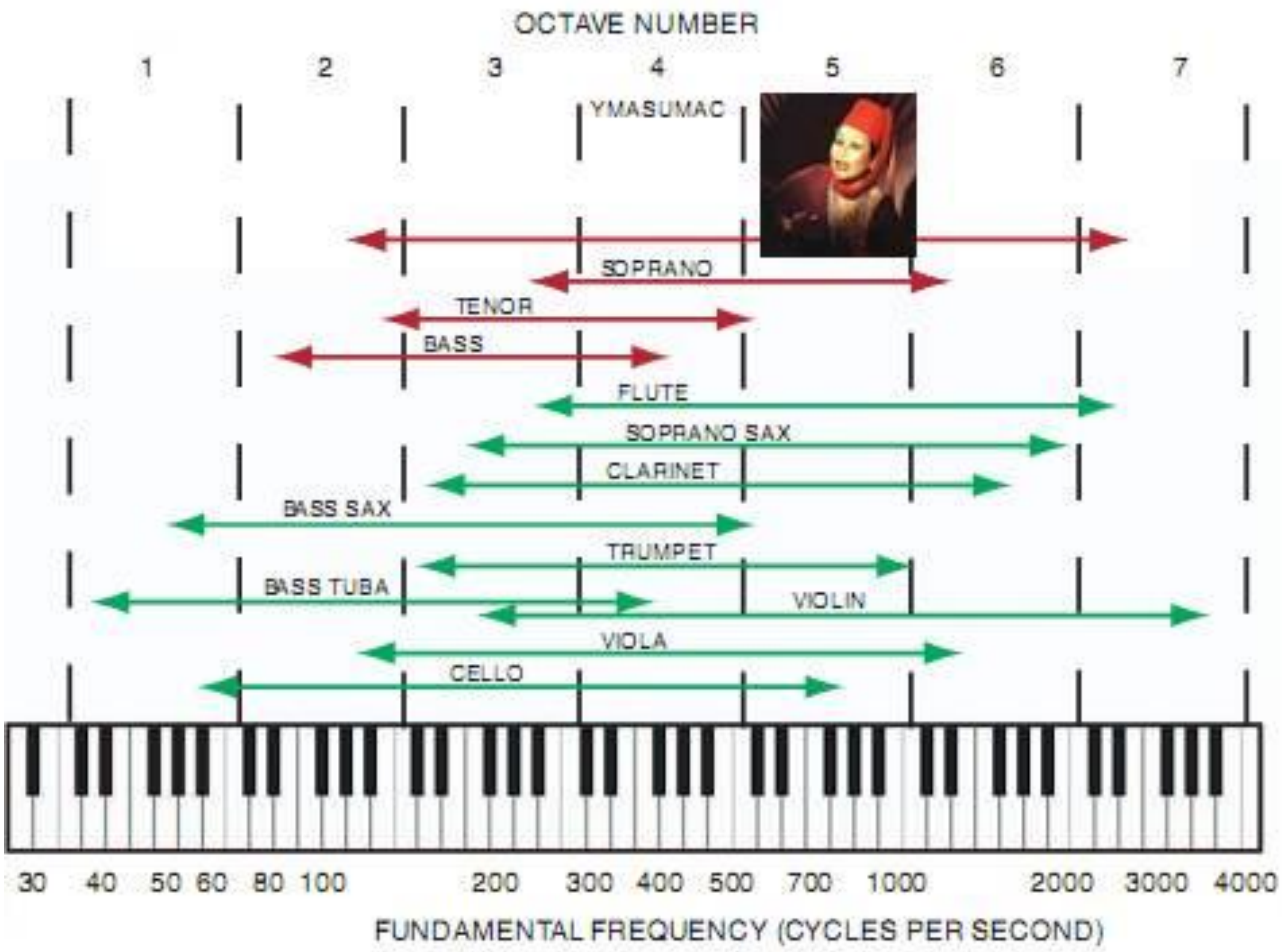
DSP Lab. Week 4
My Audio

Kyuheon Kim
Media Lab. Rm567
leeih10@khu.ac.kr
Last update : September 2, 2019

Frequency

1. A4(라)		= 440 Hz <-
2. A4#	= 440*2^(1/12)	= 466.1 Hz
3. B4(시)	= 440*2^(2/12)	= 493.8 Hz
4. C5(도)	= 440*2^(3/12)	= 523.25 Hz
5. C5#	= 440*2^(4/12)	= 554.36 Hz
6. D5(레)	= 440*2^(5/12)	= 587.33 Hz
7. D5#	= 440*2^(6/12)	= 622.25 Hz
8. E5(미)	= 440*2^(7/12)	= 659.26 Hz
9. F5(파)	= 440*2^(8/12)	= 698.46 Hz
10. F5#	= 440*2^(9/12)	= 739.99 Hz
11. G5(솔)	= 440*2^(10/12)	= 784.00 Hz
12. G5#	= 440*2^(11/12)	= 830.60 Hz
A5((라)	= 440*2	= 880 Hz



