

배열에 대하여 배웁니다.

배열 함수 사용법을 익힙니다.

웨이브폼 그래프와 차트에 대하여 배웁니다.

CHAPTER 4. 배열

1. 배열

배열의 정의와 사용법

배열

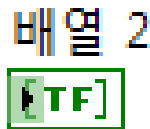
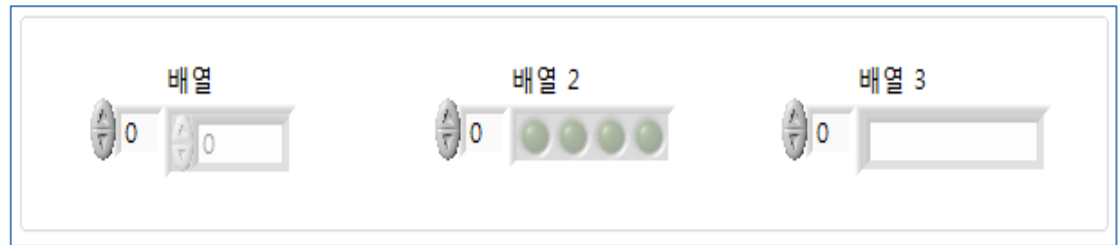
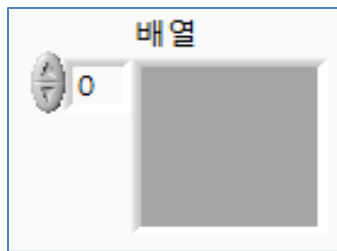
- 배열은 같은 타입의 데이터 원소를 그룹화한 것
- 배열은 원소와 차원으로 구성된다.
 - 원소는 배열을 구성하는 데이터
 - 차원은 배열의 길이, 높이, 또는 폭
- 배열은 하나 또는 그 이상의 차원을 가질 수 있다.
- 한 차원마다 최대 2,147,483,647개의 원소를 가질 수 있다. (Integer 32)

배열

- 배열 인덱스 (Index)
 - 사용자가 특정 원소에 접근할 수 있게 하기 위하여 **인덱스(Index)**를 사용한다.
- 인덱스는 0부터 시작
 - 예를 들어, $N=8$ 인 8개의 태양계 행성에 대하여 인덱스는 0에서 7이 된다. 지구는 세 번째 행성이므로 인덱스는 2이다.

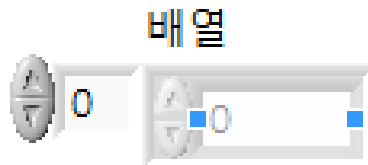
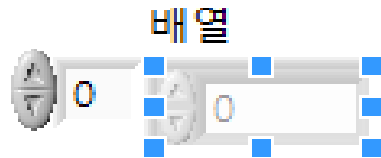
배열 만들기

- 빈 배열은 배열의 틀만 가지고 있다.
- 숫자형, 불리언, 문자열, 경로, 참조 번호, 클러스터 컨트롤 또는 인디케이터 등의 데이터 객체 또는 원소를 끌어다 놓음.



배열의 모양 바꾸기

- 배열의 원소는 기본적으로 한 개만 디스플레이 한다.
- 디스플레이 되는 원소의 개수를 늘여줄 수 있다.
 - 바깥쪽 가장 자리를 잡으면, 디스플레이 되는 원소의 개수를 늘여 줄 수 있다.



배열의 원소 개수

- LabVIEW에서는 배열을 만들 때, 원소의 데이터 타입과 컨트롤/인디케이터, 객체의 모양만 지정하고 원소 개수는 지정하지 않는다.
- LabVIEW에서 배열의 원소 개수는 다이내믹하게 할당된다.



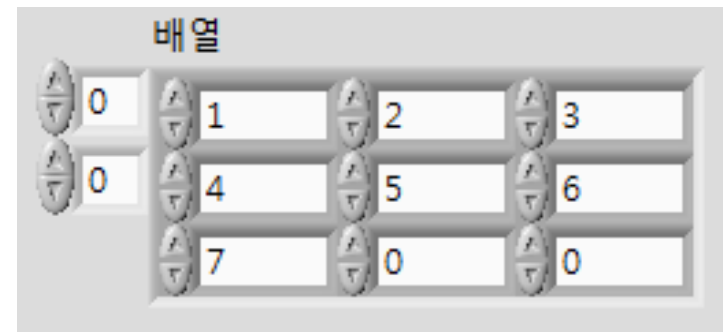
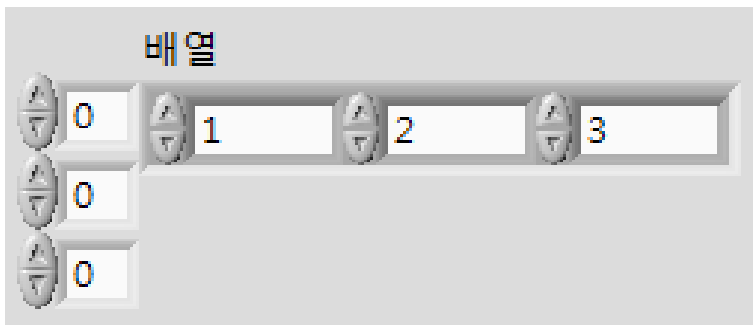
인덱스

- 배열의 차원당 한 개의 인덱스가 필요하다.
- LabVIEW에서 인덱스는 배열을 탐색하고, 블록다이어그램 함수에서 원소나, 행, 열을 인덱싱하기 위해 사용된다.
 - 인덱스는 0부터 시작하여, 1씩 증가한다.
 - 인덱스는 첫 번째 위치에 있는 원소의 인덱스를 나타낸다.
아래 그림에서 첫 번째 원소의 인덱스가 99이다.

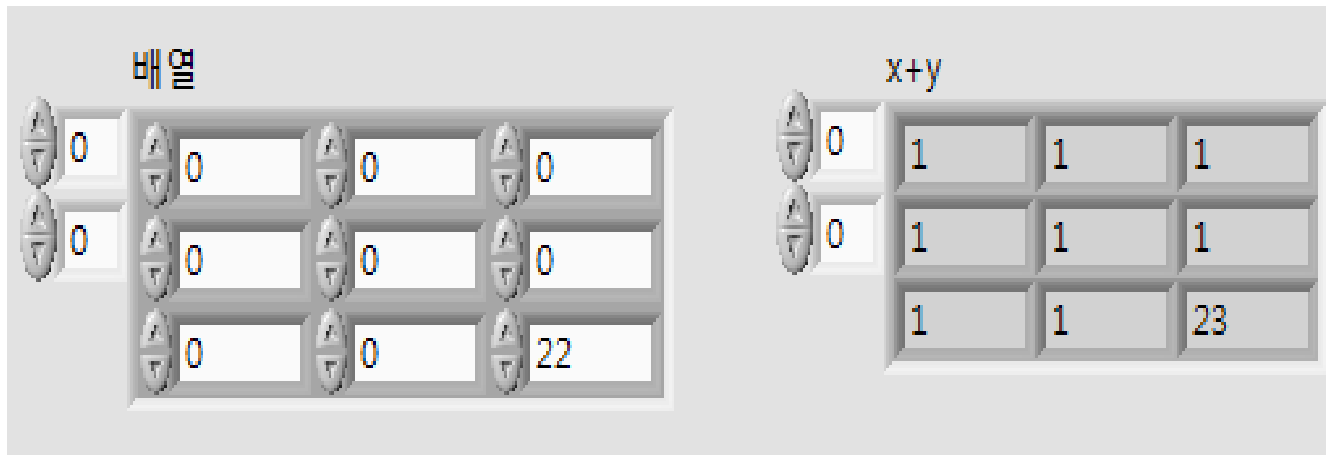
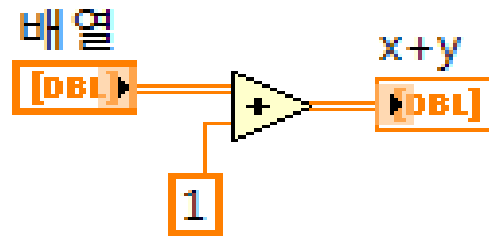


다차원 배열 만들기

- 2차원 이상의 배열을 만들고자 하는 경우에는 배열 인덱스의 단축메뉴에서 [차원 추가]를 선택하여 한 차원씩 높여 나갈 수 있다.
- 반대로 [차원 제거]를 선택하여 한 차원씩 낮출 수 있다.
- 차원이 증가하면 인덱스의 개수가 증가한다.

