\*ut/2,40);

AddPitchSymbol2Q(ut/2;,42);

AddPitchSymbol2Q(ut,44);

(7B)用AUDACITY打開(7A)所製作的3.wav並用上方放大鏡鈕放大音檔波形

並查看「你的座號/10」秒後的波形並截圖3個週期波型如下(如下為27號的波型截圖)：

|  |
| --- |
|  |

(8)本操作過程檔請匯出為PDF 檔(檔名為 20220510makeWAV.pdf) 並連同3.wav音樂檔及wav-3.cpp計3個檔案一起寄給老師 [bm2director@gmail.com](mailto:bm2director@gmail.com)