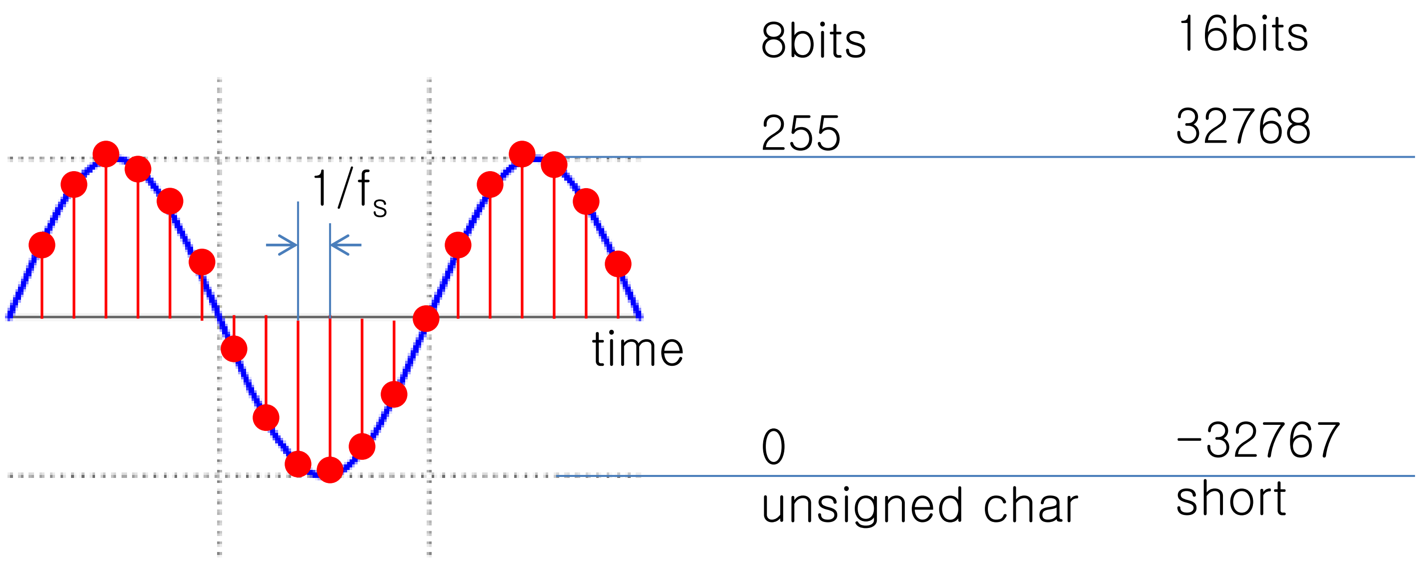
ADC (Analog-to-Digital Conversion)