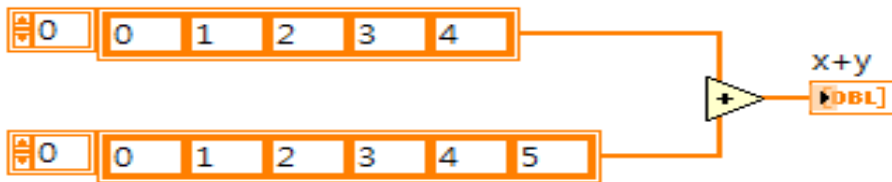


실습 4-1: 배열 만들기

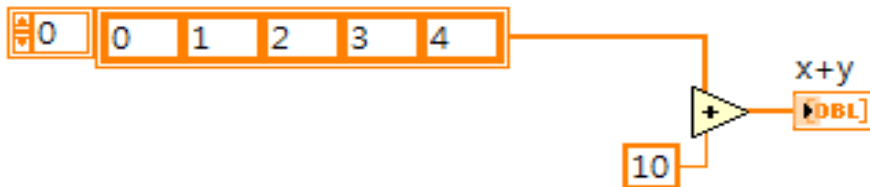


숫자형 배열의 연산

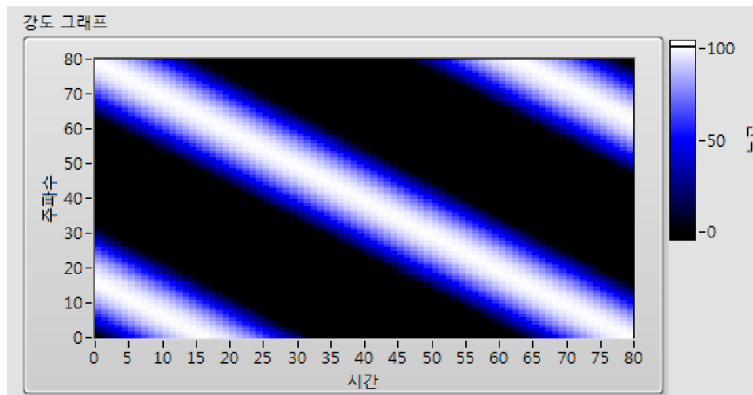
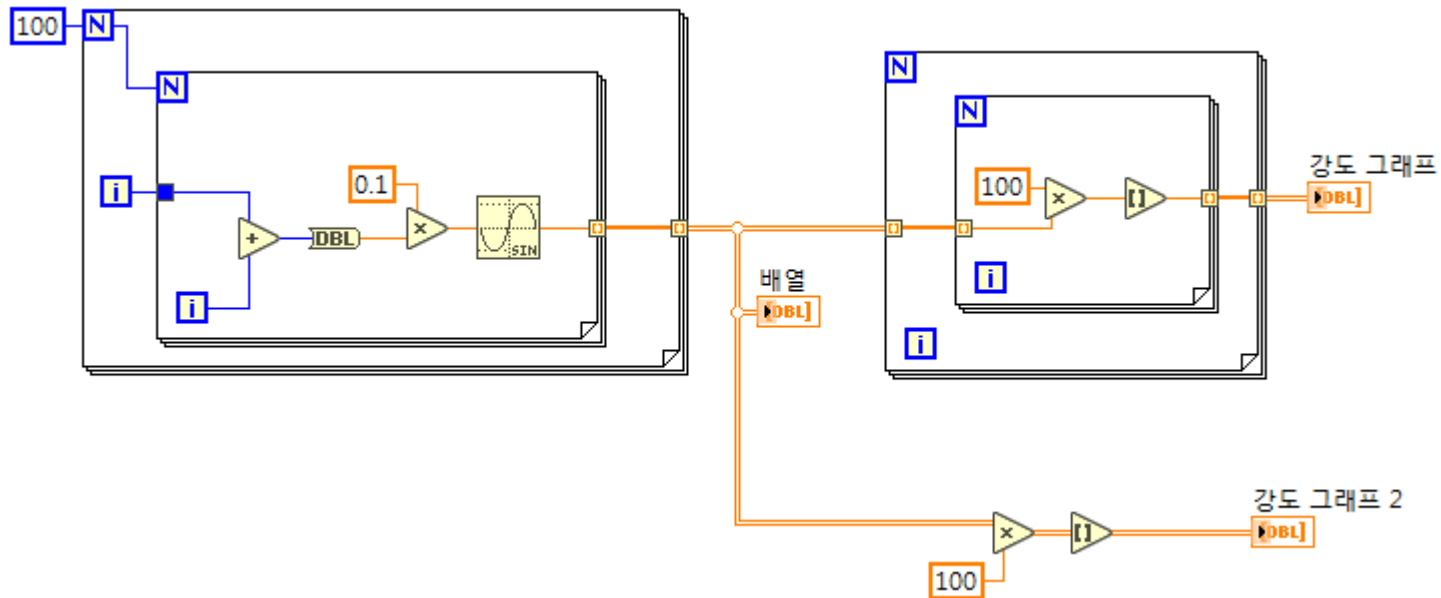
- 1D배열 + 1D배열 = 1D배열



- 1D배열 + 스칼라 = 1D배열



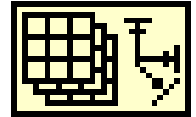
실습 4-2: 강도 그래프 사용하기



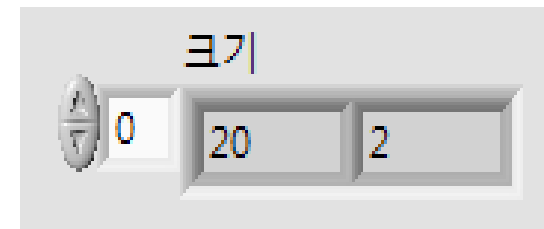
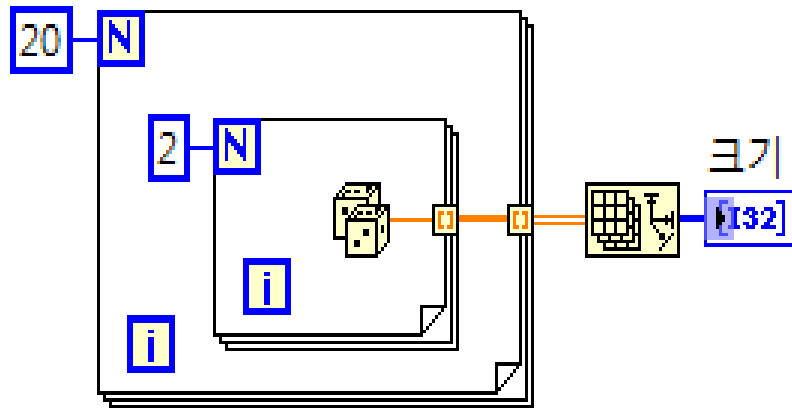
2. 배열 함수

배열 함수 사용법

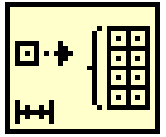
배열 크기



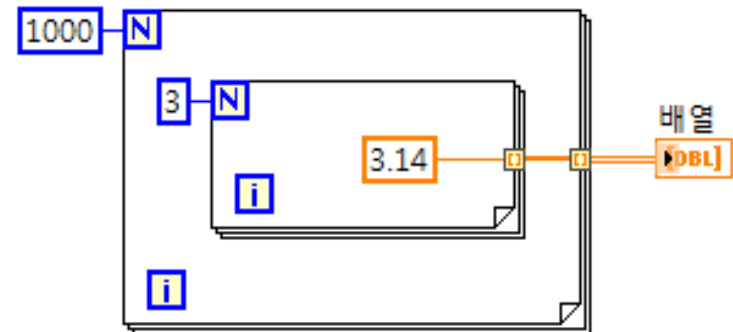
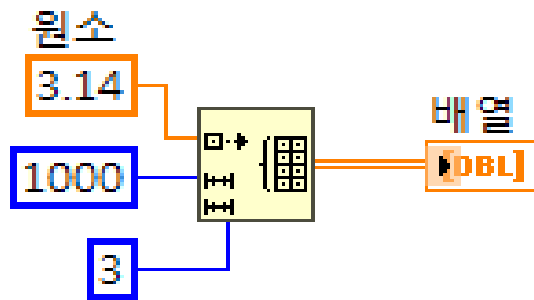
- 배열의 크기를 계산하는 함수
- 배열의 원소 개수를 반환한다.
 - 1차원 배열은 1개
 - 2차원 배열은 2개
 - 3차원은 3개의 크기를 반환



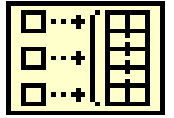
배열 초기화



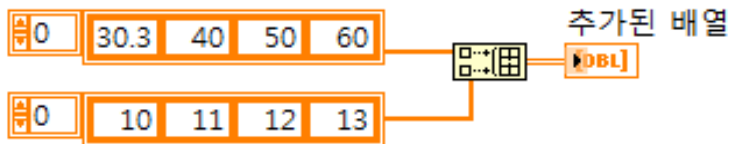
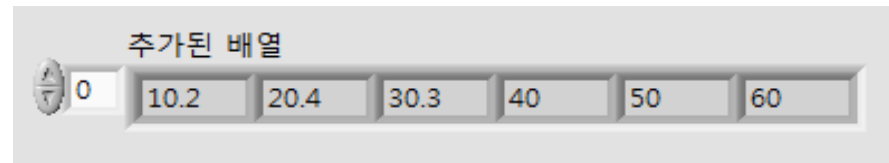
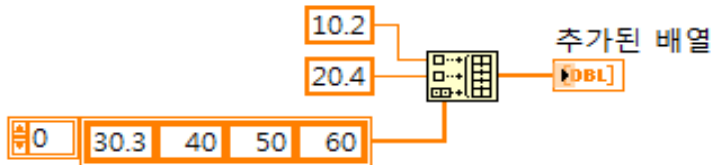
- ‘원소’ 입력으로 초기화된 n차원 배열을 생성합니다.



