

프로그래밍 언어 (실습)

실습 4. 키보드와 Beep() 함수를 사용한 5옥타브 전자 피아노 (보충설명)



교수 김 영 탁

영남대학교 정보통신공학과

(Tel : +82-53-810-2497; Fax : +82-53-810-4742

<http://antl.yu.ac.kr/>; E-mail : ytkim@yu.ac.kr)

Outline

◆ 전자피아노

- Console Input/Output
- ASCII code
- function Beep()
- keyboard input state checking
- if-else
- switch-case
- mapping table

◆ Oral Test



표준입력장치 (Keyboard) 입력

Console Input / Output

◆ Console input/output

- `printf()`; // print out formatted string
- `scanf()`: // input formatted data (multiple item)

문자 입출력 함수	의미
<code>int getchar(void)</code>	버퍼를 사용하며, 표준 입력장치 (키보드)로부터 하나의 문자를 읽어서 int 데이터 형으로 반환.
<code>void putchar(int ch)</code>	int 데이터형 인수 ch에 담긴 문자를 출력
<code>int _getch(void)</code>	버퍼를 사용하지 않고, 입력된 하나의 문자를 읽어 반환. 문자가 입력될 때, 화면에 표시(에코)를 하여 주지 않음.
<code>int _getche(void)</code>	버퍼를 사용하지 않고, 입력된 하나의 문자를 읽어 반환. 문자가 입력될 때, 화면에 표시(에코)를 하여줌.
<code>void _putch(int ch)</code>	버퍼를 사용하지 않고, int 데이터형 인수 ch에 담긴 문자를 출력
<code>scanf("%c", &ch)</code>	표준 입력장치 (키보드)로부터 문자를 읽어 ch에 저장
<code>printf("%c", ch)</code>	ch에 담긴 문자를 출력



Keyboard

◆ Standard Keyboard



ASCII (Character Set)

◆ ASCII (American Standard Code for Information) Interchange

	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0x10	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0x20	SP	!	“	#	\$	%	&	‘	()	*	+	,	-	.	/
0x30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0x40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0x50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0x60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0x70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL



키보드 입력 상태 확인

◆ _kbhit()과 GetAsyncKeyState(virtual key)

키보드 입력 상태 관련 함수	함수 설명
GetAsyncKeyState()	지정된 가상키의 상태를 반환 VK_CONTROL : Ctrl 키 VK_MENU: Alt 키
_kbhit()	키보드에서 어떤 키가 눌러지면 1을 반환



test_KeyboardInputState()

```
1 : void test_KeyboardInputState()
2 : {
3 :     int key;
4 :
5 :     printf("Check keyboard input state...\n");
6 :     printf("Enter a key (ESC to stop) :\n");
7 :     while (1)
8 :     {
9 :         key = _getch();
10 :        if (key == ESC)
11 :            return;
12 :
13 :        printf("input key: ");
14 :        if (GetAsyncKeyState(VK_CONTROL) & 0x8000)
15 :            printf("CTRL+");
16 :        if (GetAsyncKeyState(VK_MENU) & 0x8000)
17 :            printf("ALT+");
18 :        printf("%c( %#2X)\n", key, key);
19 :    }
20 : }
```

```
Check keyboard input state...
Enter a key (ESC to stop) :
input key: K(0X4B)
input key: B(0X42)
input key: l(0X6C)
input key: k(0X6B)
input key: l(0X6C)
input key: ALT+k(0X6B)
input key: CTRL+
input key: CTRL(0X8)
input key: ALT+G(0X47)
input key: ALT+A(0X41)
input key: ALT+S(0X53)
input key: ALT+D(0X44)
input key: ALT+F(0X46)
input key: ALT+G(0X47)
```



Beep() 함수를 사용한 피아노 음 생성

Beep()

◆ Beep() function

- static void **Beep**(int frequency, int duration)
- Plays the sound of a beep of a specified frequency and the specified duration through the console speaker



Frequency of Music Code

◆ Frequency of Music Codes

	1 Octave	2 Octave	3 Octave	4 Octave	5 Octave	6 Octave	7 Octave	8 Octave
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133



Simple Electric Piano

```
/* Simple Electric Piano with Keyboard (1) */
#include <conio.h> // for getch()
#include <iostream>
#include <Windows.h>

int key_to_freq(char key);
int main()
{
    char ch, alt_on = 0;
    int duration = 100; // 100 msec
    printf("\nSimple Electric Piano !!\n");
    printf("'A'~ 'J' are mapped into C, D, E, F, G, A, B code in octave 8,\n");
    printf("'Z'~ 'M' are mapped into C, D, E, F, G, A, B code in octave 7,\n");
    printf("'q'~ 'u' are mapped into C, D, E, F, G, A, B code in octave 6,\n");
    printf("'a'~ 'j' are mapped into C, D, E, F, G, A, B code in octave 5, and\n");
    printf("'z'~ 'm' are mapped into C, D, E, F, G, A, B code in octave 4,
           respectively.\n");
    printf("input next key :\n");
```



