

C언어 강의자료

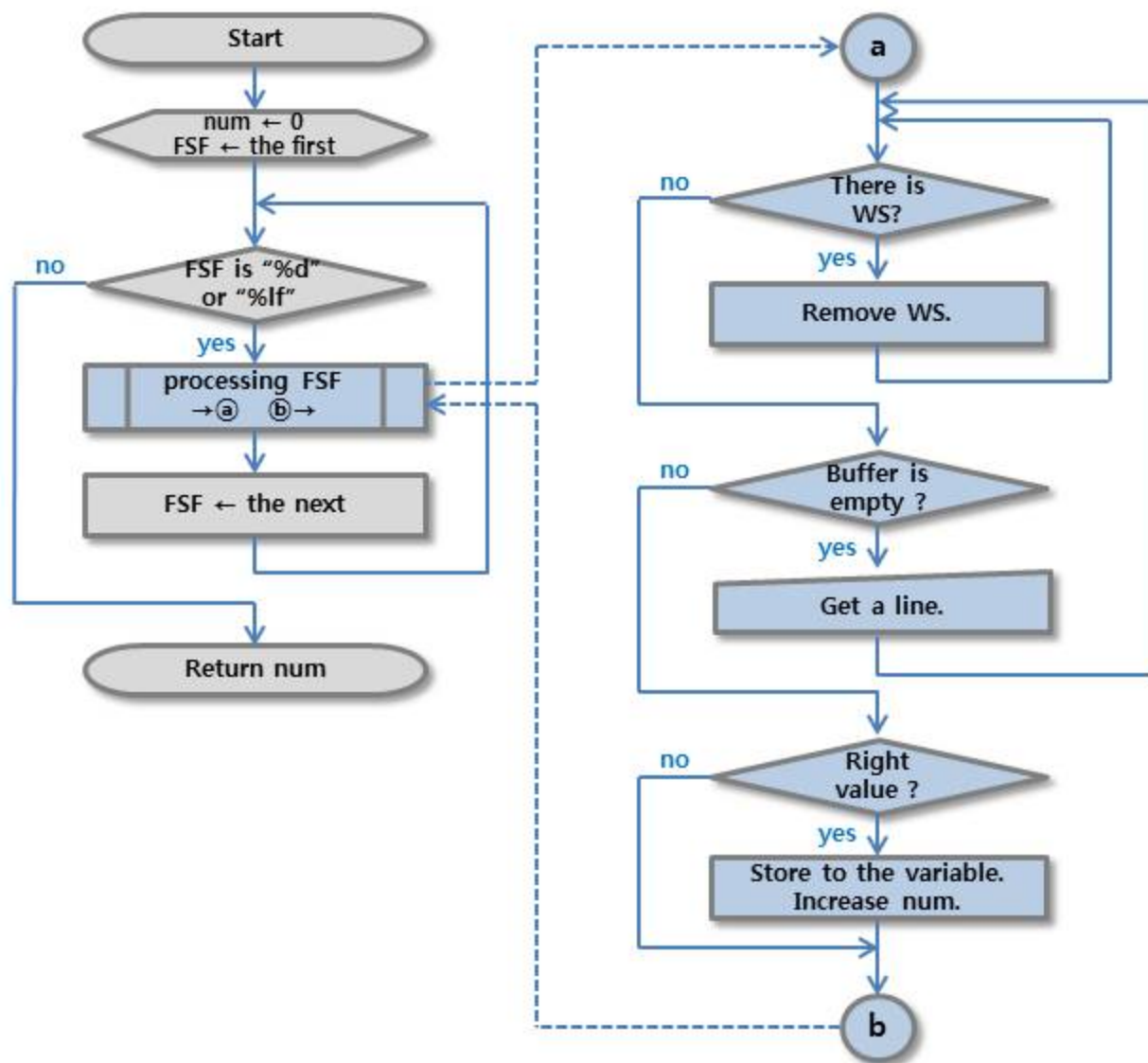
문정욱



프로그래밍 연습 5

표준 입력 방식

scanf 동작 원리



scanf 동작 원리(정수의 입력)

- 올바른 값이 입력된 경우
 - 입력된 값이 변수에 저장
 - 1(one) 반환

입출력 결과

```
35
a == 35
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a = 99;
    int retv;

    retv = scanf("%d", &a);
    printf("a == %d\n", a);
    printf("retv == %d\n", retv);
    return 0;
}
```

scanf 동작 원리(정수의 입력)