배열 만들기

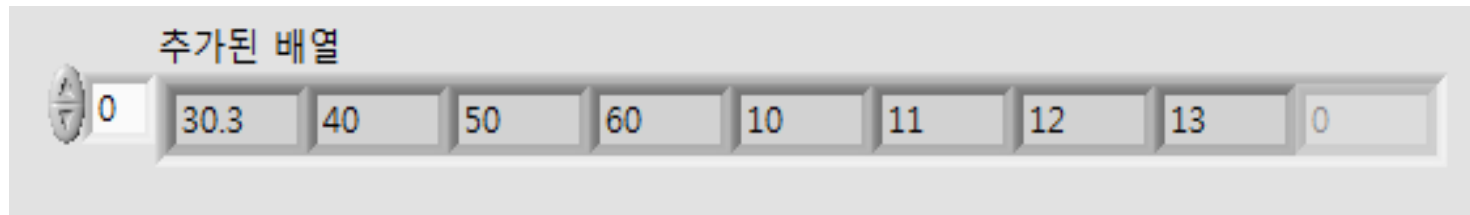
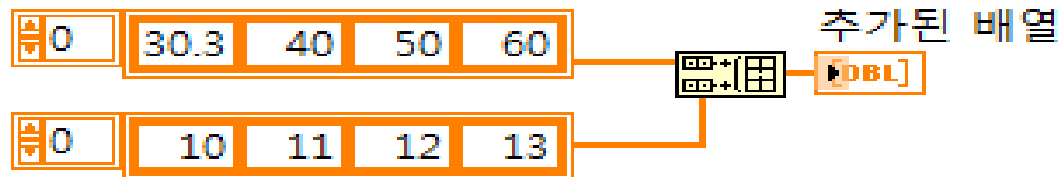


- 배열 만들기 함수를 이용하여 여러 배열을 연결한다.
- 원소나 부분배열을 배열에 추가한다.
- 배열 만들기의 단축메뉴에서 [입력 추가]를 선택하거나, 마우스로 선택해서 크기를 키워준다.

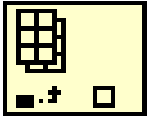


배열 만들기 - 입력 연결

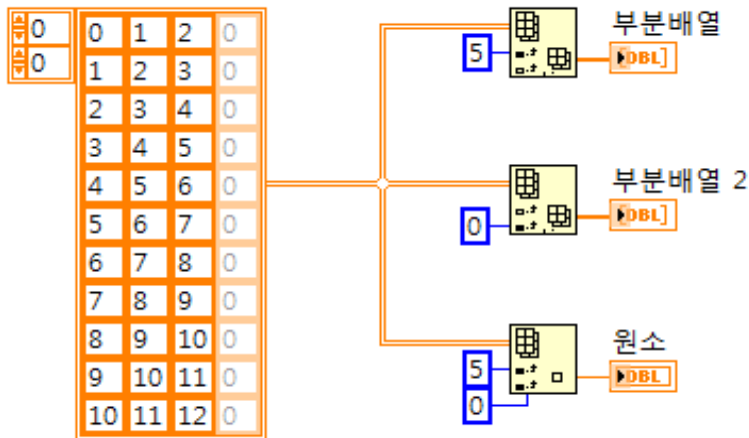
- [단축메뉴] 입력 연결
- 1차원 배열을 연결하여, 연결된 1차원 배열을 생성



배열 인덱스



- 배열에서 지정한 인덱스의 원소 또는 부분배열을 반환한다.
- 1차원 배열은 원소를 반환하고, 2차원 배열은 원소 또는 부분 배열을 반환한다.