ADC = sampling + quantization

Sampling = continuous to discrete, $2f_m < f_s$

Quantization = analog to digital, 2^B levels

Range



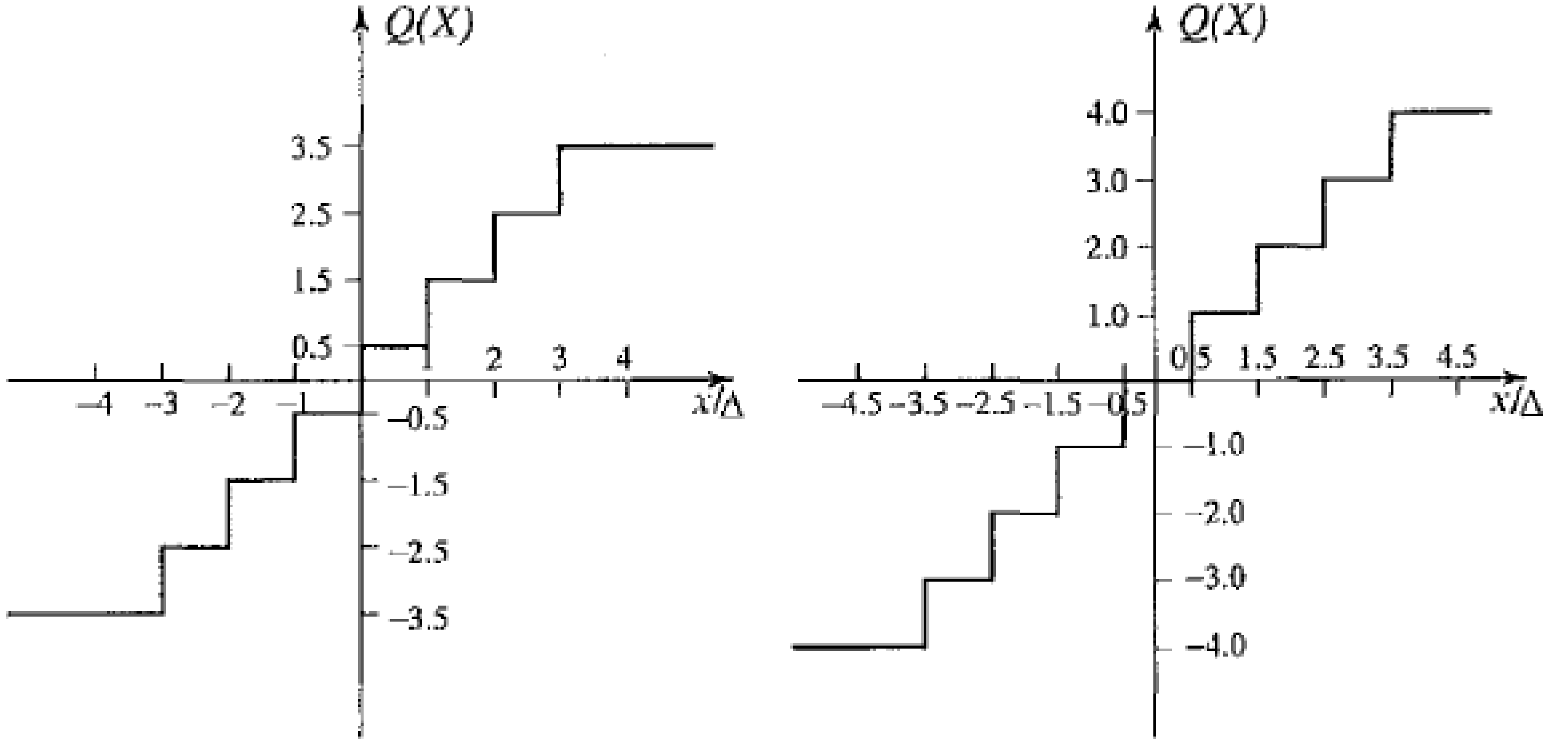
- To efficiently represent the source output, we have to reduce the number of distinct values to a much smaller set.
  - Uniform scalar quantization
  - Non-uniform scalar quantization
  - Vector quantizatoin

❖ Uniform scalar quantization

- A uniform scalar quantizer partitions the domain of input values into equally spaced intervals, except possibly at the two outer intervals.
- The endpoints of partition intervals are called the quantizer's decision boundaries.
- The output or reconstruction value corresponding to each interval is taken to be the midpoint of the interval.
- The length of each interval is referred to as the step size, denoted by the symbol Δ.
- Uniform scalar quantizers are of two types: midrise and midtread
- In case of Δ=1,

