C프로그래밍 실습

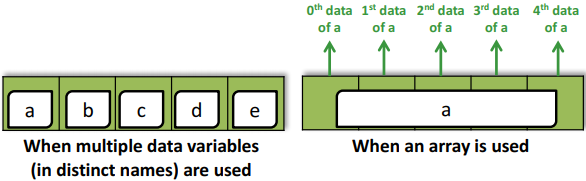
**#8 Array and Character String**

* 배열(Array)

같은 데이터형의 원소들의 집합

둘이상의 수를 처리해야할 때는 인덱스를 사용해서 관련 데이터를 처리한다.

* 각각의 데이터는 정수로 할당된다.



* Declaration and Initialization

+ Declaration format

* Data type Name [# of data(=N)]; N은 배열의 크기를 의미한다

int arr[10];

float farr[20];

+ Initialization format

* Data type Name [#of data(=N)] ={item0, item1, …, item(n-1)};

int arr[10] = {100,200,300,400,500,600,700,800,900,1000};

initialization을 하지 않을 경우 쓰레기 값이 출력된다.

+ Declaration후에 initialize는 허용되지 않는다.

ex) int arr[10];

arr = {100,200,300,400,500,600,700,800,900,1000}; (X)

+ 그러나 declaration후에 각각의 인덱스에 대해서 intialization은 가능하다.

ex) int arr[10];

arr[0]=100; arr[1]=200; arr[2]=300;

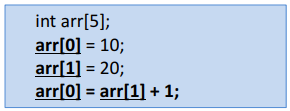
+ 배열의 사이즈보다 적거나 많게 초기화를 할 경우

- 작을 경우 = 문제없다.

int arr[10] = {1,2,3,4,5};

- 클 경우 = 컴파일러 상에서는 문제가 없지만 당신의 프로그램에 문제가 생긴다.

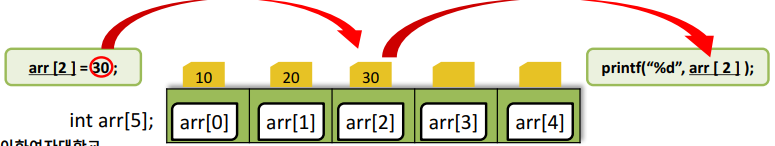
int arr[5] = {1,2,3,4,5,6,7,8,9,10}; 이 초기화는 다른 변수를 수정할 수 있다.

* Usage
* Name [index(=i) (integer)]

index란?

+ 배열의 i번째 요소

+ index는 0에서 시작해서 1씩 증가한다. (index의 범위: 0~(size of array - 1))



* Array Bounds

+ 반드시 유효한 범위 내에서 배열 인덱스를 써야한다.

Valid bounds: 0 ~ size of array -1

+ 만약 인덱스의 범위를 넘어가서 사용했을 경우에는: 문법에러가 나타나지 않는다.

그것이 실행하는 것처럼 보일 수 있다.

이상한 행동을 하는 것처럼 보일 수 있다.

충돌하거나 중단 될 수 있다.

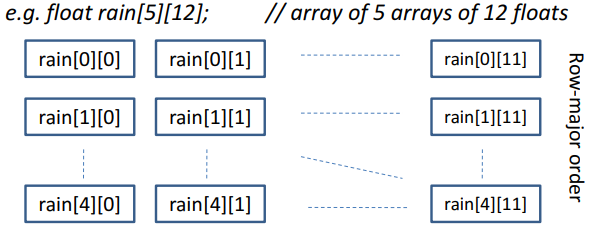
* Multidimensional Arrays

+ 하나의 요소는 multiple index로 식별 된다.

e.g. 5년동안의 매달 비가 온 날의 데이터 분석을 위해서 5x12 데이터 요소가 필요하다.

+ 나타내는 방법: 다차원 배열

배열안의 배열을 사용한다.

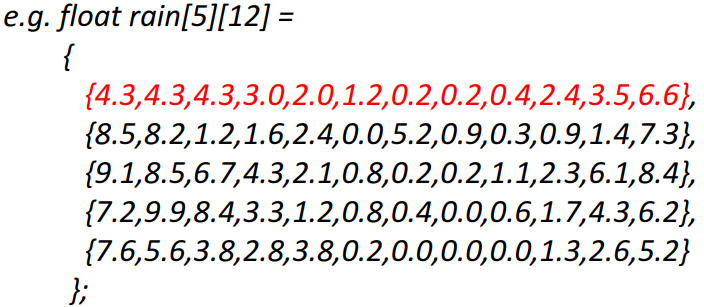


* Initializing a Two-Dimensional Array

행(row) 우선 배열을 한다.

{ }안에 각각의 행(row)를 채운다.

행을 () 사이의 배열요소로 만든다.



* 문자열 (Character Strings)

문자(character): 한 문자, 한 자리, 한 글자 ex) ‘A’ (작은 따옴표)

문자열(string): 하나이상의 문자들의 집합 (배열) ex) “Hello, C!” (큰 따옴표)

* Declaration of String Data Type
* char name[n+1];

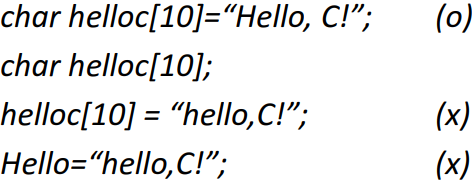
n은 문자열의 최대 길이 +1을 해야 한다.