부분배열

0 5 6 7 0 0 0

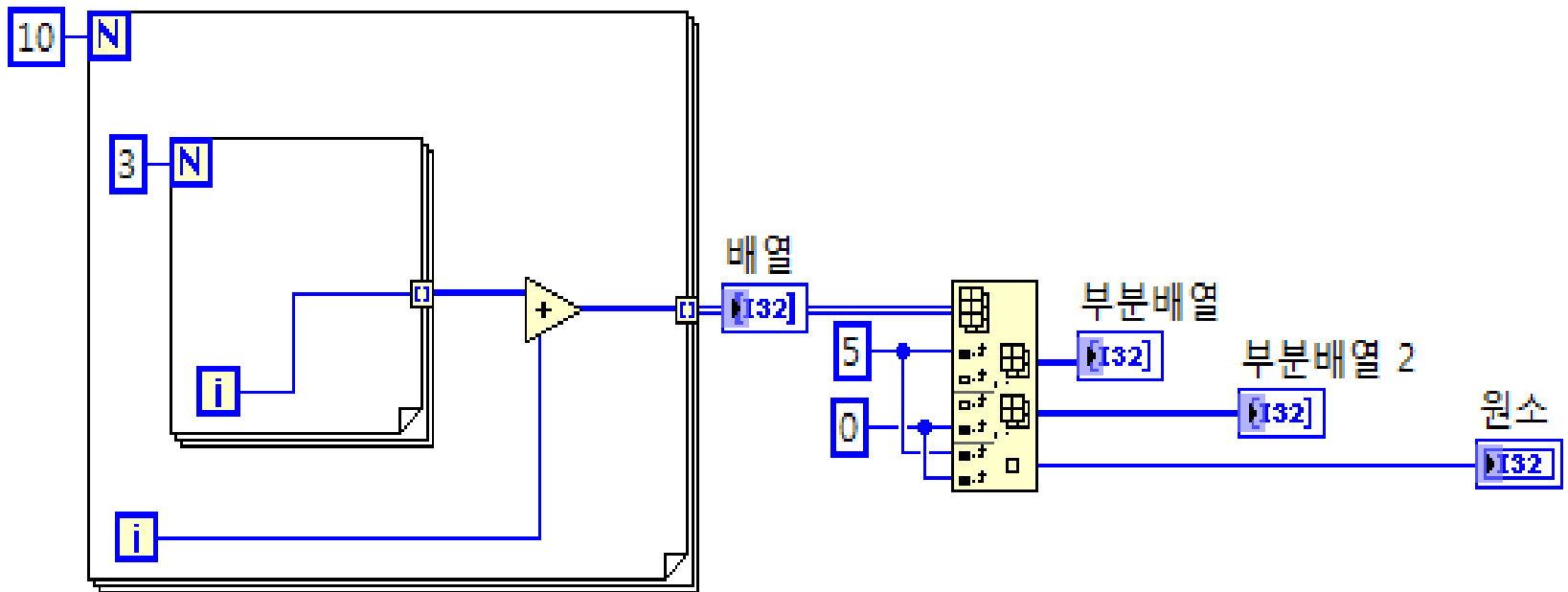
부분배열 2

0 0 1 2 3 4 5

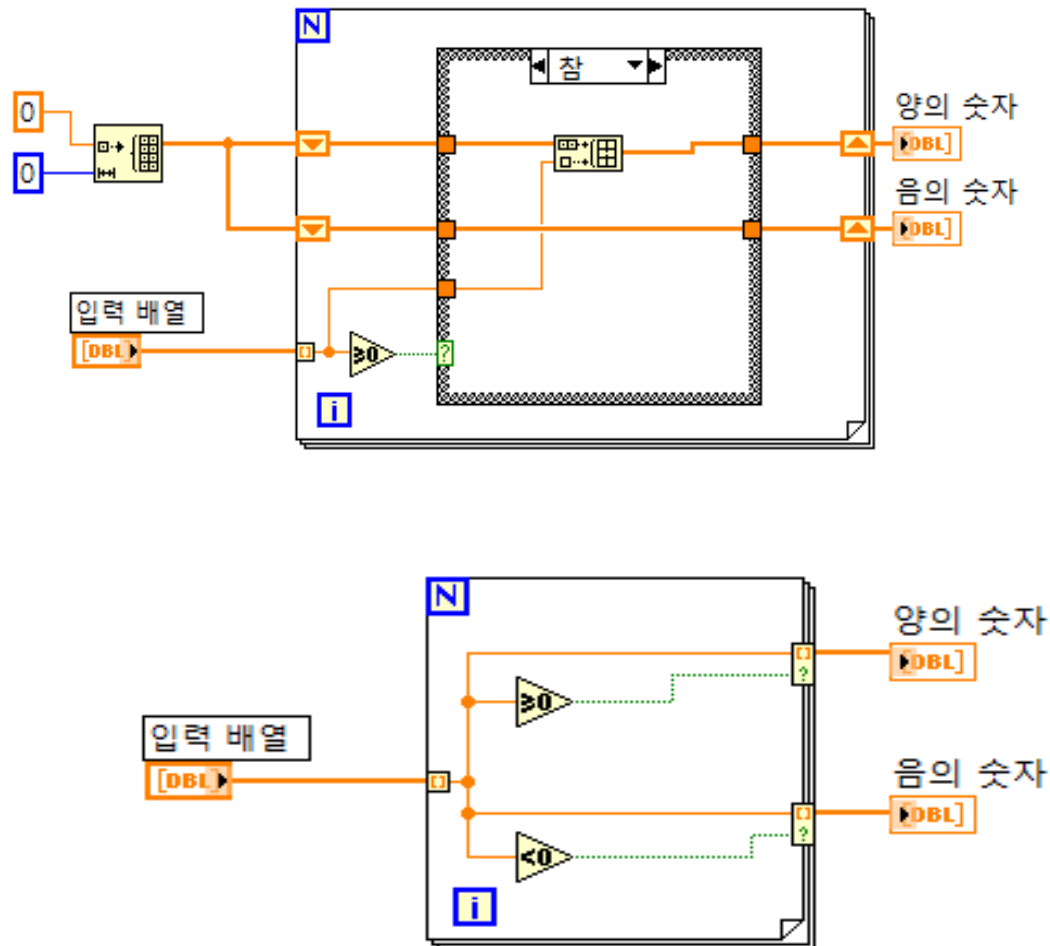
원소

5

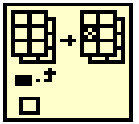
실습 4-3: 배열 인덱스 함수 사용



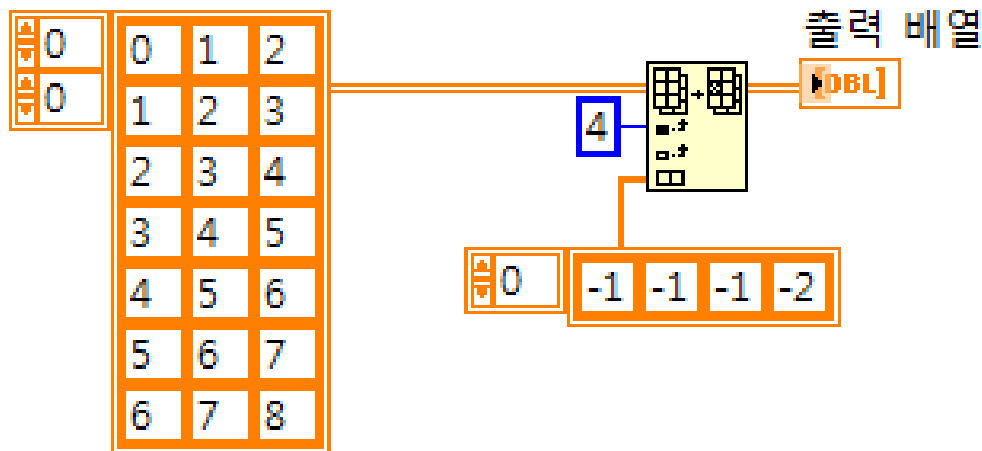
실습 4-4: 배열 원소 분류하기



배열 부분 대체



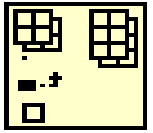
- 인덱스에서 지정하는 포인트에서 원소 또는 부분배열을 대체한다.



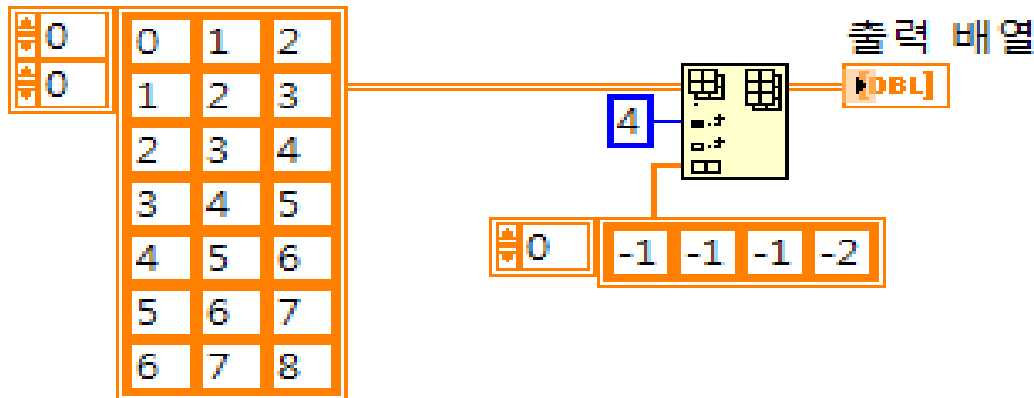
출력 배열

0	0	1	2	0
0	1	2	3	0
	2	3	4	0
	3	4	5	0
	-1	-1	-1	0
	5	6	7	0
	6	7	8	0

배열 삽입



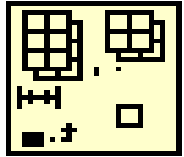
- 인덱스에서 지정한 포인트에 원소 또는 부분 배열을 삽입한다.



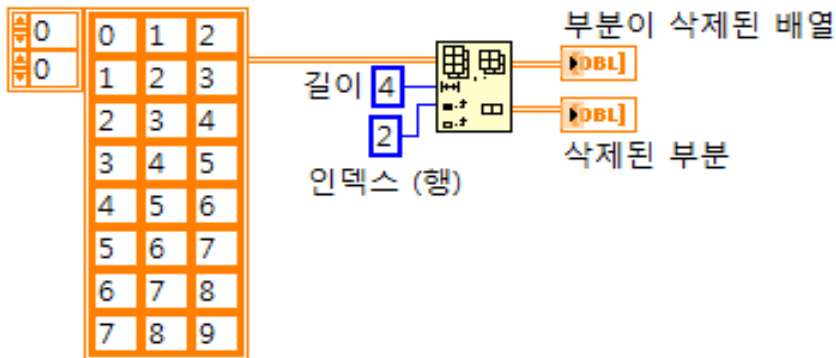
출력 배열

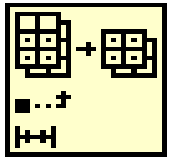
0	1	2	0
1	2	3	0
2	3	4	0
3	4	5	0
-1	-1	-1	0
4	5	6	0
5	6	7	0
6	7	8	0

배열로부터 삭제



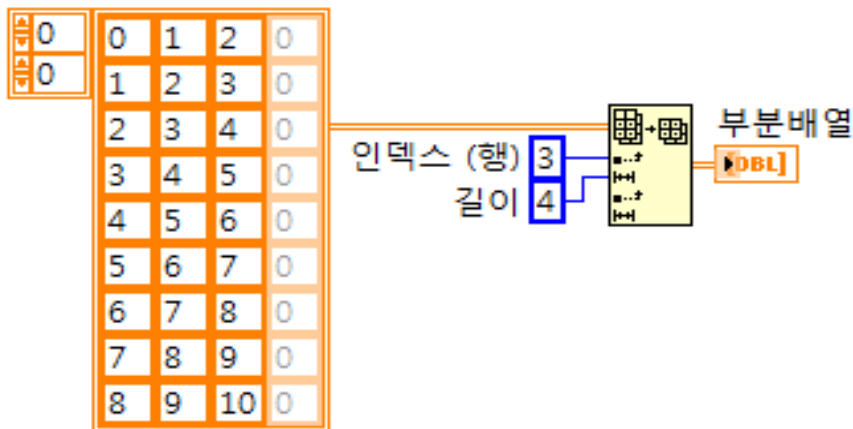
- 원소 또는 부분배열을 인덱스에서부터 시작하여 삭제한다.
- 부분이 삭제된 배열에서 편집된 배열을 반환하고, 삭제된 부분에서 삭제된 원소 또는 부분배열을 반환한다.
- 입력된 배열과 같은 차원의 배열 두 개를 출력한다.





배열 잘라내기

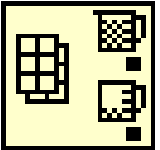
- 인덱스에서 시작하여 길이만큼을 잘라서 반환한다.
- 인덱스와 길이는 배열의 차원만큼 지원된다.
- 1차원 배열은 한 개의 인덱스와 길이가 제공되고, 2차원 배열은 두 개의 인덱스와 길이가 지원된다.



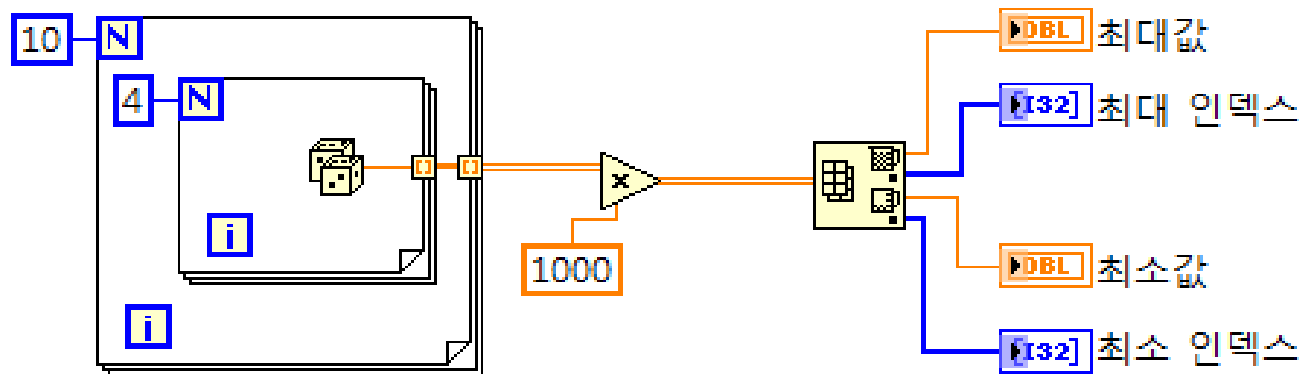
부분배열

0	3	4	5
0	4	5	6
	5	6	7
	6	7	8

배열 최대 & 최소

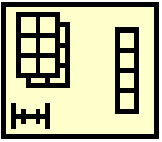


- 배열의 최대값과 최소값을 반환한다.
- 각 값의 인덱스도 함께 반환한다.

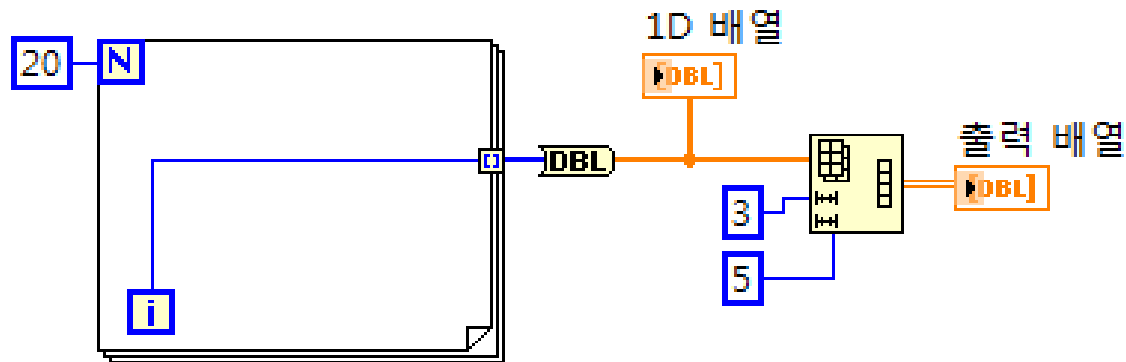


최대값	최대 인덱스
968.331	0 2 3
최소값	최소 인덱스
47.858	0 4 1

배열 차원 변경



- [n차원 배열] 입력과 [차원 크기] 입력을 받아서, [m차원 배열]을 출력한다.
- [차원 크기] 입력에 따라 배열의 차원을 변경한다.