$$Q_{midrise}(x) = \lceil x \rceil - 0.5 \tag{8.4}$$

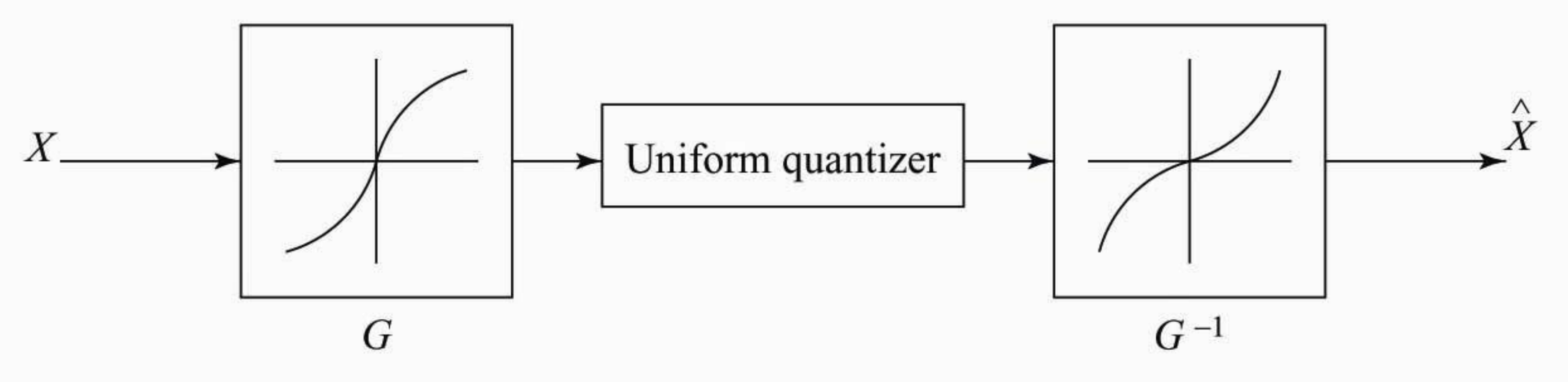
$$Q_{midtread}(x) = \lfloor x + 0.5 \rfloor \tag{8.5}$$


Q(X)
3.5
2.5
1.5
0.5
2
3
4
–4
–3
–2
–1
–0.5
–1.5
–2.5
–3.5
x/Δ
Q(X)
4.0
3.0
2.0
1.0
0.5
1.5
2.5
3.5
4.5
–4.5
–3.5
–2.5
–1.5
–0.5
–1.0
–2.0
–3.0
–4.0
x/Δ

(a) (b)

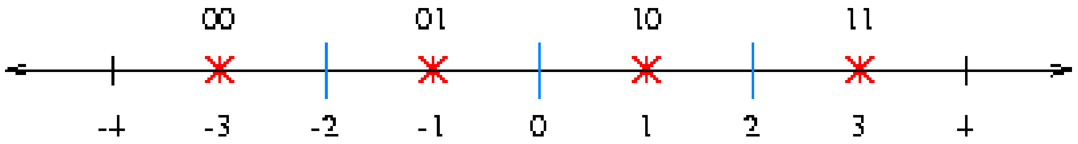
❖ Companded quantizer

- the input is mapped by a compressor function G and then quantized using a uniform quantizer.
- After transmission, the quantized values are mapped back using an expanded function G^{-1}.

