11. 문자와 문자열

7주차

문자 포인터 배열의 2종류

◆ 문자 포인터 배열이란
- 여러 개의 문자열 하나의 배열로 표현하는 방법

◆ 문자열 상수들의 주소 배열

```
char *pa[] = { "JAVA", "C#", "C++" };
```

- 문자열 저장에 필요한 최소한의 공간만 사용
- pa 변수로써 문자열 내용 변경 불가능. (실행 오류)

◆ 문자의 2차원 배열을 상수문자열로서 초기화

```
char ca[][5] = { "JAVA", "C#", "C++" };
```

- 2차원 배열 구성을 위해 0으로 채워지는 공간이 생김
- ca 변수로써 문자열 내용 변경 가능.

다양한 문자열 관련 함수 (strarray.c)

```
#include <stdio.h>

int main(void) {
    char *pa[] = { "JAVA", "C#", "C++" };
    char ca[][5] = { "JAVA", "C#", "C++" };

    //각각의 3개 문자열 출력
    //pa[0][2] = 'v';        //실행 오류 발생
    //ca[0][2] = 'v';        //수정 가능
    printf("%s ", pa[0]); printf("%s ", pa[1]); printf("%s₩n", pa[2]);
    printf("%s ", ca[0]); printf("%s ", ca[1]); printf("%s₩n", ca[2]);

    //문자 출력
    printf("%c %c %c₩n", pa[0][1], pa[1][1], pa[2][1]);
    printf("%c %c %c₩n", ca[0][1], ca[1][1], ca[2][1]);

    return 0;
}
```

[결과]
JAVA C# C++
JAVA C# C++
A # +
A # +

모두 소문자로 변환

명령행 인자

- DOS 창에서 > dir /w
  - 프로그램 dir를 개발한다면 옵션에 해당하는 "/w" 를 어떻게 인식할까?
    -> 명령행 인자(command line arguments)
- main(int argc, char *argv[])
  - main 함수에서 두 개의 인자 argc와 argv를 통해 받을 수 있음.
  - argc : 명령행에서 입력한 문자열의 수
  - argv : 명령행에서 입력한 문자열들. 문자 포인터 배열.
  - 위의 경우,
    argv=2, argc[0]=프로그램경로, argv[1]= "/w"
- Visual C++에서 명령행 인자 설정하기
  - 프로젝트명에서 right click -> "속성"
  - [디버깅] - [명령 인수] 의 입력 상자에 인자를 기술
  - 위의 경우,
    /w

다양한 문자열 관련 함수 (commandarg.c)

```
#include <stdio.h>

int main(int argc, char *argv[]) {
    int i = 0;

    printf("실행 명령행 인자(command line arguments) >>₩n");
    printf("argc = %d₩n", argc);
    for (i = 0; i < argc; i++)
        printf("argv[%d] = %s₩n", i, argv[i]);

    return 0;
}
```

[결과]
실행 명령행 인자(command line arguments) >>
argc = 1
argv[0] = C:₩jychoi₩부천대₩C언어_2018₩TestProject₩x64₩Debug₩TestProject.exe

프로그래밍 연습 10

```c
#define _CRT_SECURE_NO_WARNINGS
#include <string.h>
#include <stdio.h>
int toint(char *anum) {
    int num = 0;
    while (*anum != 0) {
        num = num * 10 + (*anum - '0');
        anum++;
    }
    return num;
}
int main() {
    char str[20];
    printf("정수 입력 : ");
    scanf("%s", str);
    printf("atoi() 이용 : %d₩n", atoi(str));
        printf("toint() 이용 : %d₩n", toint(str));
    return 0;
}
```

[결과]
정수 입력 : 123
atoi() 이용 : 123
toint() 이용 : 123

프로그래밍 연습 11

```
#define _CRT_SECURE_NO_WARNINGS
#include <string.h>
#include <stdio.h>
int toint(char *anum) {
    .........
}
int main(int argc, char *argv[]) {
    int num1, num2;

    if (argc != 3) {
        printf("Usage : test {num1} {num2}₩n₩n");
        return;
    }
    num1 = toint(argv[1]);
    num2 = toint(argv[2]);
    printf("더한 결과=%d₩n", num1 + num2);

    return 0;
}
```

[결과]
test.exe 10 20
더한 결과=30