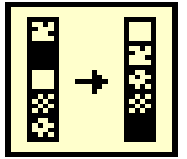


The front panel shows the input and output array dimensions. The '1D 배열' (1D Array) input is set to 0. The '출력 배열' (Output Array) is a 3D array with dimensions 0, 5, 10, 0, 0, 0.

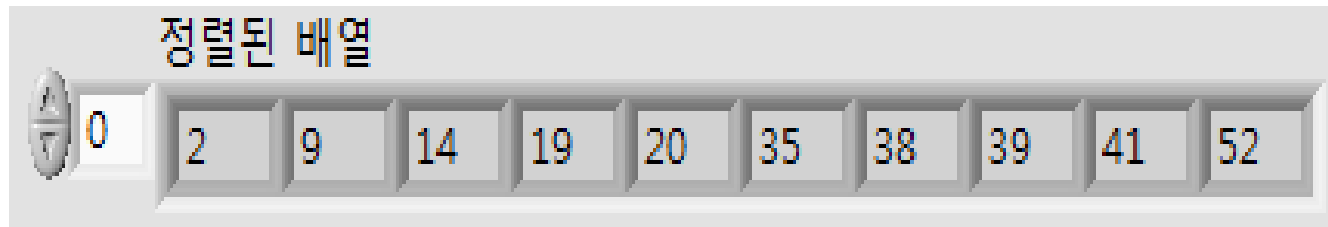
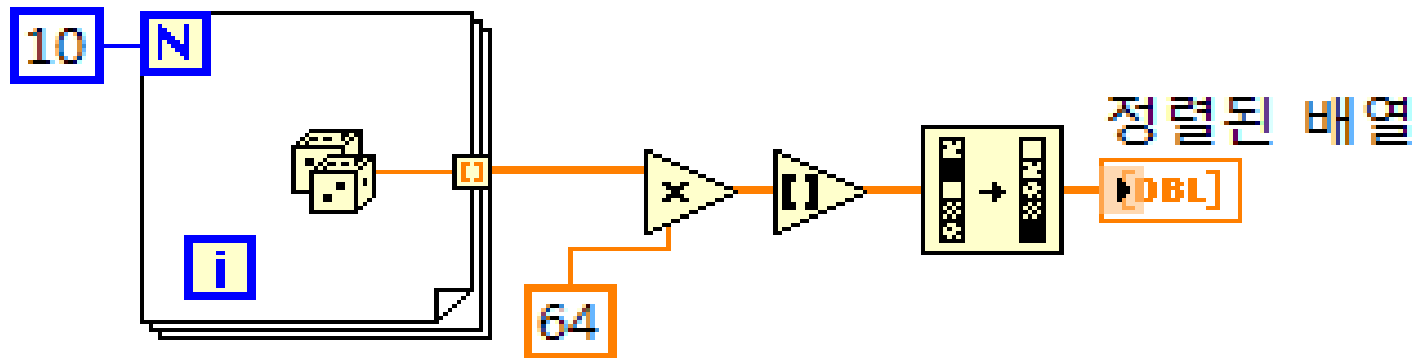
1D 배열					
0	0	1	2	3	4

출력 배열						
0	0	1	2	3	4	0
0	5	6	7	8	9	0
	10	11	12	13	14	0
	0	0	0	0	0	0

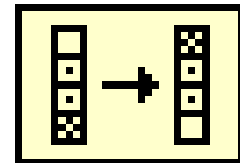
1D 배열 정렬



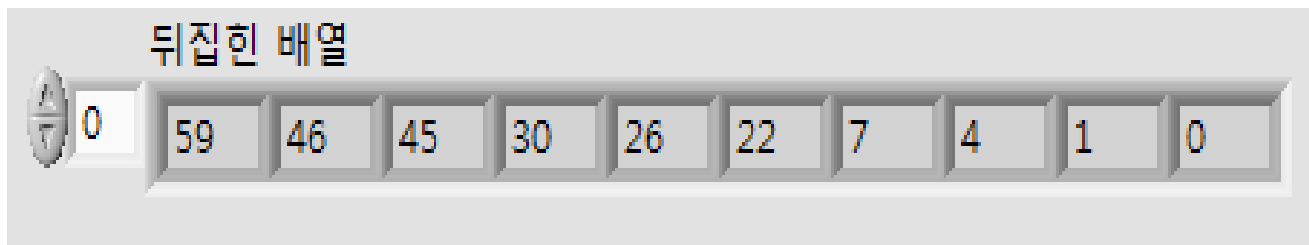
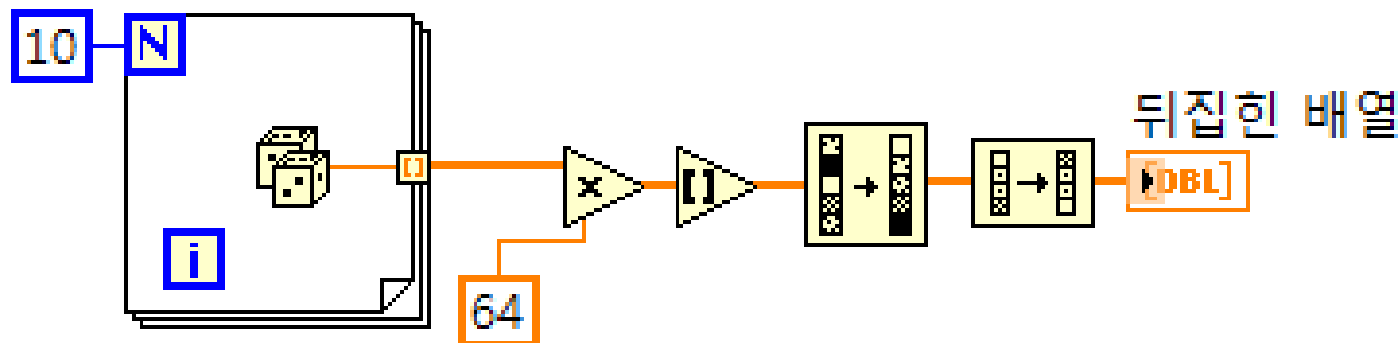
- 오름차순으로 정렬한다.



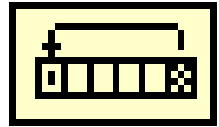
1D 배열 뒤집기



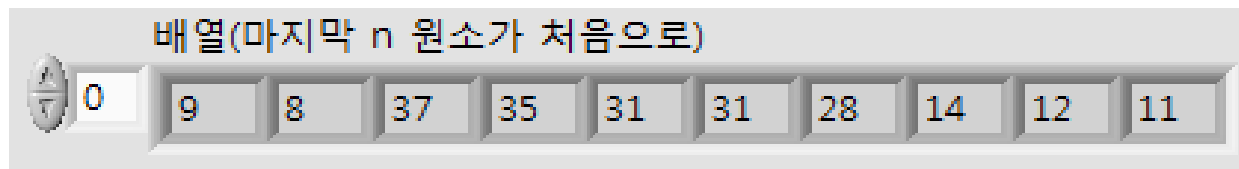
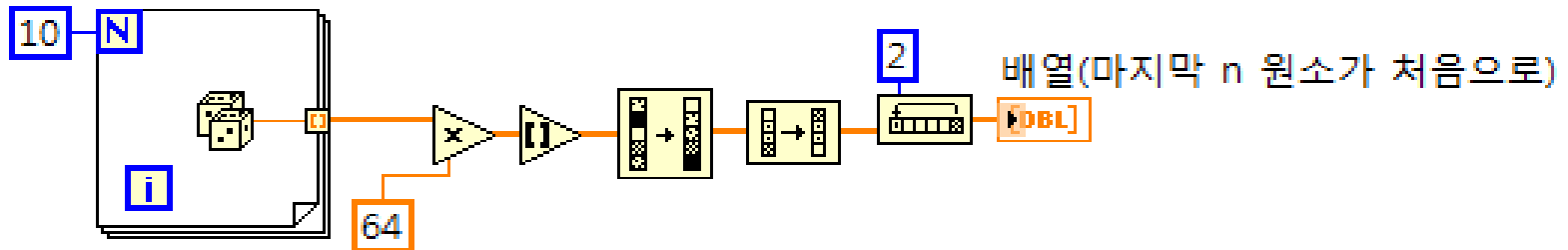
- 배열의 원소의 순서를 반대로 한다.