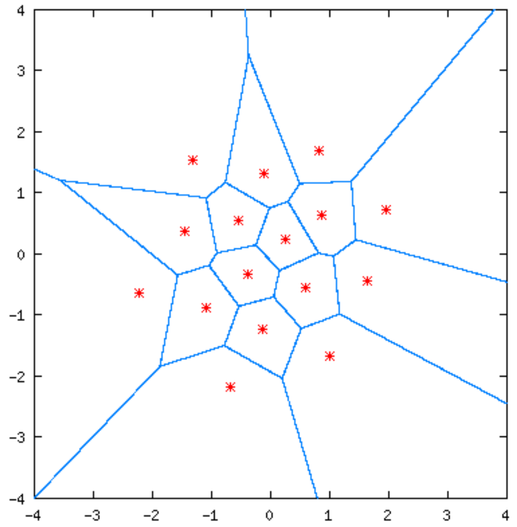


❖ Vector quantization

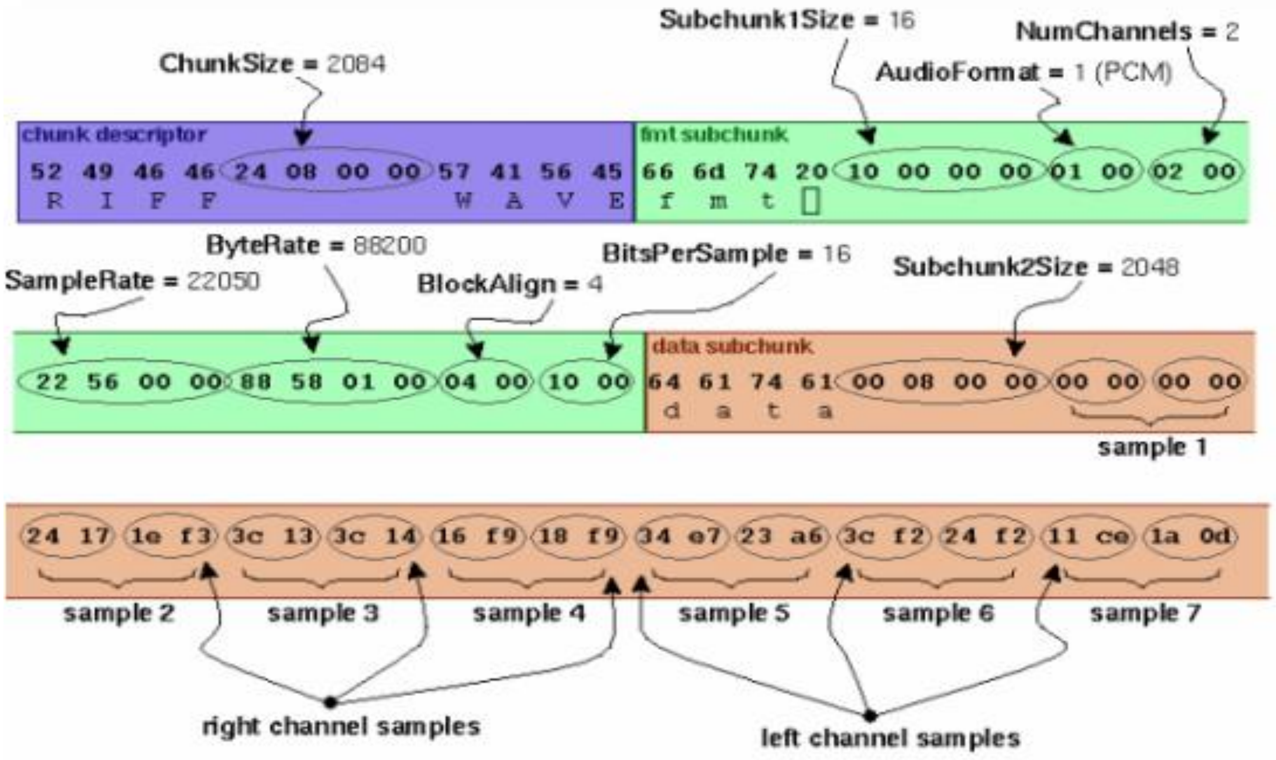
- Any compression system performs better if it operates on vectors or groups of samples rather than on individual symbols or samples.



- n-component code vector represents vectors that lie within a region in n-dimensional space.
- A collection of these code vectors forms the codebook for the vector quantizer.



Wave File Header (44 bytes long)




```
#define WORD unsigned short
#define DWORD unsigned int

void writeHeader(char *WaveFile, int nn)
{
        // RIFF chunk
        strcpy(WaveFile,"RIFF");
        *(DWORD *)(WaveFile+4) = nn-8;
        strcpy(WaveFile+8,"WAVE");
        // fmt chunk
        strcpy(WaveFile+12,"fmt"); WaveFile[15] = ' ';
        *(DWORD *)(WaveFile+16) = 16;    // cksize
        *(WORD *)(WaveFile+20) = 1;      // wFormatTag PCM 1
        *(WORD *)(WaveFile+22) = 1;      // nChannels                 mono 1  stereo 2
        *(DWORD *)(WaveFile+24) = 8000;  // nSamplesPerSec      8000Hz or 44100Hz
        *(DWORD *)(WaveFile+28) = 8000;  // nAvgBytesPerSec
        *(WORD *)(WaveFile+32) = 1;      // nBlockAlign = nChannels*wBitsPerSample/8
        *(WORD *)(WaveFile+34) = 8;      // wBitsPerSample           8bits or 16bits
        // data chunk
        strcpy(WaveFile+36,"data");
        *(DWORD *)(WaveFile+40) = nn;    // cksize
        return;
}
```

Read header of a wave file

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define WORD unsigned short
#define DWORD unsigned int