문자열을 배열로 한글자씩 담는다.

* Initialization of String Data Type

+ 큰 따옴표를 이용한다.

+ 문자열은 오직 declaration과 initialized가 동시에 될 때만 “=”로 initialized 될 수 있다.



* Type char Array and the Null Character

+ string data type은 “문자들의 배열”이다.

배열은 동일한 데이터 형을 가진 하나이상의 데이터로 구성된다.

인접한 (근처의) 메모리 공간에 저장된다.

+ 정의에 의하면 모든 string 데이터는 반드시 마지막에 ‘\0’, null character으로 끝난다.

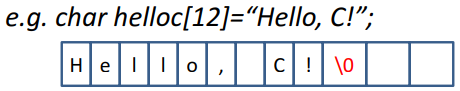
이것이 배열의 크기에 +1을 해야 하는 이유이다.

* char name[the maximum number of characters +1];



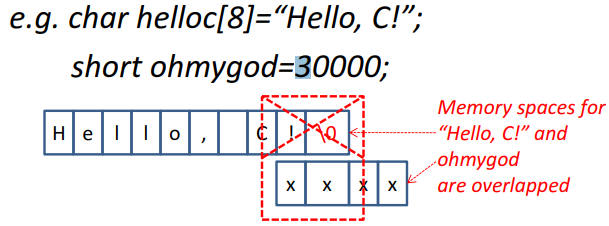
+ 문자열 데이터가 선언 된 것 보다 짧은 문자열로 초기화 되었을 때

남은 메모리 공간은 사용되지 않는다.

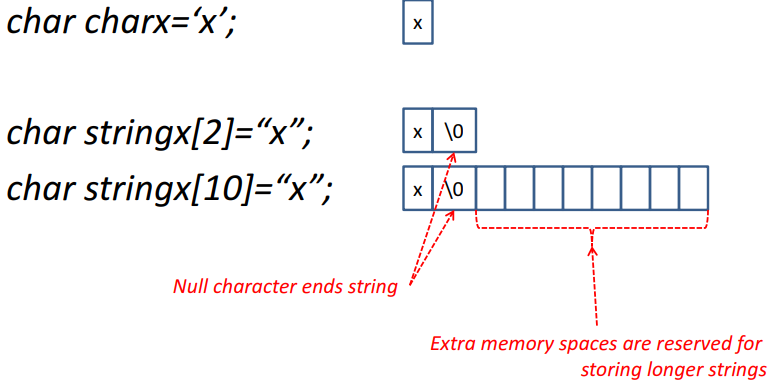


+ 문자열 데이터가 선언 된 것 보다 긴 문자열로 초기화 되었을 때

주변 데이터 혹은 자체가 파괴된다. (overlapped)

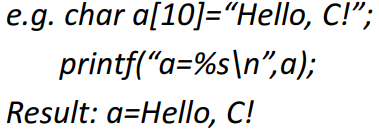


* String VS Characters



* printf() 함수

%s: 문자열



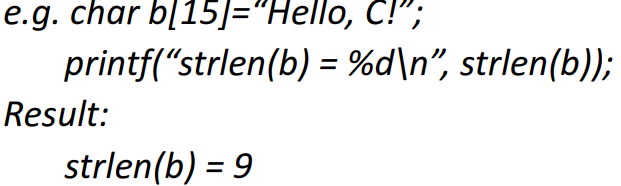
* strlen() 함수

strlen(arg)

arg에 저장 되어있는 문자의 숫자를 도출한다.

arg: ‘\0’으로 끝나느 문자열

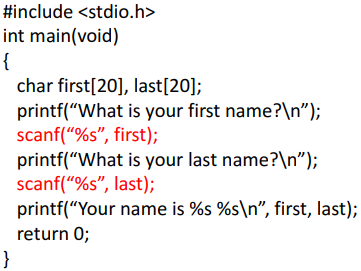
‘\0’은 문자로 카운팅 되지 않는다.



* scanf() 함수

변수의 이름을 scanf()안에 넣는다.

배열의 경우 &를 사용하지 않아도 된다. (or char의 배열인 문자열)

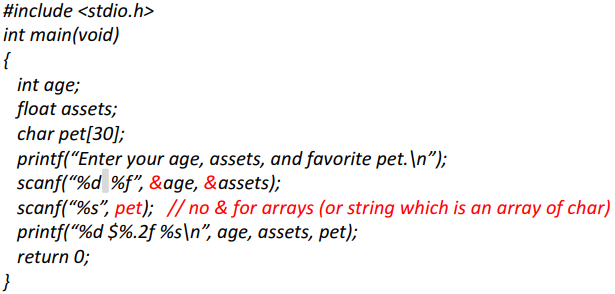


* Using scanf() for Multiple inputs

+ scanf()를 이용해서 하나이상의 입력을 어떻게 받을 까?

scanf()를 각기 다른 줄에 사용한다

scanf()안에 여러 개임을 명시한다.



+ scanf() 함수는 whitespaces로 사용자의 입력을 각각의 변수로 나누는 방법을 결정한다.

whitespaces: newline, tabs, spaces -> 이것들은 모두 scanf()에서 똑같다.

* Regular Characters in the format String in scanf()

scanf()를 사용하면 문자들을 형식문자열 (“”사이)에 넣을 수 있다.



컴파일러가 숫자를 입력하고 쉼표를 입력한 다음 두번째 숫자를 입력한다는 것을 알린다.