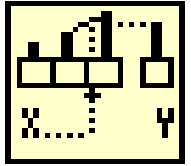
1D 배열 회전



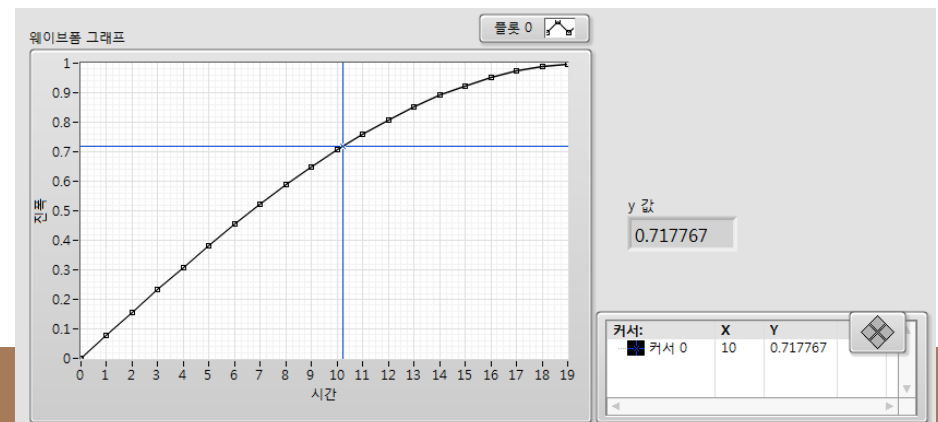
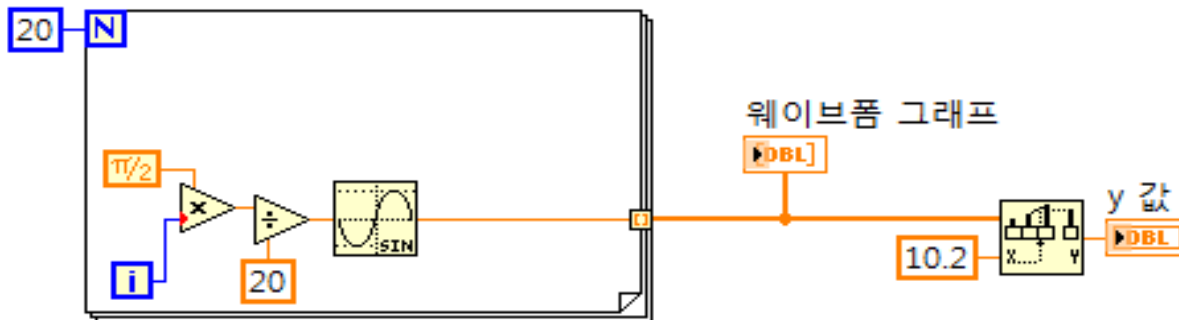
- 배열의 원소를 n 에 의해 지정된 횟수와 방향으로 회전한다.



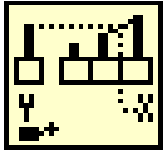
1D 배열 보관



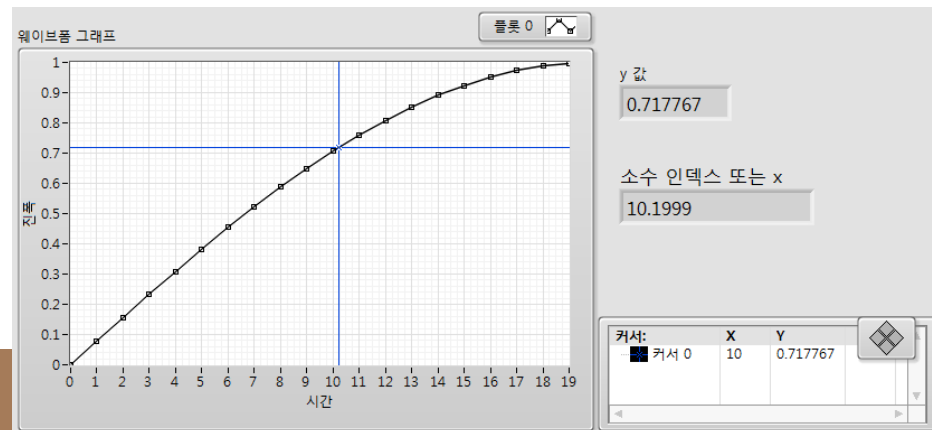
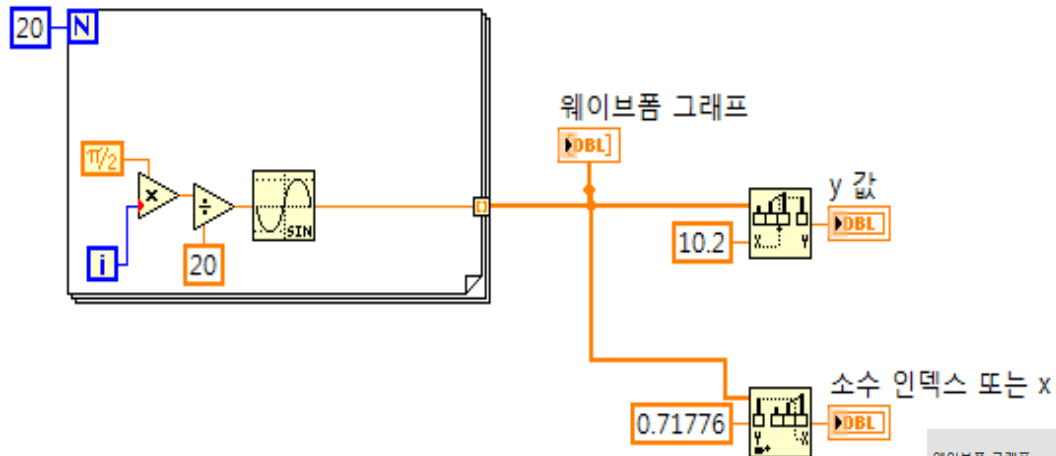
- 배열의 인덱스는 정수형 데이터이다.
- 인덱스와 인덱스 사이의 소수 인덱스에 대응하는 보관된 원소 값을 필요로 하는 경우에 1D 배열 보관을 사용한다.



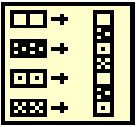
1D 배열 임계점



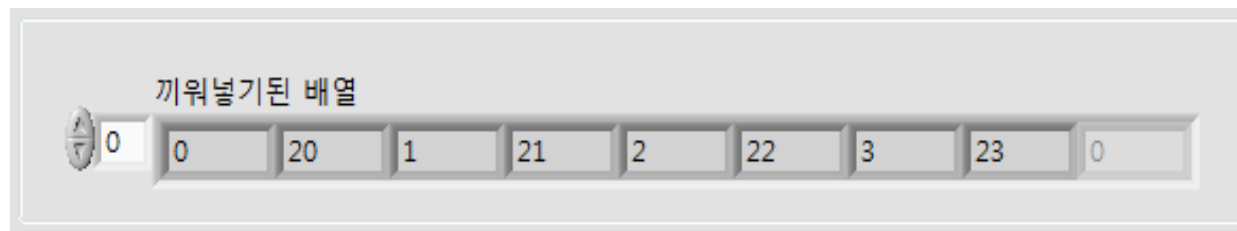
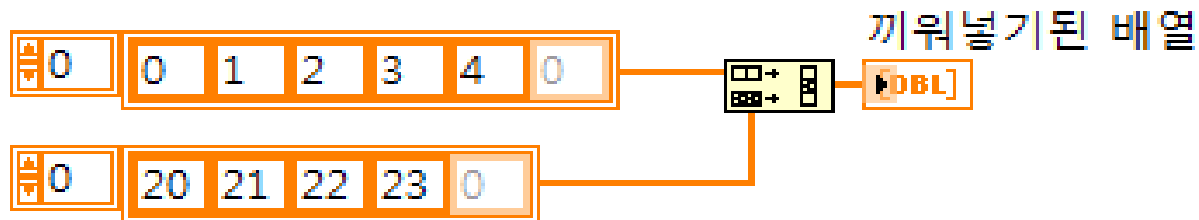
- [임계점 y]와 시작 인덱스를 입력 받아서, 그 값에 대응하는 첫 번째 [소수 인덱스 또는 x] 값을 반환한다.
- 1D 배열 보간과 반대의 경우이다.

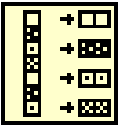


1D 배열 끼워넣기



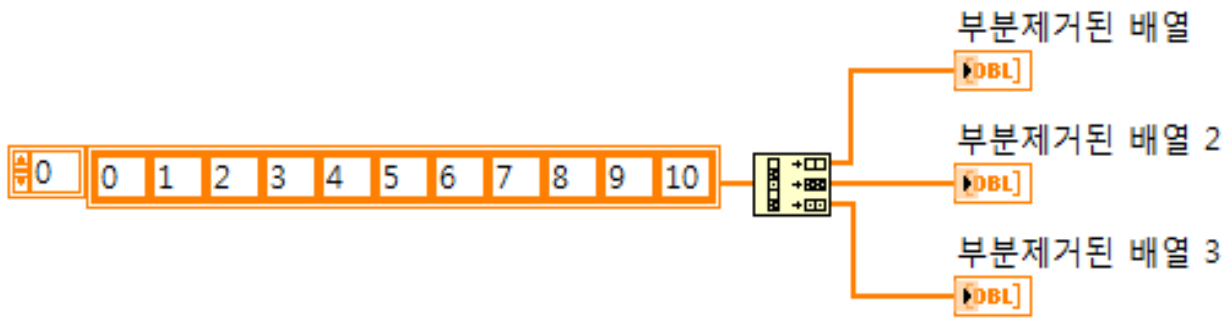
- 입력된 1D 배열들의 원소를 하나씩 끼워 넣어서, 새로운 1D 배열을 만든다.





1D 배열 부분제거

- 끼워넣기 함수의 반대 개념이다.
- 한 개의 1D 배열을 여러 개의 1D 배열로 나눈다.
- 인덱스 별로 하나씩 제거하면서 나누는 방식을 사용한다.