void main(){
// 1. Read the image file.
    char f[44];
    FILE *fff;
    if((fff = fopen("Beatles-LetItBe-wav.wav","rb")) == NULL){ printf("cant open Beatles-LetItBe-wav.wav.\n");  exit(123);}
// 2. Read the first 44 bytes
    if((fread(f,1,44,fff) != 44)){ printf("cant read Beatles-LetItBe-wav.wav..\n"); exit(444);      }
    fclose(fff);
// 3. Read all fields.
    printf("RIFF     %c%c%c%c\n",f[0],f[1],f[2],f[3]);
    printf("filesize %d\n",*(DWORD*)(f+4));  // sub-chunk 2 size
    printf("WAVE     %c%c%c%c\n",f[8],f[9],f[10],f[11]);
    printf("cksize   %d\n",*(DWORD*)(f+16));
    printf("channels %d\n",*(WORD*)(f+22));
    printf("fs       %d\n",*(DWORD*)(f+24));
    printf("bytes/s  %d\n",*(DWORD*)(f+28));
    printf("bits/sam %d\n",*(WORD*)(f+34));
    printf("data     %c%c%c%c\n",f[36],f[37],f[38],f[39]);
    printf("cksize   %d\n",*(DWORD*)(f+40)); // sub-chunk 2 size
    delete f;
    getchar();
}
```

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define WORD unsigned short
#define DWORD unsigned int
#define NN 4096
#define N  1024

void main()
{
// 1.   Read the image file.
    char *f;
    f = new char[48];
    FILE *fff,*fout,*foutr;
    if((fff = fopen("Beatles-LetItBe-wav.wav","rb")) == NULL){
        printf("cant open Beatles-LetItBe-wav.wav.₩n");
        exit(123);
    }
// 2.    Read the first 48 bytes => write header
    if((fread(f,1,48,fff) != 48)){
        printf("cant read Beatles-LetItBe-wav.wav..₩n");
        exit(444);
    }
    if((fout = fopen("Beatles-LetItBe-xxx.wav","wb")) == NULL){
        printf("cant open Beatles-LetItBe-xxx.wav.₩n");
        exit(123);
    }
// 3.   Read all fields.
    printf("RIFF     %c%c%c%c₩n",f[0],f[1],f[2],f[3]);
    printf("filesize %d₩n",*(DWORD*)(f+4));  // sub-chunk 2 size
    printf("WAVE     %c%c%c%c₩n",f[8],f[9],f[10],f[11]);
    printf("cksize   %d₩n",*(DWORD*)(f+16));
    printf("channels %d₩n",*(WORD*)(f+22));
    printf("fs       %d₩n",*(DWORD*)(f+24));
    printf("bytes/s  %d₩n",*(DWORD*)(f+28));
    printf("bits/sam %d₩n",*(WORD*)(f+34));
    printf("data     %c%c%c%c₩n",f[36],f[37],f[38],f[39]);
    printf("cksize   %d₩n",*(DWORD*)(f+40)); // sub-chunk 2 size
    DWORD FileSize = *(DWORD*)(f+40);

    if((fwrite(f,1,48,fout) != 48)){
        printf("cant write header on Beatles-LetItBe-xxx.wav..₩n");
        exit(444);
    }

    FileSize /= 4;     // 4bytes per sample
// 4.   modify audio signal to mono
    short leftdata[N],rightdata[N],alldata[N*2];
    for(int n = 0; n < FileSize; n += N){ // all data
        if((fread(alldata,4,N,fff) != N)){
            printf("cant read %d-th data from Beatles-LetItBe-wav.wav..₩n",n);
            exit(244);
        }
        // store left/right data
        int ii,iii;
        for(ii=iii=0;ii<N;ii++,iii+=2){
            leftdata[ii]  = alldata[iii];
            rightdata[ii] = alldata[iii+1];
        }
        // modify left/right data
        // ......
        // 이 부분이 여러분들이 프로그램할 자리.... Good luck!!
        // ......
        // restore left/right data
        for(ii=iii=0;ii<N;ii++,iii+=2){
            alldata[iii] = leftdata[ii];
            alldata[iii+1] = rightdata[ii];
        }
// 5.   write on outut file
        if((fwrite(alldata,4,N,fout) != N)){
            printf("cant write %d-th data on Beatles-LetItBe-xxx.wav..₩n",n);
            exit(246);
        }
    } // all data
    fclose(fff);
    fclose(fout);
    delete f;
}
```