- 값 앞에 WS가 있을 경우
 - WS는 무시한 후 값을 저장
 - 1(one) 반환

입출력 결과

```
45
a == 45
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

※ WS: White space, (space, tab, new line)

```
#include <stdio.h>

int main(void)
{
    int a = 99;
    int retv;

    retv = scanf("%d", &a);
    printf("a == %d\n", a);
    printf("retv == %d\n", retv);
    return 0;
}
```

scanf 동작 원리(정수의 입력)

- 잘못된 값이 입력된 경우
 - 변수는 이전 값 유지
 - 0(zero) 반환

입출력 결과

```
abc
a == 99
retv == 0
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a = 99;
    int retv;

    retv = scanf("%d", &a);
    printf("a == %d\n", a);
    printf("retv == %d\n", retv);
    return 0;
}
```

scanf 동작 원리(정수의 입력)

- 부분적 입력이 가능한 경우
 - 부분적 입력을 값으로 인식
 - 1(zero) 반환

입출력 결과

```
123abc
a == 123
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
0abc
a == 0
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
123 456
a == 123
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
123.456
a == 123
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int main(void)
```

```
{
```

```
    int a = 99;
```

```
    int retv;
```

```
    retv = scanf("%d", &a);
```

```
    printf("a == %d\n", a);
```

```
    printf("retv == %d\n", retv);
```

```
    return 0;
```

```
}
```

scanf 동작 원리(실수의 입력)

- 올바른 값이 입력된 경우
 - 입력된 값이 변수에 저장
 - 1(one) 반환

입출력 결과

```
123.456
a == 123.456000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
123
a == 123.000000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
123.
a == 123.000000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
.456
a == 0.456000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
0.
a == 0.000000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
.0
a == 0.000000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int main(void)
```

```
{
```

```
    double a = 99.99;
```

```
    int retv;
```

```
    retv = scanf("%lf", &a);
```

```
    printf("a == %f\n", a);
```

```
    printf("retv == %d\n", retv);
```

```
    return 0;
```

```
}
```


scanf 동작 원리(실수의 입력)

- 값 앞에 WS가 있을 경우
 - WS는 무시한 후 값을 저장
 - 1(one) 반환

입출력 결과

```
123.456
a == 123.456000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

※ WS: White space, (space, tab, new line)

```
#include <stdio.h>

int main(void)
{
    double a = 99.99;
    int retv;

    retv = scanf("%lf", &a);
    printf("a == %f\n", a);
    printf("retv == %d\n", retv);
    return 0;
}
```

scanf 동작 원리(실수의 입력)

■ 잘못된 값이 입력된 경우

- 변수는 이전 값 유지
- 0(zero) 반환

입출력 결과

```
abc
a == 99.990000
retv == 0
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
.abc
a == 99.990000
retv == 0
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
. 123.456
a == 99.990000
retv == 0
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    double a = 99.99;
    int retv;

    retv = scanf("%lf", &a);
    printf("a == %f\n", a);
    printf("retv == %d\n", retv);
    return 0;
}
```

scanf 동작 원리(실수의 입력)

- 부분적 입력이 가능한 경우
 - 부분적 입력을 값으로 인식
 - 1(zero) 반환

입출력 결과

```
123.456abc
a == 123.456000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
123.456 7.8
a == 123.456000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
0.abc
a == 0.000000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
.0abc
a == 0.000000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
0. 7.8
a == 0.000000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
.0 7.8
a == 0.000000
retv == 1
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int main(void)
```

```
{
```

```
    double a = 99.99;
```

```
    int retv;
```

```
    retv = scanf("%lf", &a);
```

```
    printf("a == %f\n", a);
```

```
    printf("retv == %d\n", retv);
```

```
    return 0;
```

```
}
```

scanf 동작 원리(여러 값의 입력)

- 여러 개의 값을 입력할 때 반환 값

입출력 결과

```
11
22 33
a == 11
retv1 == 1
b == 22, c == 33
retv2 == 2
계속하려면 아무 키
```

입출력 결과

```
11 22 33
a == 11
retv1 == 1
b == 22, c == 33
retv2 == 2
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
11
22 abc
a == 11
retv1 == 1
b == 22, c == -85
retv2 == 1
계속하려면 아무 키
```

