

C/C++

structure



pointer



function



array[]



switch/case

for, while

프로그래밍 기초



malloc/free



if else

Lab 6: 함수

scanf() 호출 후 new line('\n') 제거

- 숫자와 문자를 각각 scanf() 호출로 입력 받음
 - 첫 번째 scanf("%d", &num)는 정상 동작함
 - 두 번째 scanf("%c", &ch)는 정상 동작 하지 않음
 - 숫자 입력 후 Enter키를 누르면 개행문자('\n')가 입력됨

실행 결과

Input a number: 5

typed number: 5

Type y or n: typed char:

'y'나 'n' 입력 안됨

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

```
int main()
{
```

```
    int num;
    char ch;
```

```
    printf("Input a number: ");
    scanf("%d", &num); // 첫 번째 scanf()
    printf("typed number: %d\n", num);
```

```
    printf("Type y or n: ");
    scanf("%c", &ch); // 두 번째 scanf() 입력 안됨
                      // 이전에 입력된 Enter키를 읽어옴
```

```
    printf("typed char: %c\n", ch);
    return 0;
```


```
}
```

- 첫 번째 scanf()에서 숫자와 Enter키 입력
- 두 번째 scanf()에서 ch에는 '\n'이 저장되어 있음

```
scanf_newline.c x
Lab > lab06_chap09 > scanf_newline.c > main()

8  int main()
9  {
10     int num;
11     char ch;
12
13     printf("Input a number: ");
14
15     scanf("%d", &num); // 첫 번째 scanf()
16     printf("typed number: %d\n", num);
17
18     printf("Type y or n: ");
19     scanf("%c", &ch); // 두 번째 scanf() 입력 안됨
20                       // 이전에 입력된 Enter키를 읽어옴
21     printf("typed char: %c\n", ch);
22     return 0;
23 }
```

scanf() 호출 이후 개행 문자('\n') 제거 방법

- 1) scanf(“%d”, &n); 호출 이후
 - `char ch = getchar();` 호출로 ‘\n’을 읽고 버림
 - 권장 방법
- 2) scanf(“%d%c”, &n, &ch); 호출
 - ch에 ‘\n’ 문자 저장되며 ch는 사용하지 않음
- 3) scanf(“%c”, &ch);
 - “ %c” 앞에 공백 추가
 - Enter(‘\n’) 키가 공백으로 인식

scanf() 호출 후 new line('\n') 제거

방법 #1

```
#include <stdio.h>

int main()
{
    int num;
    char ch;

    printf("Input a number: ");
    scanf("%d", &num);
    printf("typed number: %d\n", num);

    // 방법 1
    ch = getchar();

    printf("Type y or n: ");
    scanf("%c", &ch);
    printf("typed char: %c\n", ch);
    return 0;
}
```

- getchar() 호출로 버퍼에 남아 있는 Enter키를 읽음

방법 #2

```
#include <stdio.h>

int main()
{
    int num;
    char ch;

    printf("Input a number: ");
    scanf("%d%c", &num, &ch); // 방법 2
    printf("typed number: %d\n", num);

    printf("Type y or n: ");
    scanf("%c", &ch);
    printf("typed char: %c\n", ch);
    return 0;
}
```

- 정수 및 문자(Enter키)를 입력 받고 처리하지 않음

```
Input a number: 5
number: 5
Type y or n: y
typed char: y
```

scanf() 호출 후 new line('\n') 제거

방법 #3

```
#include <stdio.h>

int main()
{
    int num;
    char ch;

    printf("Input a number: ");
    scanf("%d", &num);
    printf("typed number: %d\n", num);

    printf("Type y or n: ");
    scanf(" %c", &ch);
    printf("typed char: %c\n", ch);
    return 0;
}
```

입력 버퍼에 남아 있는 모든 공백
문자(tab, space, Enter등) 무시

```
Input a number: 5
number: 5
Type y or n: y
typed char: y
```

Lab 1. 홀수, 짝수 및 절대값 계산 프로그램

- 하나의 정수를 입력받은 후 홀수, 짝수를 구분하고 해당 정수의 절대값을 출력하는 프로그램을 작성하시오. (제출 파일: lab06_01.c)
 - `int evenodd(int n)` 함수
 - 주어진 정수가 짝수이면 0, 홀수이면 1을 반환
 - `int absolute(int n)` 함수
 - 주어진 정수의 절대값을 반환
 - 사용자가 'y'나 'Y'를 입력하면 계속 반복됨
- 주의사항
 - `scanf("%d", &n)` 이후 개행문자('\n') 제거

```
Input an integer number: 5
5: Odd number
absolute(5): 5
-----
Again? (y/n): y
Input an integer number: -10
-10: Even number
absolute(-10): 10
-----
Again? (y/n): Y
Input an integer number: -6
-6: Even number
absolute(-6): 6
-----
Again? (y/n): n
Finish
```

Lab 2. 1부터 n까지 합을 구하는 프로그램

- 표준 입력으로 양의 정수(int n)를 입력 받아서 1부터 n까지의 합을 구하는 프로그램을 작성 하시오. (제출 파일: lab06_02.c)
 - main() 함수
 - n의 값이 0보다 작거나 같으면 숫자 n을 다시 입력 받음
 - getsum(n) 함수 호출 및 리턴값을 받아서 총합 출력
 - Sum from 1 to n = 총합 출력
 - 함수 int getsum(int n)
 - 1부터 n까지 합을 구하는 과정을 출력
 - 리턴값: 1부터 n까지의 합
- 실행 결과

```
Input a number: 0
Try Again (0 <= 0)
Input a number: 10
  1 =  0 +  1
  3 =  1 +  2
  6 =  3 +  3
 10 =  6 +  4
 15 = 10 +  5
 21 = 15 +  6
 28 = 21 +  7
 36 = 28 +  8
 45 = 36 +  9
 55 = 45 + 10
Sum from 1 to 10 = 55
```