MP3 to WAV Converter

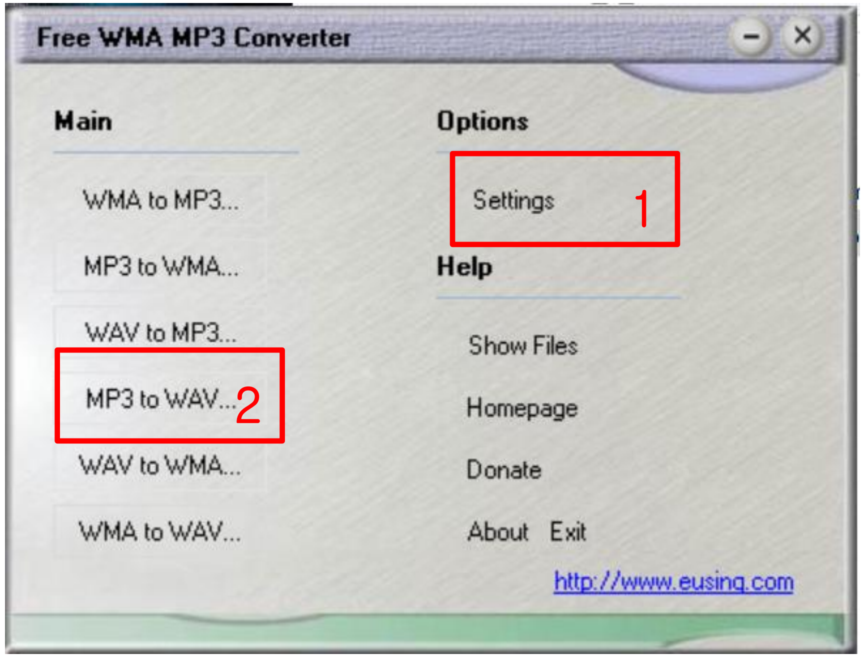
MP3 to wav converter "FWMCSetup.zip"

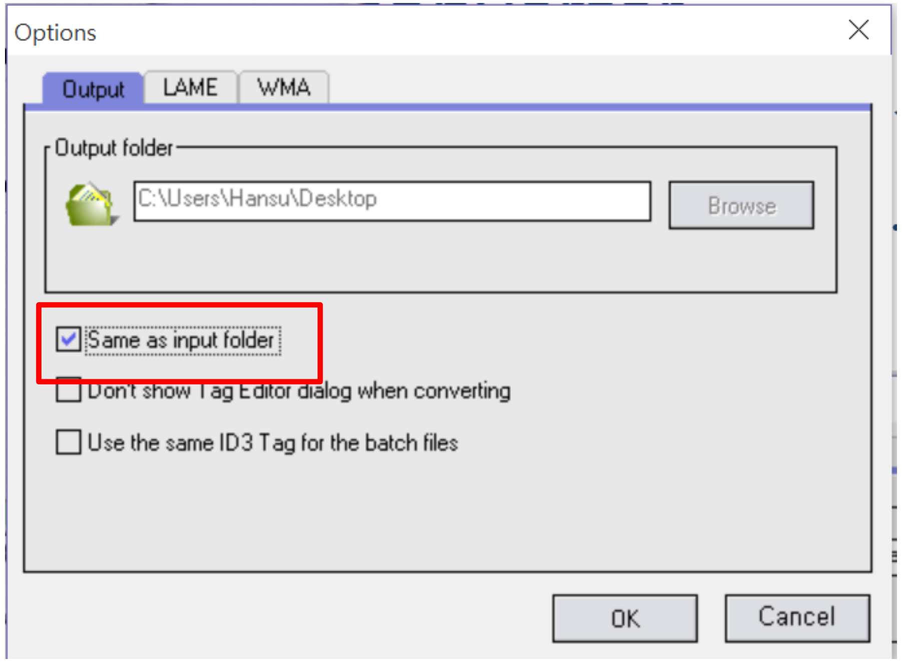
Free WMA MP3 Converter를 설치 후 실행한다.

그림1 과 같은 화면이 나오면 **1. Settings**을 누른다.

그림2 와 같은 화면이 나오면 **Same as input folder**를 체크!

그림1 에서 **2. MP3 to WAV** 클릭 후 convert할 음원 선택!





Week 4 assignment

Let It Be 음악에, 중간에 산토끼를 넣어라.

(ex) 첫 '솔'은 0.5초, f_s=44100이면, f_{max}=22050

```
dt = 1/44100.0;
 for(t=0;  t<0.5;  t+=dt){
    *(leftdata++) = *(rightdata++) = (short)(20000.0*sin(2*PI*784*t));
 }
```

1.	A4(라)		= 440	Hz <-
2.	A4#	= 440*2^(1/12)	= 466.1 Hz	
3.	B4(시)	= 440*2^(2/12)	= 493.8 Hz	
4.	C5(도)	= 440*2^(3/12)	= 523.25 Hz	
5.	C5#	= 440*2^(4/12)	= 554.36 Hz	
6.	D5(레)	= 440*2^(5/12)	= 587.33 Hz	
7.	D5#	= 440*2^(6/12)	= 622.25 Hz	
8.	E5(미)	= 440*2^(7/12)	= 659.26 Hz	
9.	F5(파)	= 440*2^(8/12)	= 698.46 Hz	
10.	F5#	= 440*2^(9/12)	= 739.99 Hz	
11.	G5(솔)	= 440*2^(10/12)	= 784.00 Hz	
12.	G5#	= 440*2^(11/12)	= 830.60 Hz	
	A5((라)	= 440*2	= 880 Hz	

