


# 실습 - FIR filtering in C

Seoul National University of Science and Technology

# 실습 소개

- 간단한 FIR 필터링을 C언어로 구현한다.

필터 차수(order):  $N$   
필터 길이(length):  $N + 1$

$$y(n) = \sum_{k=0}^N b(k) x(n-k)$$


필터 계수(filter coefficients):  $\{b(0), b(1), \dots, b(N)\}$



$N + 1$  개

$$y(n) = b(0) x(n) + b(1) x(n-1) + \dots + b(N) x(n-N)$$

# 입력 버퍼의 이해

$$y(n) = b(0) x(n) + b(1) x(n-1) + \dots + b(N) x(n-N)$$

예)  $N = 7$ 인 경우

**input signal**

2, -1, 1, 4, 5, 8, 13, 12, 7, 1, -3, ...

**x\_buffer array**

initial	0	0	0	0	0	0	0	0
1st sample	0	0	0	0	0	0	0	2
2nd sample	0	0	0	0	0	0	2	-1
3rd sample	0	0	0	0	0	2	-1	1
8th sample	2	-1	1	4	5	8	13	18
9th sample	-1	1	4	5	8	13	18	12

$x(n-7)$   $x(n-6)$   $x(n-5)$   $x(n-4)$   $x(n-3)$   $x(n-2)$   $x(n-1)$   $x(n)$

# C 프로그램

$$y(n) = b(0) x(n) + b(1) x(n-1) + \dots + b(N) x(n-N)$$

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define MAX_FILTER_LEN 257
```

```
short x_buffer[MAX_FILTER_LEN];
```

```
float coeffs[MAX_FILTER_LEN];
```

```
void main(int argc, char *argv[]) {
```

```
    FILE *fi, *fo;
```

```
    short filter_len; // filter length
```

```
    short k;
```

```
    short x; // input sample
```

```
    short y; // output sample
```

```
    float ftemp;
```

```
    if (argc < 4) {
```

```
        printf("Usage : %s input output filter_length\n", argv[0]);
```

```
        exit(-1);
```

```
    }
```

총 4개 => argc = 4

C:\W???> .\Debug\myfir???? input.snd output.snd 8

argv[0]

argv[1]

argv[2]

argv[3]

# C 프로그램

```
//if ((fi = fopen(argv[1], "rb")) == NULL) {  
if (fopen_s(&fi, argv[1], "rb") != 0) {  
    printf("Input file [%s] is not found.\n", argv[1]);  
    exit(-1);  
}  
//fo = fopen(argv[2], "wb");  
fopen_s(&fo, argv[2], "wb");
```

filter\_len = atoi(argv[3]); ← ascii to integer: 문자열을 정수형으로 바꾸어주는 built-in 함수

```
// Simple filter coefficients (moving average filter)  
for (k = 0; k < filter_len; k++) {  
    coeffs[k] =  
};
```

← 필터계수 모두 1/N로 같음

# C 프로그램

