10\_문자열

학습목표

문자열이 컴퓨터 내부에서 어떻게 표현되는지를 알 수 있다.

문자열 입출력 함수들을 사용할 수 있다.

문자열을 비교하고 복사할 수 있다.

여러 개의 문자열을 저장하고 처리할 수 있다.

1. 문자열

문자열(String)은 “Hello”와 같은 문자들의 모임이다. 문자열은 큰 따옴표를 이용하여 표현된다. 큰 따옴표로 표시하는 문자열은 변경이 불가능하다. 즉 문자열 상수라고 할 수 있다. 문자열은 문자형 배열에 저장할 수 있다,

Char str[6] = “Hello”;

C언어에서는 문자열의 끝이 반드시 NULL문자(\0)라는 특수한 값으로 끝나야 하기 때문에 본래 문자 수보다 1 더 크게 배열의 크기를 만들어야 한다. 문자열은 정수형 변수들과는 달리 정상적인 데이터와 쓰레기 값을 스스로 분리할 수 있는 능력이 없기 때문에 문자열의 끝을 표시하는 NULL문자가 마지막에 반드시 필요한 것이다.

문자형 배열을 초기화하는 두가지 방법

첫 번째 방법은 배열을 초기화 하듯이 각 배열 요소 값들을 중괄호 안에 넣어주는 방법이다. 이 경우 배열의 마지막 요소에는 반드시 NULL문자를 넣어주어야 한다.

두 번째 방법은 문자열 상수를 사용하여 초기화하는 방법으로 편리한 방법이다. 배열의 요소에는 문자열 상수 안의 문자들이 자동으로 저장된다. 이 방법을 사용하면 컴파일러가 자동으로 문자열의 끝에 NULL문자를 추가한다. 만약 배열의 크기를 지정하지 않으면 컴파일러가 문자의 개수에 맞추어 자동으로 배열의 크기를 설정한다. 이 방법은 문자의 개수를 셀 필요가 없어서 편리하다.

1. 문자 입출력

사용자로부터 문자를 입력 받으려면 다음과 같은 함수를 사용할 수 있다.

getchar() 하나의 문자를 받아서 반환한다. 버퍼를 사용한다.

putchar() 하나의 문자를 받아서 출력한다.

getch() 하나의 문자를 받아서 반환한다. 버퍼를 사용하지 않는다.

putch() 하나의 문자를 받아서 출력한다.

Getchar()는 사용자가 엔터키를 누르기 전까지 글자들을 프로그램으로 보내지 않는다. Getchar()는 버퍼를 사용하기 때문이다.

1. 문자열 입출력

문자열 입력하거나 출력하려면 다음과 같은 함수들을 사용할 수 있다.

scanf(“%s”, s); 문자열을 읽어서 배열s[]에 저장

printf(“%s”, s); 배열 s[]에 저장되어 있는 문자열을 출력한다.

gets\_s(char s[], int length); 한 줄의 문자열을 읽어서 배열s[]에 저장한다.

puts(char, s[]); 배열 s[]에 저장되어 있는 한 줄의 문자열을 출력한다.

Gets\_s(char buffer[], sizeof buffer); //gets\_s의 사용법

1. 문자열 처리

문자열을 사용하다보면 두 개의 문자열을 붙이는 작업이나 두 개의 문자열을 서로 비교하는 작업들이 필요해진다. 문자열 함수들은 string.h에 선언되어 있다.

Strlen(s) 문자열 s의 길이를 구한다.

Strcpy(s1, s2) s2를 s1에 복사한다.

Strcat(s1, s2) s2를 s1의 끝에 붙여넣는다.

Strcmp(s1, s2) s1과 s2를 비교한다. 사전적인 순서에서 어떤 문자열이 앞에 있는지를 알려준다. S1이 s2앞에 있으면 음수가 반환되고 s1과 s2가 같으면 0 s1이 s2의 뒤에 있으면 양수가 반환된다.

문자열을 수치로 변환

“1.23456”과 1.23456은 어떤 차이가 있을까? 앞은 문자열이고 뒤는 실숫값이다. 컴퓨터 안에서는 완전히 다르게 취급한다. 문자열을 정수나 실수로 변환하려면 어떻게 해야 할까? 문자열을 정숫값으로 변환하려면 Atoi()함수를 사용하면 된다. 문자열을 실숫값으로 변환하려면 atof()을 사용하면 된다. 헤더 파일 stdlib.h를 포함해야 쓸 수 있다.

1. 문자열과 포인터

문자열을 저장할 때, 이제까지는 문자형 배열만을 사용하였다. 하지만 하나의 방법이 더 있다. 문자 포인터를 선언하고 여기에 문자열의 주소를 저장하는 방법이다.

1. 문자열의 배열

문자열이 여러 개 필요한 경우에는 어떤 구조를 사용하여 문자열들을 저장하면 좋을까? 여러 개의 1차원 문자형 배열을 사용하여도 되지만 2차원 문자형 배열을 사용하는 것이 여러모로 간편하다.