Simple Electric Piano

```
/* Simple Electric Piano with Keyboard (2) */

while ((ch = _getch()) != ESC)
{
    alt_on = 0;
    if (GetSynchKeyState(VK_MENU) & 0x8000)
        alt_on = 1;
    freq = mappingKeyToFreq(ch, alt_on);
    if (freq != -1) {
        printf("input key (%c) : freq (%3d)\n", ch, freq);
        Beep(freq, duration);
    } else {
        printf("input key (%c) is wrong key input..\n", ch);
    }
}
}
```



Alt 키를 사용한 반음 키 확인

◆ 반음이 있는 음계

- C (도), D(레), F(파), G(솔), A(라)

◆ 키보드를 사용한 전자피아노에서 반음을 나타내는 방법

- 키와 함께 Alt 키를 누르고, 이 상태를 확인
- GetAsyncKeyState() 함수를 VK_MENU와 함께 호출하여 Alt 키가 눌러졌는지 확인

◆ 반음 키에 대한 처리

- (Alt 키가 함께 눌러져) 반음 키가 입력되었으면 해당 음계의 주파수를 반음 높은 음계의 주파수로 설정



key_to_freq() – 방법 1

◆ mapping by if-else

```
int key_to_freq (char key, char alt_on)
{
    int freq;

    if ((key == 'z') && (alt_on == 0)) {
        freq = 262; printf("Octave 4 - 도 (C ), 261.63 Hz\n",);
    } else if ((key == 'z') && (alt_on == 1)) {
        freq = 277; printf("Octave 4 - 도# (C# ), 277.18 Hz\n",);
    } else if (( key == 'x') && (alt_on == 0)) {
        freq = 294; printf("Octave 4 - 레 (D), 293.66 Hz\n");
    } else if (( key == 'x') && (alt_on == 1)) {

        . . . . .
    }
}
```



key_to_freq() – 방법 1

◆ mapping by if-else

```
    } else if (( key == 'H') && (alt_on == 0)) {  
        freq = 7040 printf("Octave 8 - 라 (A), 7040.00 Hz\n");  
    } else if (( key == 'H') && (alt_on == 1)) {  
        freq = 7459 printf("Octave 8 - 라# (A#), 7458.62 Hz\n");  
    } else if ( key == 'J') {  
        freq = 7902; printf("Octave 8 - 시 (B), 7902.13 Hz\n");  
    } else {  
        freq = -1; // wrong key  
    }  
  
    return freq;  
}
```



Example of Execution

◆ Execution of Simple Electric Piano

```
Simple Electric Piano !!
'Q' ~ 'U' are mapped into C, D, E, F, G, A, B code in octave 6,
'A' ~ 'J' are mapped into C, D, E, F, G, A, B code in octave 5,
'Z' ~ 'M' are mapped into C, D, E, F, G, A, B code in octave 4,
'q' ~ 'u' are mapped into C, D, E, F, G, A, B code in octave 3,
'a' ~ 'j' are mapped into C, D, E, F, G, A, B code in octave 2, and
'z' ~ 'm' are mapped into C, D, E, F, G, A, B code in octave 1, respectively.

piano key <ESC to end> : a : Octave 5 ? 도<C>, 523.25Hz
piano key <ESC to end> : s : Octave 5 ? 레<D>, 587.32Hz
piano key <ESC to end> : d : Octave 5 ? 미<E>, 659.25Hz
piano key <ESC to end> : f : Octave 5 ? 파<F>, 698.46Hz
piano key <ESC to end> : g : Octave 5 ? 솔<G>, 783.99Hz
piano key <ESC to end> : h : Octave 5 ? 라<A>, 880.00Hz
piano key <ESC to end> : j : Octave 5 ? 시<B>, 987.77Hz
piano key <ESC to end> : q : Octave 6 ? 도<C>, 1046.5Hz
piano key <ESC to end> : w : Octave 6 ? 레<D>, 1174.66Hz
piano key <ESC to end> : e : Octave 6 ? 미<E>, 1318.51Hz
piano key <ESC to end> : r : Octave 6 ? 파<F>, 1396.91Hz
piano key <ESC to end> : t : Octave 6 ? 솔<G>, 1567.98Hz
piano key <ESC to end> : y : Octave 6 ? 라<A>, 1760.00Hz
piano key <ESC to end> : u : Octave 6 ? 시<B>, 1975.53Hz
piano key <ESC to end> : Z : Octave 7 ? 도<C>, 2093.00Hz
piano key <ESC to end> : X : Octave 7 ? 레<D>, 2349.32Hz
piano key <ESC to end> : C : Octave 7 ? 미<E>, 2637.02Hz
piano key <ESC to end> : V : Octave 8 ? 파<F>, 2793.83Hz
piano key <ESC to end> :
```



