

## CS3514 Laboratory Session:

Name: **Nathan Crowley**

Student ID: **118429092**

Repeat the following sections for each question

### Question Number: **Tone Generation**

#### Answer

\*\*\*\*\*

Link to YouTube video showing program running on arduino:

<https://youtu.be/LG8EC5etPKw>

\*\*\*\*\*

I started by defining the pin used for the buzzer. Followed by defining the frequencies for each note. (D = 294 Hz). I then have two arrays, notes and PossibleNotes. I use notes to hold the integer frequencies of each note. And PossibleNotes is used to check the serial monitor input against the possible notes that can be played.

Then in the setup() I set the buzzer as an output and set the serial to begin with a bit rate of 9600 bits/second.

In the loop() I begin by checking if there is input in the serial monitor. If there is still some input I read the input into a variable called inputNote, print it to the serial monitor and then use conditional statements to check if the note is in the array of PossibleNotes and to play the note if correct. Also delay when a space is inputted as to simulate a period of silence.

I also have two functions isNote() and play().

- isNote() = takes in a char input and checks this against the elements in Possiblenotes.

- play() = takes in an integer and run a while loop that plays the note for a given period.

## Code:

```
/*
* Nathan Crowley - 118429092
* Code to play an inputted note (from Serial Monitor) on a buzzer connected to an Arduino UNO.
* **Note: since I defined F as 22 there is an error mentioning that I am redefining F.
*     I am aware of this error and I have left it as it looks better having all notes in Capital letters.
*     And the code still works with this warning.
****
*TO RUN:
*  open Serial monitor.
*  enter notes (Capital letters only).
*  press send.
*Example:
* - Jingle Bells.
    EEE EEE EGCDE  FFF FFEE EE  EDDDED G  EEE EEE EGCDE GGFDCC
* - Happy Birthday song.
    CCDCFE  CCDCGF  CC CC A FED A A AFGF
* - Twinkle twinkle.
    CCGGAAG FFEEDDC GGFFEED GGFFEEDCCGGAAG FFEE DD C
*/

//set pin 9 as pin for buzzer.
#define buzzer 9

//frequencies of notes.
#define C 263
#define D 294
#define E 330
#define F 350          //be careful of error with redefining F.
#define G 392
#define A 440
#define B 467
#define highC 1046
```

```

//lists to check if input notes are correct.

int notes[] = {C,D,E,F,G,A,B};           //list of frequencies used in the play function.

char PossibleNotes[] = {'C','D','E','F','G','A','B'}; //list of notes used to compare with Serial input as to play the
//correct note.


char inputNote;                          // what type the users input from the serial monitor is.

/*Important as we compare this with char in possiblenotes list to determine what note to play.


int spaceDelay = 25;                     // how long space input will delay for.
int notedelay = 100;                     //how long each note is played for.

void setup() {

    pinMode(buzzer,OUTPUT);              //set buzzer as output.

    Serial.begin(9600);                   //begin the serial input with rate of 9600 bits per second.
}

void loop() {

    if(Serial.available()>0){             //is there input available.

        inputNote = Serial.read();        //save the input as inputNote.

        Serial.print(inputNote);

        //conditional statement to check what to do with input

        if(isNote(inputNote)){            //if input in set of notes then play it.

            //play the inputed note

            for(int i=0; i<sizeof(PossibleNotes); i++){ //loop through list of possible notes

                if(PossibleNotes[i] == inputNote){      //if note is in this list

                    play(notes[i]);                     //call fucntion to play the corresponding note

                }

            }

            /**why I use both PossibleNote and Notes:

            *   use PossibleNotes to compare the input character with list of characters

            *   use notes list to select the corresponding frequency.

            */

        }

    }
}

```

```

else if(isSpace(inputNote)){ //or if its a space have period of silence.

    delay(spaceDelay);

}

//else ignore.

}

}

//fucntion to check if the input is in the set of notes

boolean isNote(char input){ //return true or false for the fucntion in conditon if statement.

for(int i=0; i<sizeof(PossibleNotes); i++){

    if(PossibleNotes[i] == input){

        return true;

    }

}

return false;

}

//fucntion to calculate period based on frequency

int play(float frequency){ //NB** must be floats to allow the calculation!!!!

    int var = 0;

    while(var < notedelay){ //while loop to play the note for 200 miliseconds.

        float period;

        period = ((1/frequency)*1000000); //multiply 0.0038 seconds by million to convert to microseconds.

        digitalWrite(buzzer,HIGH); //digital output on

        delayMicroseconds(period/2); //delay is half the period

        digitalWrite(buzzer,LOW); //digital output off

        delayMicroseconds(period/2); //delay is half the period

        var++;

    }

}

```

Photos:

