07\_배열

학습목표

-배열의 개념

-배열의 선언과 초기화

-일차원 배열

-배열과 문자열

-다차원 배열

1. 배열이란?

지금까지 학습하였던 변수는 오직 하나의 값만을 저장할 수 있었다. 하지만 복잡한 응용 프로그램에서는 많은 값을 한꺼번에 저장할 수 있는 저장 장소가 필요하다. 배열은 이런 목적으로 만들어진 자료형이다. 배열을 사용하면 여러 개의 변수를 생성할 수 있다.

변수 선언이 단독 주택이라면 배열은 아파트 단지라고 할 수 있다.

배열(Array)은 동일한 타입의 데이터가 여러 개 저장되어 있는 데이터 저장 장소이다. 배열 안에 들어있는 각각의 데이터들은 번호에 의하여 식별된다. 이 번호를 인덱스(index) 라고 한다. 배열에 저장된 데이터를 배열 요소(array element)라고 한다. 배열에서는 인덱스를 이용하여 배열 요소에 접근한다. 배열은 메모리의 연속적인 공간에 저장된다. 예를 들어 배열 요소 s[0]과 s[1]은 실제 메모리에서도 서로 붙어 있다.

배열을 이용하면 여러 개의 값을 하나의 이름으로 처리할 수 있다.

배열을 사용하려면 먼저 배열을 선언하여야 한다.

형식: 자료형 배열이름[배열크기];

Ex) int s[10]; double f[50];

요소들의 번호는 0부터 시작하기 때문에 int s[10]를 선언하면 s[10]은 없다.

배열 요소 접근

배열의 요소에 접근하려면 scores[5]와 같이 대괄호 안에 요소의 인덱스(번호)를 적어주면 된다. 배열 요소는 변수와 100% 동일하다. 따라서 값을 배열 요소에 저장할 수 있고 배열 요소에 저장된 값을 꺼낼 수도 있다.

1. 배열의 초기화

배열을 초기화하려면 값들의 리스트를 콤마로 분리하여 중괄호 {}로 감싼 후에 이것을 배열을 선언할 때 대입해주면 된다. 즉 배열은 선언하면서 동시에 초기화할 수 있다.

+Tip 배열 요소의 개수를 계산하는 방법

배열에 들어있는 자료를 처리하려면 배열의 처음부터 끝까지 반복하여야 하는 경우가 많다. 따라서 배열의 크기는 꼭 알아야 하는 정보이다. 배열 안에 들어 있는 요소의 개수를 자동적으로 계산하는 방법이 있다. 바로 sizeof연산자를 사용하는 것이다. 우리가 알다시피 sizeof연산자는 자료형이나 변수의 크기를 바이트 단위로 계산하는 연산자이다. sizeof연산자를 이용하여 배열 전체의 크기를 구하고 이것을 배열 요소의 크기로 나누게 되면 배열 요소가 몇 개나 있는지 쉽게 계산할 수 있다.

형식: size = sizeof(s)/sizeof(s[0]); >> 배열의 크기 = 전체 바이트의 크기/배열 요소 하나의 크기

1. 버블 정렬

배열과 관련된 중요한 개념이 바로 정렬이다. 정렬(sorting)은 물건을 크기순으로 나열하는 것을 의미한다. 예를 들어 책들은 제목순, 저자순 또는 발간연도순으로 정렬아 가능하다. 정렬은 컴퓨터 공학에서 가장 기본적이고 중요한 알고리즘 중의 하나로 일상생활에서 많이 사용된다. 우리는 버블 정렬에 대해 공부해볼 것이다. 버블 정렬은 가장 효율적인 정렬 방법은 아니지만 가장 이해하기 쉽다.

크기순으로 버블 정렬할 경우

1. 인접한 블록 2개를 비교하여서 크기 순서대로 되어 있지 않으면 위치를 바꾼다.

이렇게 하면 가장 큰 블록이 가장 끝에 배치된다.

1. 다시 인접한 블록 2개를 비교하여서 순서대로 되어 있지 않으면 위치를 바꾼다. 두 번째 패스가 끝나면 2번째로 큰 블록이 가장자리에 2번째 위치에 놓이게 된다.
2. 위의 과정을 블록의 개수만큼 되풀이하면 모든 블록이 크기순으로 정렬된다.
3. 다차원 배열

지금까지 사용한 배열은 1차원 배열이었다. 다차원 배열이란 배열 요소를 다차원으로 가질 수 있는 배열을 뜻한다. 2차원, 3차원, n차원 배열이 가능하다. 그러나 다차원이 되면 필요한 메모리 공간이 급격하게 늘어나게 되므로 주의해야 한다. 특별한 경우를 제외하고는 3차원 이상의 다차원 배열은 피하는 것이 좋다.

-2차원 배열

2차원 배열은 행과 열을 나타내는 2개의 인덱스를 가진다.

형식: int s[3][5]; (앞이 행, 뒤가 열이다.)

총 15개의 요소를 가지고 있다.

2차원 배열도 1차원과 마찬가지로 선언과 동시에 초기화할 수 있다.

다만 같은 행에 속하는 초기값들을 중괄호 {}로 따로 묶어주어야 한다. 각 초기값들은 쉼표로 구분된다.