입출력 결과

```
11 abc def
a == 11
retv1 == 1
b == -858993460, c == -858993460
retv2 == 0
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int main(void)
```

```
{
```

```
    int a, b, c;
```

```
    int retv1, retv2;
```

```
    retv1 = scanf("%d", &a );
```

```
    retv2 = scanf("%d%d", &b, &c );
```

```
    printf("a == %d\n", a );
```

```
    printf("retv1 == %d\n", retv1);
```

```
    printf("b == %d, c == %d\n", b, c );
```

```
    printf("retv2 == %d\n", retv2);
```

```
    return 0;
```

```
}
```

scanf 동작 원리(입력 버퍼의 동작)

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

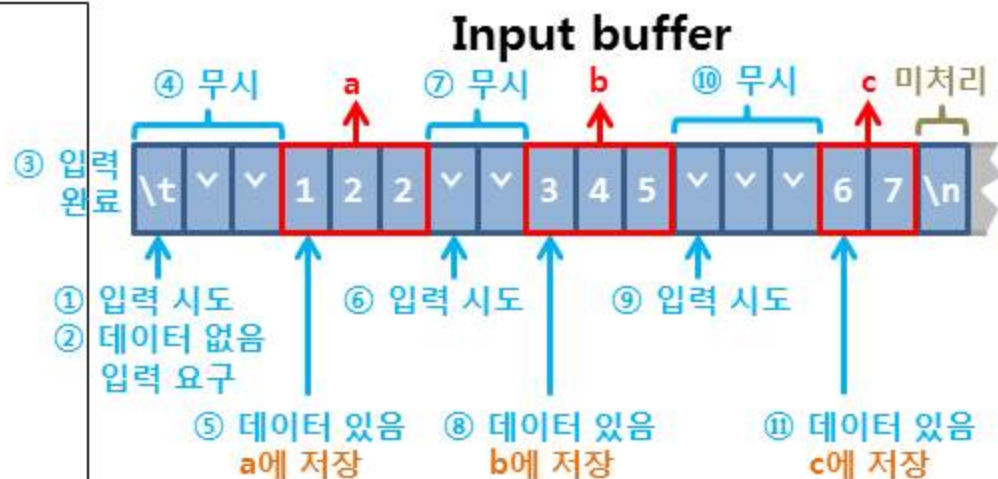
```
int main(void)
{
```

```
    int a, b, c;
```

```
    scanf("%d", &a );
    printf("a==%d\n", a );
```

```
    scanf("%d%d", &b, &c );
    printf("b==%d, c==%d\n ", b, c );
    return 0;
```

```
}
```



scanf 동작 원리(입력 버퍼의 동작)

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

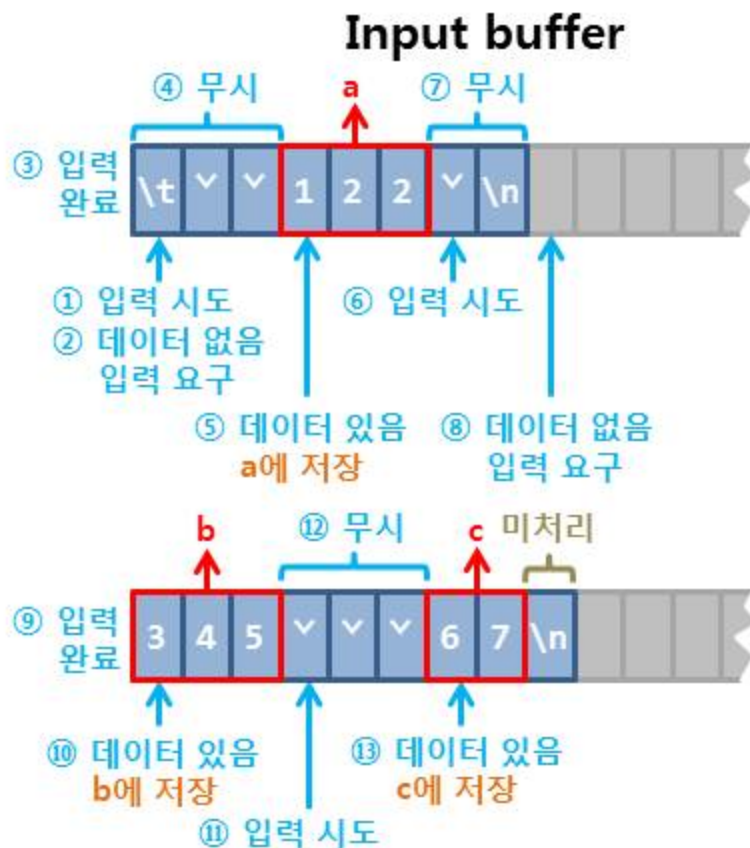
```
int main(void)
{
```

```
    int a, b, c;
```

```
    scanf("%d", &a );
    printf("a==%d\n", a );
```

```
    scanf("%d%d", &b, &c );
    printf("b==%d, c==%d\n ", b, c );
    return 0;
```

```
}
```



scanf 동작 원리(ctrl-z의 입력)