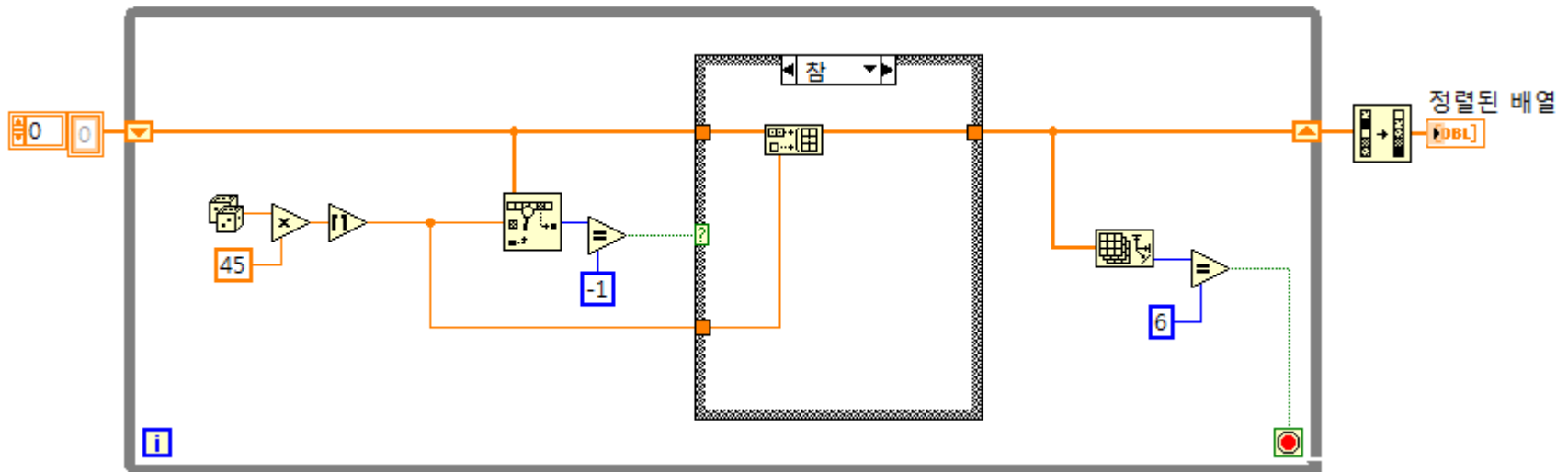


부분제거된 배열					
0	0	3	6	0	0

부분제거된 배열 2					
0	1	4	7	0	0

부분제거된 배열 3					
0	2	5	8	0	0

실습 4-5: 로또 6/45



3. 그래프

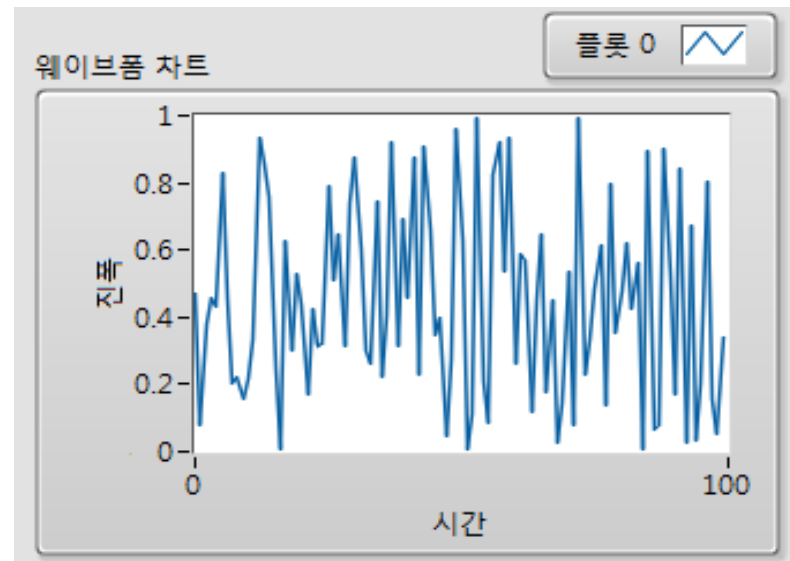
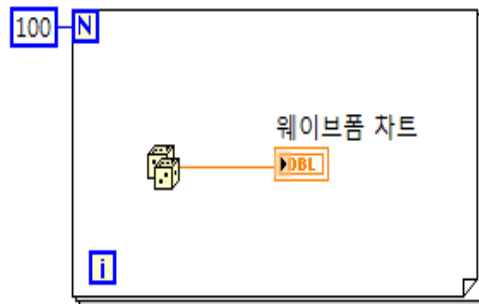
웨이브폼 차트

웨이브폼 그래프

XY 그래프

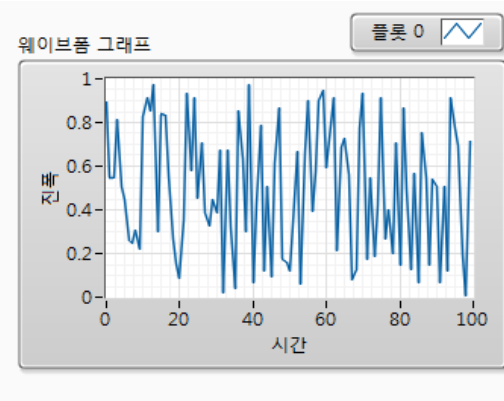
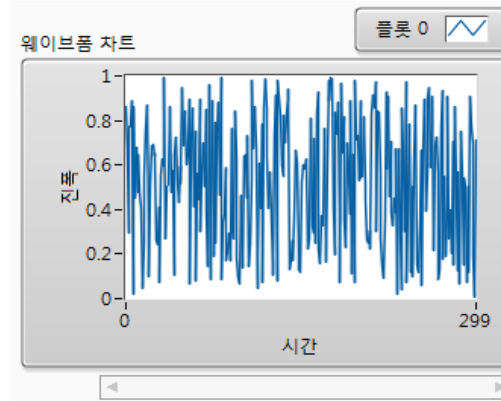
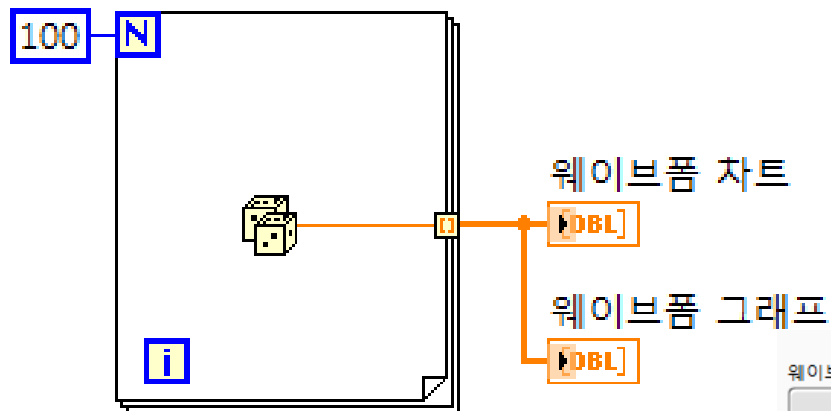
웨이브폼 차트

- 숫자형 인디케이터
- 예전 데이터, 즉 히스토리 데이터까지 보여주는 점에서 숫자형 인디케이터와 차이.
- 히스토리 길이의 기본값은 **1024**이다. 단축메뉴에서 [차트 히스토리 길이]를 선택하여 바꿀 수 있다.



