

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 6강 - 반복문

학습 목표

반복문

- 1) 반복문을 이용해 특정 조건이 부합할 때 특정한 명령어를 반복해봅니다.
- 2) 구구단 등의 예제를 작성해보며 반복문의 내부적 원리를 이해합니다.

반복문

FOR문

- 1) FOR문 내부의 조건에 부합하면 계속해서 특정한 구문을 실행합니다.
- 2) 반복문을 탈출하고자 하는 위치에 break 구문을 삽입합니다.

```
for (초기화; 조건; 반복 끝 명령어) {  
    // 반복적으로 실행할 부분  
}
```

반복문

1부터 100까지의 정수 출력하기

```
#include <stdio.h>

int main(void) {
    for (int i = 0; i <= 100; i++) {
        printf("%d\n", i);
    }
    system("pause");
}
```

반복문

1부터 N까지의 합 출력하기

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void) {
    int n, sum = 0;
    printf("n을 입력하세요. ");
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        sum = sum + i;
    }
    printf("%d\n", sum);
    system("pause");
}
```

반복문

무한 루프

- 1) 무한 루프(Infinite Loop)란 종료 조건 없이 한없이 반복되는 반복문을 의미합니다.
- 2) 일부러 무한 루프를 발생시키는 경우도 있지만 일반적인 경우 개발자의 실수로 인해 발생합니다.

```
for (초기화; 조건; 반복 끝 명령어) {  
    // 조건이 항상 참(True)인 경우 무한 루프 발생  
}
```

반복문

무한 루프 예제 ①

```
#include <stdio.h>

int main(void) {
    for (;;) {
        // 조건문의 내용이 없으면 항상 참(True)
        printf("Hello World!\n");
    }
    system("pause");
}
```

반복문

무한 루프 예제 ②

```
#include <stdio.h>

int main(void) {
    for (int i = 0; i <= 100; i--) {
        printf("Hello World!\n");
    }
    system("pause");
}
```


반복문

-1이 입력될 때까지 더하기

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void) {
    int sum = 0;
    for (; 1;) {
        int x;
        scanf("%d", &x);
        if (x == -1) break;
        sum += x;
    }
    printf("%d\n", sum);
    system("pause");
}
```

반복문

WHILE문

- 1) WHILE문의 조건에 부합하면 계속해서 특정한 구문을 실행합니다.
- 2) 반복문을 탈출하고자 하는 위치에 break 구문을 삽입합니다.

```
while (조건) {  
    // 반복적으로 실행할 부분  
}
```

반복문

특정 문자를 N번 출력하기

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void) {
    int n;
    char a;
    scanf("%d %c", &n, &a);
    while (n--) {
        printf("%c ", a);
    }
    system("pause");
    return 0;
}
```

반복문

특정 숫자의 구구단 출력하기

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main(void) {
    int n;
    scanf("%d", &n);
    int i = 1;
    while (i <= 9) {
        printf("%d * %d = %d\n", n, i, n * i);
        i++;
    }
    system("pause");
}
```

반복문

중첩된 반복문

- 1) 중첩된 반복문이란 반복문 내부에 다른 반복문이 존재하는 형태의 반복문입니다.
- 2) 반복문이 중첩될수록 연산 횟수는 제곱 형태로 늘어납니다.

반복문

구구단 출력하기 (WHILE문)

```
#include <stdio.h>

int main(void) {
    int i = 1;
    while (i <= 9) {
        int j = 1;
        while (j <= 9) {
            printf("%d * %d = %d\n", i, j, i * j);
            j++;
        }
        printf("\n");
        i++;
    }
    system("pause");
}
```

반복문

구구단 출력하기 (FOR문)

```
#include <stdio.h>

int main(void) {
    for (int i = 1; i <= 9; i++) {
        for (int j = 1; j <= 9; j++) {
            printf("%d * %d = %d\n", i, j, i * j);
        }
        printf("\n");
    }
    system("pause");
}
```

반복문

FOR문과 WHILE문의 관계

- 1) 모든 FOR문은 WHILE문으로 변경할 수 있으며 모든 WHILE은 FOR문으로 변경할 수 있습니다.
- 2) C언어 소스코드가 최적화 되면서 어셈블리어 단에서는 동일한 명령어로 동작하기 때문입니다.

배운 내용 정리하기

반복문

- 1) 반복문의 종류로는 FOR문과 WHILE문이 있으며 상호 치환이 가능합니다.
- 2) 의도하지 않은 무한 반복이 발생하지 않도록 유의해야 합니다.
- 3) 행렬, 그래프 등의 연산에 있어서 이중 반복문이 매우 많이 활용됩니다.