key_to_freq() – 방법 2

◆ mapping by switch case

```
int key_to_freq (char key, char alt_on)
{
    int freq;
    switch (key)
    {
        case ('z') :
            if (alt_on == 0) {freq = 262; printf("Octave 4 - 도 (C), 261.63 Hz\n");}
            else {freq = 277; printf("Octave 4 - 도# (C#), 277.18 Hz\n");}
            break;
        case ('x') :
            if (alt_on == 0) {freq = 294; printf("Octave 4 - 레 (D), 293.66 Hz\n"); }
            else {freq = 311; printf("Octave 4 - 레# (D#), 311.13 Hz\n"); }
            break;
        case ('c') : freq = 330; printf("Octave 4 - 미 (E), 329.63 Hz\n"); break;

        . . . . .
    }
```



key_to_freq() – 방법 2

◆ mapping by switch case

```
case ('Y') :  
    if (alt_on == 0) { freq = 7040; printf("Octave 8 - 라 (A), 7040.00 Hz\n"); }  
    else { freq = 7459; printf("Octave 8 - 라# (A#), 7040.00 Hz\n"); }  
    break;  
case ('U') : freq = 7902; printf("Octave 8 - 시 (B), 7902.13Hz\n"); break;  
default:    freq = -1; // wrong key  
}  
return freq;  
}
```



key_to_freq() – 방법 3

◆ using characteristics of ASCII codes

- 'a' ~ 'z' : 0x61 ~ 0x7A
- 'A' ~ 'Z' : 0x41 ~ 0x5A

◆ Generation of shift_on flag and index

```
if ((key >= 'a') && (key <= 'z')) {  
    shift_on = 0;  
    index = key - 'a';    // index: 0 ~ 25  
} else if ((key >= 'A') && (key <= 'Z')) {  
    shift_on = 1;  
    index = key - 'A';    // index: 0 ~ 25  
} else {  
    index = -1; // wrong key  
}
```



반음 음계가 고려된 Freq Mapping Table

◆ 반음이 있는 음계

- C (도), D(레), F(파), G(솔), A(라)

◆ Frequency Mapping Table

- 입력 key ASCII code, Shift Key, Alt Key의 값을 인덱스로 사용하는 3차원 배열을 구성
- 각 원소는 해당 옥타브-음계의 주파수 값 저장
- 만약 해당 키가 전자피아노에서 사용되지 않으면 -1 저장



◆ Frequency Mapping Table

shift_on = 0 (Off)

Octave 6: q, w, e, r, t, y, u

Octave 5: a, s, d, f, g, h, j

Octave 4: z, x, c, v, b, n, m

shift_on = 1 (On)

Octave 8: A, S, D, F, G, H, J

Octave 7: Z, X, C, V, B, N, M

int Key_to_Freq[2][2][26]; //alt_on, shift_on, a_z

```
int Key_to_Freq[2][2][26] =
{
    /* alt_on == 0*/
    {
        { 523, 392, 330, 660, 1323, 698, 784, 880, -1, 988, -1, -1, 494,
          440, -1, -1, 1048, 1396, 588, 1568, 1976, 349, 1176, 294, 1760, 262 },
        { 4186, 3136, 2637, 5274, -1, 5588, 6272, 7040, -1, 7902, -1, -1, 3951,
          3520, -1, -1, -1, -1, 4699, -1, -1, 2794, -1, 2349, -1, 2093 }
    },
    /* alt_on == 1*/
    {
        { 554, 415, 330, 660, 1323, 740, 830, 932, -1, 988, -1, -1, 494,
          466, -1, -1, 1109, 1480, 622, 1661, 1976, 370, 1245, 311, 1865, 277 },
        { 4435, 3322, 2637, 5274, -1, 5920, 6645, 7459, -1, 7902, -1, -1, 3951,
          3729, -1, -1, -1, -1, 4978, -1, -1, 2960, -1, 2489, -1, 2217 }
    }
};
```



Oral Test

Oral Test

Q 4.1 표준입력장치 (키보드)로 부터의 개별 문자 입력에서 입력 버퍼를 사용하는 방식과 입력 버퍼를 사용하지 않는 방식의 차이점을 설명하고, 각각 어떤 함수를 사용하여야 하는지에 대하여 설명하라.

(**Keypoint:** getchar()와 _getch() 비교)

Q 4.3 입력 문자로부터 매핑 테이블의 인덱스 (alt_on, shift_on, ch_index)를 산출하는 방법에 대하여 설명하라.

(**Keypoint:** GetSynchKeyState() 함수, VK_MENU, VKVK_CONTROL)

Q 4.3 전자피아노 구현에서 mappingKey_to_Freq() 함수를 switch – case 구조로 구현하는 경우와 key_to_freq 매핑 테이블을 사용하는 구조로 구현하는 경우의 장단점을 비교하여 설명하라.

(**Keypoint:** 필요한 연산 (비교, 계산 등)의 최대, 최소, 평균 회수를 설명)

Q 4.4 키보드와 Beep() 함수를 사용하여 전자피아노를 구현할 때, 전자피아노 음에 설정되어 있는 않는 문자가 입력되는 경우에 어떻게 오류 처리를 하는가에 대하여 설명하라.

(**Keypoint:** switch-case 문에서 처리 방법과 Key_to_Freq[][][] 매핑 테이블에서의 처리 방법을 각각 설명)