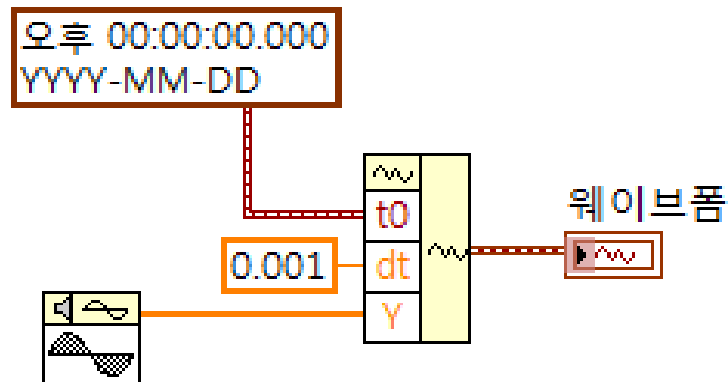
웨이브폼 그래프

- 배열 형태의 데이터를 플롯
- 1차원 숫자형 배열, 2차원 숫자형 배열, 웨이브폼 데이터 등을 플롯

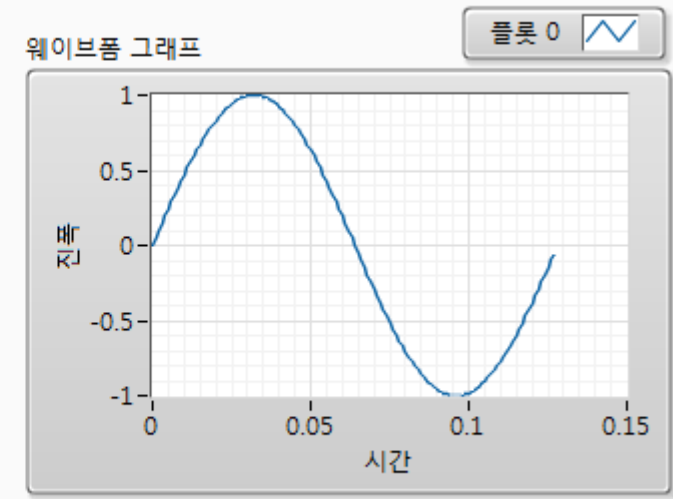
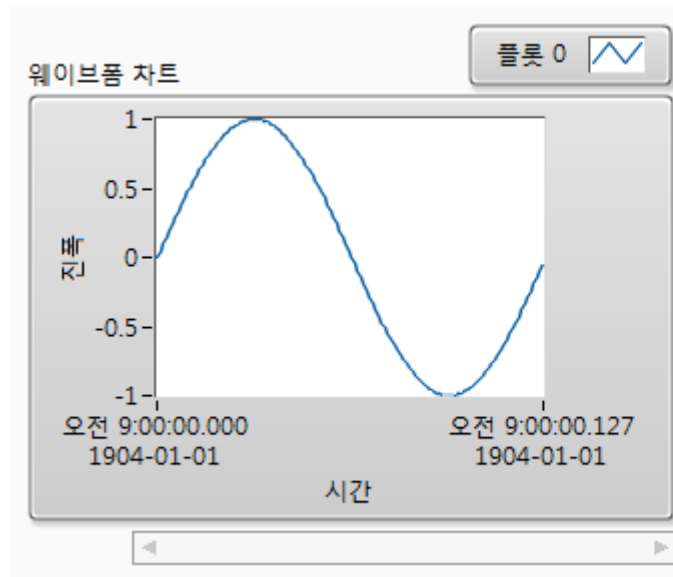
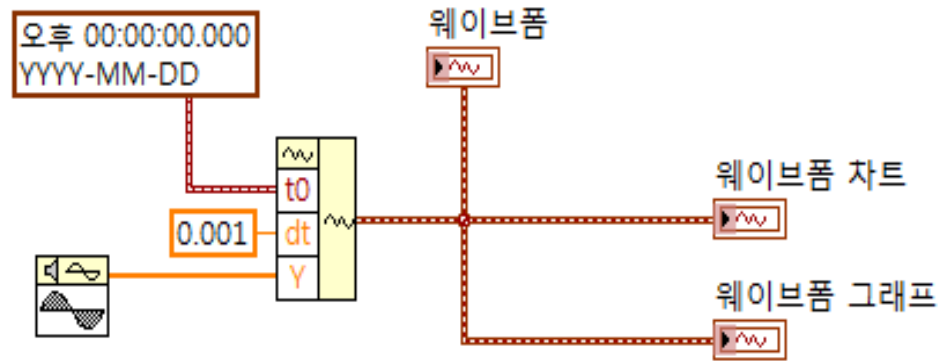


웨이브폼 데이터

- 웨이브폼 데이터는 Y 배열, 초기 시간 t_0 , 시간 간격 dt , 그리고 속성 (attributes) 배리언트로 구성된 데이터의 구조체이다.

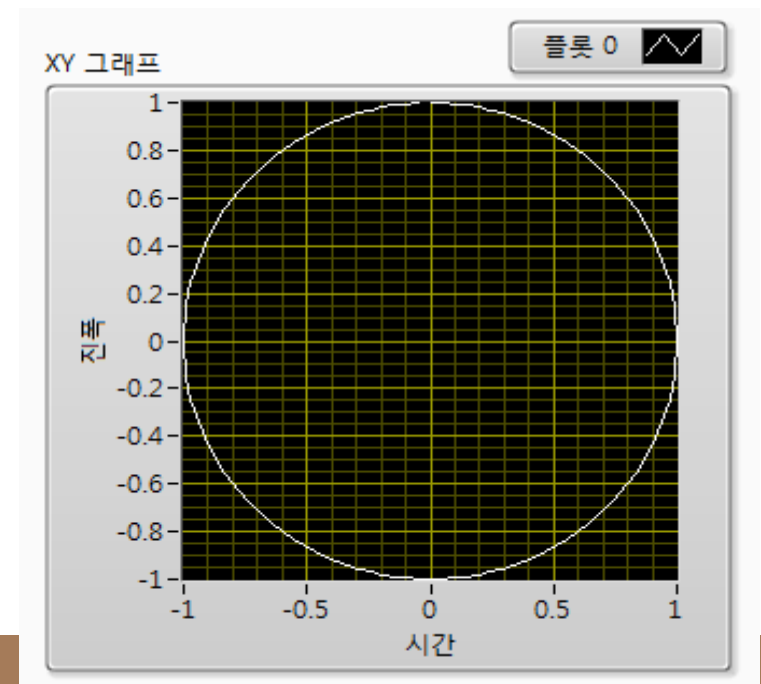
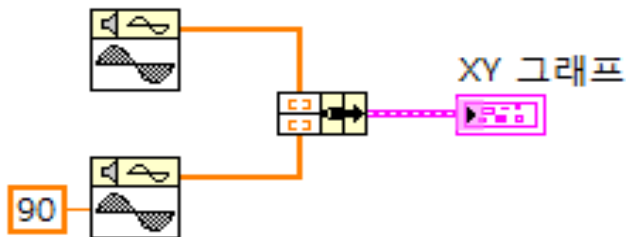


웨이브폼	
t0	Y 0
오후 00:00:00 YYYY-MM-DD	0
dt	0.049068
0.001000	0.098017
	0.14673
	0.19509

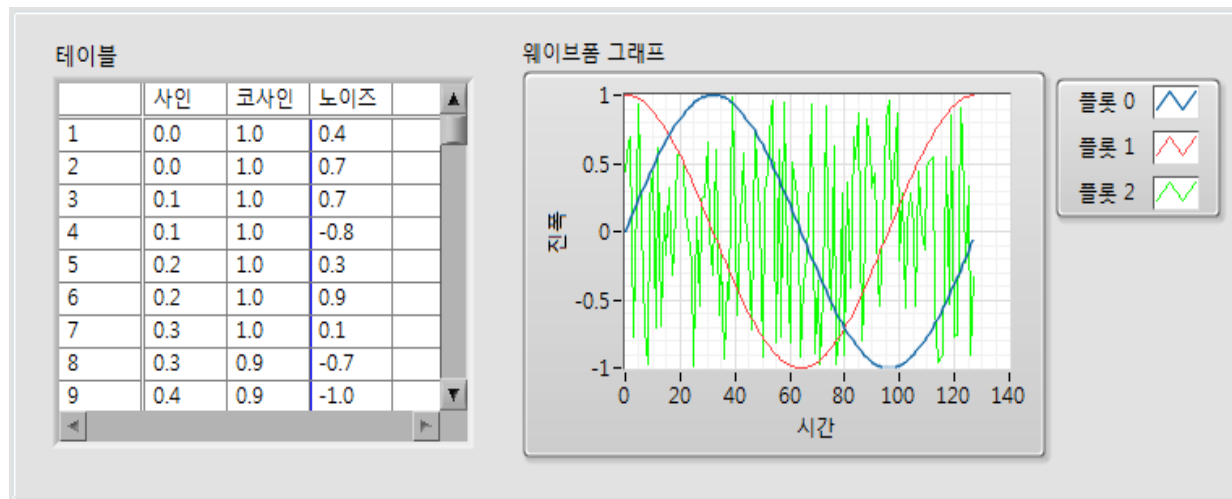
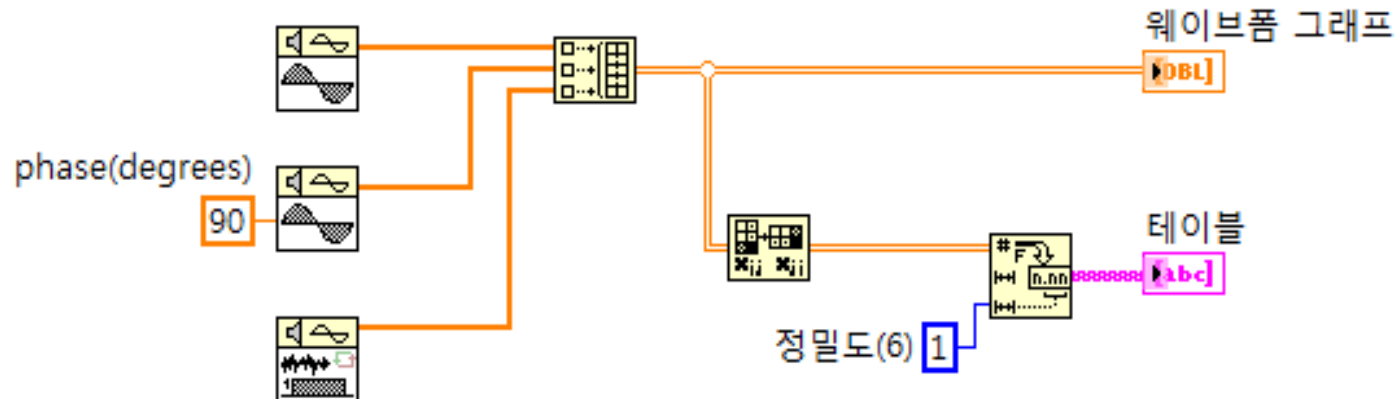


XY 그래프

- 웨이브폼 그래프나 차트는 X축이 일정한 간격이라는 것을 전제함.
- 시간 축이 임의의 간격인 경우에는 XY 그래프를 이용하여 플롯 .
- 두 개의 배열은 **묶기** 함수를 사용하여 묶어줘야 됨.



실습 4-6: 웨이브폼 그래프 플롯하기



실습 4-7: 웨이브폼 데이터 타입