```
while (1) {
```

```
    if (fread(&x, sizeof(short), 1, fi) == NULL) break;
```

```
    x_buffer[          ] = x;
```

```
    ftemp = 0.0;
```

```
    for (k = 0; k< filter_len; k++) {
```

```
        ftemp +=
```

```
    }
```

```
    if (ftemp < -32768.0)
```

```
        ftemp = -32768.0;
```

```
    else if (ftemp > 32767.0)
```

```
        ftemp = 32767.0;
```

```
    y = (short)ftemp;
```

```
    fwrite(&y, sizeof(short), 1, fo);
```

```
    printf("%d\n", y);
```

```
    // Update filter history
```

```
    for (k = 0; k<          ; k++) {
```

```
        x_buffer[k] = x_buffer[k + 1];
```

```
    }
```

```
} // end of while()
```

```
    fclose(fi); fclose(fo);
```

```
} // end of main()
```

$$y(n) = b(0) x(n) + b(1) x(n-1) + \dots + b(N) x(n-N)$$

**input signal**

2, -1, 1, 4, 5, 8, 13, 18, 12, 7, 1, -3, ...

**x\_buffer array**

initial	0	0	0	0	0	0	0	0
1st sample	0	0	0	0	0	0	0	2
2nd sample	0	0	0	0	0	0	2	-1
3rd sample	0	0	0	0	0	2	-1	1
8th sample	2	-1	1	4	5	8	13	18
9th sample	-1	1	4	5	8	13	18	12

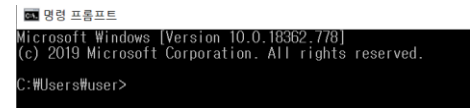
x(n-7) x(n-6) x(n-5) x(n-4) x(n-3) x(n-2) x(n-1) x(n)

# C 프로그램 실행

※ 프로젝트 폴더 위치에 input.snd 파일을 복사해놓고, batch 파일(rrr.bat)을 만들어 실행한다. 이때 현재 폴더 아래 Debug 폴더에 실행파일(myfir?????.exe)이 제대로 있는지 확인한다.

방법1) 명령프롬프트(DOS command window)에서 다음과 같이 실행

C:\W???>.WDebugWmyfir???? input.snd output.snd 8



방법2) rrr.bat 파일 더블클릭

**<rrr.bat 내용>**

.WDebugWmyfir???? input.snd output.snd 8

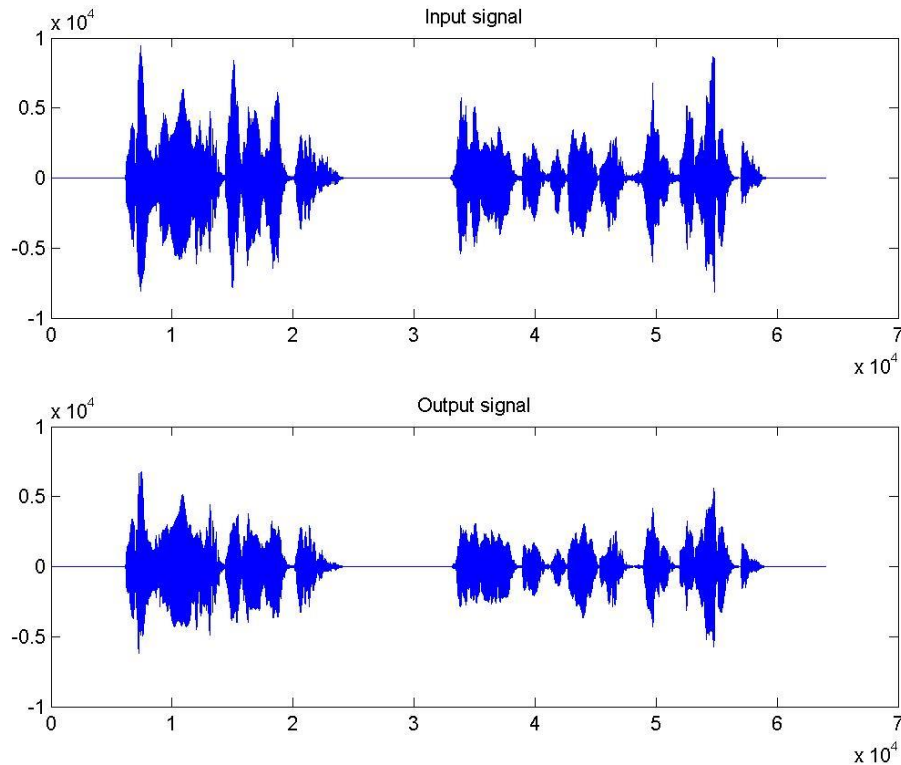
이때, 위의 문자열들은 argv[0] argv[1] argv[2] argv[3]에 해당한다.

# MATLAB을 이용한 결과 검증

```
Fs = 8000;
fi = fopen('input.snd','rb');
x = fread(fi, inf, 'int16');
fclose(fi);
fi = fopen('output.snd','rb');
y = fread(fi, inf, 'int16');
fclose(fi);
figure(1)
subplot(2,1,1); plot(x);
title('Input signal');
subplot(2,1,2); plot(y);
title('Output signal');
print -djpeg 'fig1.jpg'
figure(2)
subplot(2,1,1); specgram(x);
title('Input signal');
subplot(2,1,2); specgram(y);
title('Output signal');
print -djpeg 'fig2.jpg'
soundsc(x,Fs);
pause
soundsc(y,Fs);
```

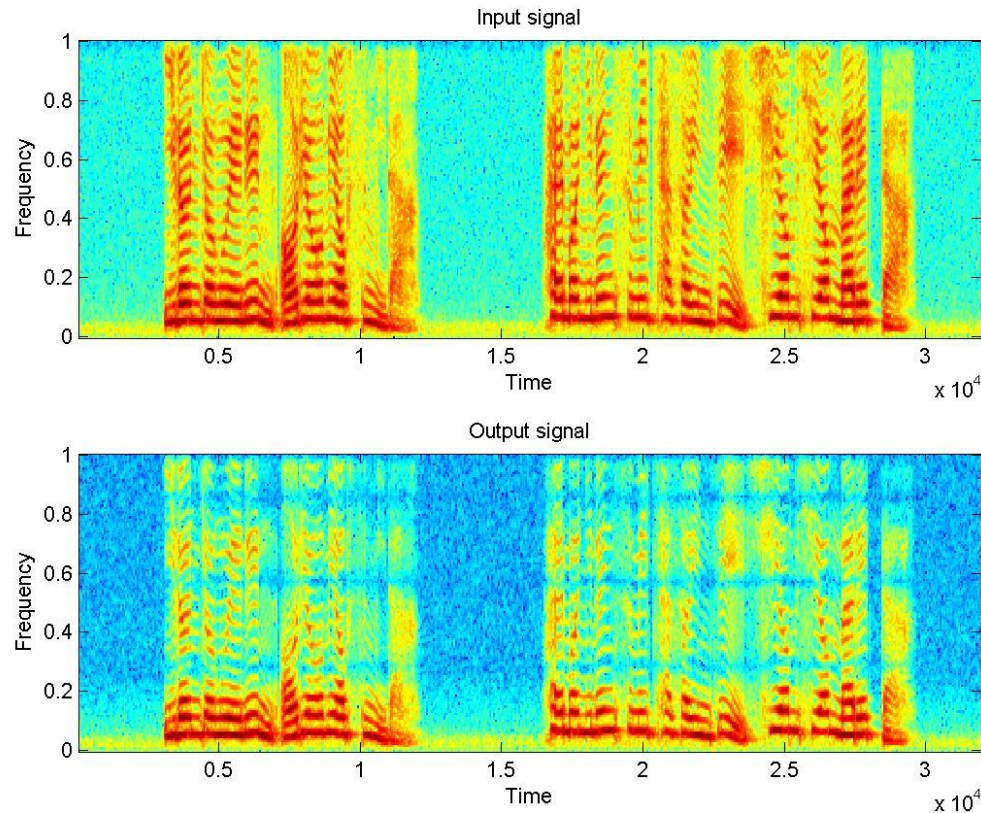


# 실행 결과 - 파형



파형으로는 어떤 필터링을 했는 지 분석이 어렵다

# 실행 결과 - 스펙트로그램



저역통과 필터링(low-pass filtering)을 수행하였음을 알 수 있다