loop statement

loop statement

FOR문

1) FOR문 내부의 조건에 부합하면 계속해서 특정한 구문을 실행함.
2) 반복문을 탈출하고자 하는 위치에 break 구문을 삽입함.

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

loop statement

1부터 100까지의 정수 출력하기

```c
#include <stdio.h>

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

loop statement

1부터 N까지의 합 출력하기

```c
#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);
}
```

loop statement

무한 루프

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

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

loop statement

무한 루프 예제 ①

```c
#include <stdio.h>

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

loop statement

무한 루프 예제 ②

```c
#include <stdio.h>

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

}
```

loop statement

-1이 입력될 때까지 더하기

```c
#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);
}
```

loop statement

WHILE문

1) WHILE문의 조건에 부합하면 계속해서 특정한 구문을 실행함.
2) 반복문을 탈출하고자 하는 위치에 break 구문을 삽입함.

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

loop statement

특정 문자를 N번 출력하기

```c
#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);
  }
  return 0;
}
```

loop statement

특정 숫자의 구구단 출력하기

```c
#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++;
  }
}
```

loop statement

중첩된 반복문

1) 중첩된 반복문이란 반복문 내부에 다른 반복문이 존재하는 형태의 반복문임.
2) 반복문이 중첩될수록 연산 횟수는 제곱 형태로 늘어남.

loop statement

구구단 출력하기 (WHILE문)

```c
#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++;
  }
}
```

loop statement

구구단 출력하기 (FOR문)

```c
#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");
  }
}
```

loop statement

FOR문과 WHILE문의 관계

1) 모든 FOR문은 WHLE문으로 변경할 수 있으며 모든 WHILE은 FOR문으로 변경할 수 있음.
2) C언어 소스코드가 최적화 되면서 어셈블리어 단에서는 동일한 명령어로 동작하기 때문임.