- ctrl-z가 입력된 경우
 - 더 이상 입력할 데이터가 없음을 의미함.
 - EOF(-1, End Of File) 반환
 - 주의) ctrl-z는 -1이 아니다. 입력이 끝났음을 알리는 신호일 뿐이다.

입출력 결과

```
^Z
a == 99
retv == -1
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a = 99;
    int retv;

    retv = scanf("%d", &a);
    printf("a == %d\n", a);
    printf("retv == %d\n", retv);
    return 0;
}
```

scanf 동작 원리(ctrl-z의 입력)

- 다른 데이터와 함께 ctrl-z가 입력된 경우
 - 제어문자(SUB, 0x1A)로 인식
 - 정수 입력을 시도할 경우 0(입력 fail 의미) 반환

입출력 결과

```
^Z
a == 99
retv == 0
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a = 99;
    int retv;

    retv = scanf("%d", &a);
    printf("a == %d\n", a);
    printf("retv == %d\n", retv);
    return 0;
}
```


scanf 동작 원리(ctrl-z의 입력)

■ 배열에 수열을 저장 (1)

• 간접 길이 표현 방식

- 수열의 끝을 마침 신호로 표현하여 수열의 길이를 간접적으로 표현하는 방식.
- 일반적으로 수열의 마침 신호는 EOL(End of Line), EOF(End of File), Ending Token, FAIL(reading FAIL) 등을 사용한다.
- 왼쪽 소스코드는 EOF, FAIL을 수열의 마침 신호로 사용한다.



```
#include <stdio.h>

int main(void)
{
    int a[1024];
    int i, n;

    for(i=0;!(scanf("%d", &a[i]))!=1);++i)
        ;
    n = i;

    for(i=0;i<n;++i)
        printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```

배열의 길이

수열의 길이

scanf 동작 원리(ctrl-z의 입력)

■ 배열에 수열을 저장 (2)

• 직접 길이 표현 방식

- 수열의 길이를 수열의 앞에 표현하는 방식.
- 단, 명시된 길이와 실제 길이가 다를 경우 둘 중 최소값을 수열의 길이로 정함.

입출력 결과

5
11 22 33 44 55
11 22 33 44 55
계속하려면 아무 키나 누르십시오 . . .

입출력 결과

3 11 22 33
11 22 33
계속하려면 아무 키나 누르십시오 . . .

입출력 결과

4
11 22 33 44 55
11 22 33 44
계속하려면 아무 키나 누르십시오 . . .

입출력 결과

5
11 22 33
^Z
11 22 33
계속하려면 아무 키나 누르십시오 . . .

실제 수열의 길이가
명시된 수열의 길이보다 작을 때는
실제 수열의 길이를 우선으로 한다.

```
#include <stdio.h>

int main(void)
{
    int a[1024];
    int i, n;

    scanf("%d", &n);
    for(i=0; i<n && !(scanf("%d", &a[i])!=1); ++i)
        ;
    n = i; // because i <= n

    for(i=0; i<n; ++i)
        printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```

scanf 동작 원리(ctrl-z의 입력)

■ 배열에 수열을 저장 (3)

• 고정 길이 표현 방식

- 수열의 길이를 미리 정하여 직간접적 표현이 불필요.
- 입력된 수열의 길이와 정해진 길이가 다를 경우 데이터 오류로 간주.

입출력 결과	
11 22 33 44 55 계속하려면 아무 키	11 22 33 44 ^Z Not enough data
11 22 33 end Wrong data 계속하려면 아무 키	11 22 33 44 ^Z Wrong data 계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

int main(void)
{
    int a[5];
    int sz=5, i, retv;

    for(i=0;i<sz;++i) {
        retv = scanf("%d",&a[i]);
        if(retv == EOF) {
            printf("Not enough data\n");
            return -1;
        }
        if(retv == 0) {
            printf("Wrong data\n");
            return -2;
        }
    }

    for(i=0;i<sz;++i)
        printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```