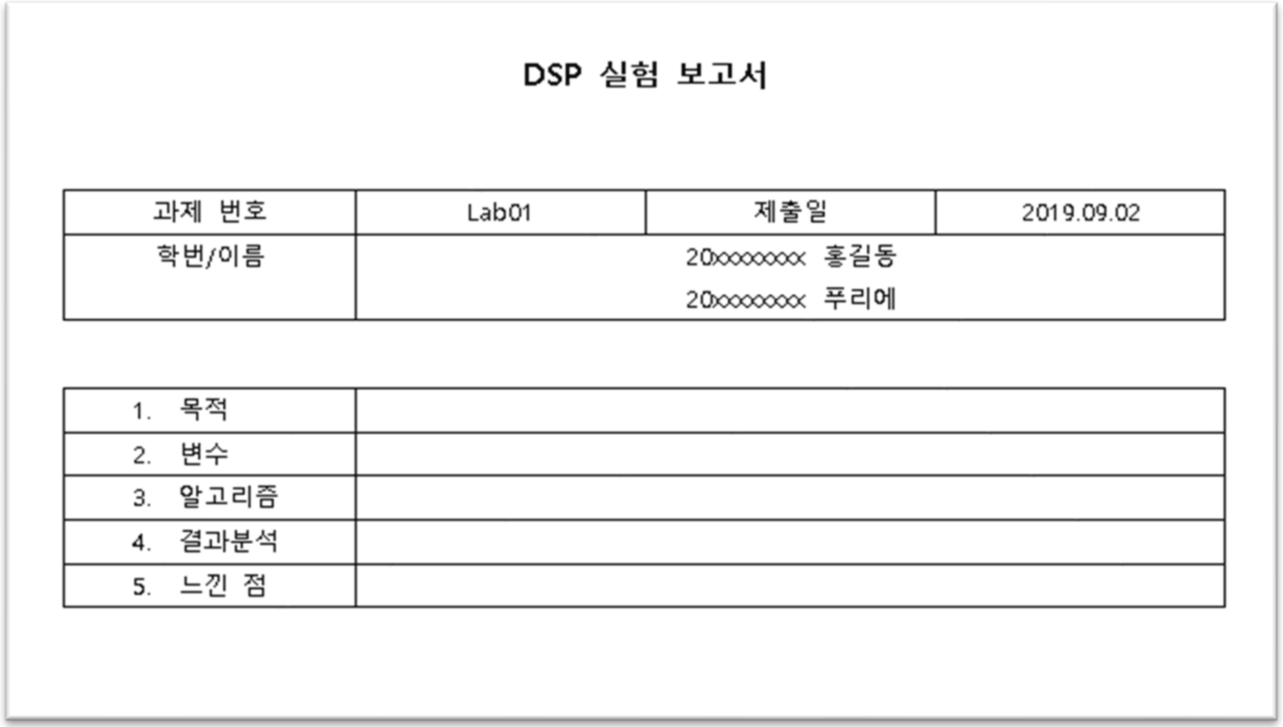
Week 4 assignment

"KLAS에 제출할 때 다음 사항을 꼭 지켜주세요"

1. 파일명 : "Lab00_요일_대표자이름.zip"

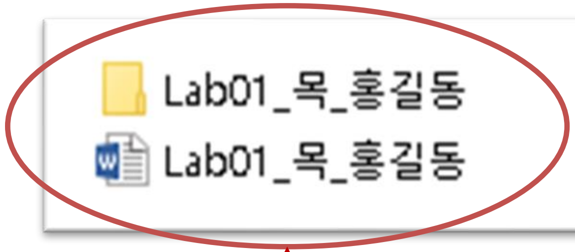
Ex) Lab01_목_홍길동.zip **(압축 툴은 자유롭게 사용)**

2. 제출 파일 **(보고서와 프로그램을 압축해서 제출)**
   - 보고서 파일 (hwp, word): 이름, 학번, 목적, 변수, 알고리즘(순서), 결과 분석, 느낀 점
   - 프로그램


DSP 실험 보고서

과제 번호	Lab01	제출일	2019.09.02
학번/이름	20xxxxxxxx 홍길동 20xxxxxxxx 푸리에		

1. 목적	
2. 변수	
3. 알고리즘	
4. 결과분석	
5. 느낀 점	



C++ 프로그램 파일
보고서

압축하여 제출