

Jaewon Choi  
Keunsan Park  
Haeun Lee  
(snucsl.ta@gmail.com)

Systems Software &  
Architecture Lab.

Seoul National University

Spring 2023

# 4190.103A-001: Programming Practice Lab. 2



공지

# 출석

- 지각 / 결석
  - 수업 시작후 15분까지 지각 허용
  - 수업중 2번 랜덤하게 출석 체크 (1번이라도 없을시 결석처리)
- 지정좌석제
  - 3/9(목) 좌석 고정 (~중간고사)
  - 중간고사 이후 변경 추후 공지

main 함수

# main 함수

- 코드를 컴파일 한 뒤 실행파일을 만들어 실행하면 코드는 main 함수부터 실행이 됨.
  - 코드는 실행 범위는 대괄호를 통해 표현됨.
  - `int main(void)` : 함수의 정의
    - `void` : 아무 것도 받지 않음
    - `int` (정수형) 값을 돌려줌.
      - 예제에서는 성공적인 종료를 나타내는 0을 반환함.

```
main.c ×
1  #include <stdio.h>
2
3  int main(void) {
4      printf("hello, world\n");
5      return 0;
6  }
7
```

# 전처리기

# 전처리기

- 작성한 코드를 컴파일하여 컴퓨터가 실행할 수 있는 실행파일 만들 때, 예시에서는 컴퓨터에게 다음과 같은 사실을 알려줌.
  - `stdio.h` 라는 파일에 명시된 함수들을 해당 파일에서 가져다 쓸 것이다.
  - 따라서 해당 파일에서 정의되지 않은 함수를 사용하더라도, 해당 함수의 정의가 `stdio.h`에 있다면 컴파일 에러를 생성하지 말 것!
  - 지난 실습 시간 사용했던 출력을 위한 `printf` 함수, `scanf` 함수 등을 불러 사용할 수 있음.

```
main.c ×  
1  #include <stdio.h>  
2  
3  int main(void) {  
4      printf("hello, world\n");  
5      return 0;  
6  }  
7
```

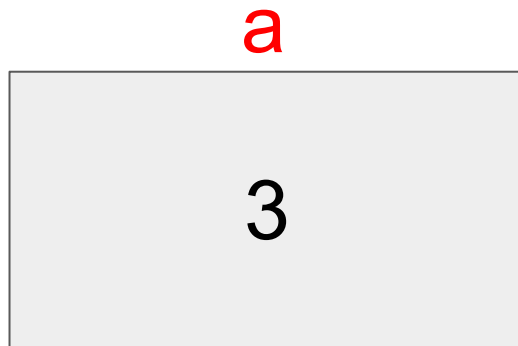
변수



# 변수

- 은행에 스마트폰을 보관할 수 없는 것처럼(?) 변수도 자료형에 따라 적절한 저장공간을 할당 받아야 내용물을 저장할 수 있음.
  - 예시에서는 4 byte 정수를 표현하는 int 자료형의 저장 공간 a를 만들고, 이후에 a에 3이라는 값을 할당함.

```
1 #include <stdio.h>
2
3 int main(void){
4     int a;
5     a = 3;
6
7     return 0;
8 }
9
```



# 변수

- C 언어는 별도로 자료형을 지정하지 않는 python과 다르게 저장하고자 하는 변수에 걸맞는 저장 공간을 만들어주어야 함.
- C 언어에서 기본적으로 제공하는 자료형으로는 아래 그림과 같은 것들이 있음.
  - 자료형은 기본적으로 제공되는 것이기 때문에, 별도의 전처리를 통해 자료형의 사용을 명시할 필요가 없음.

자료형(data type)	할당되는 메모리 크기	표현 가능한 데이터의 범위	
정수형	char	1 바이트	-128 ~ +127
	short	2 바이트	-32768 ~ +32767
	int	4 바이트	-2147483648 ~ +2147483647
	long	4 바이트	-2147483648 ~ +2147483647
실수형	float	4 바이트	$3.4 \times 10^{-37} \sim 3.4 \times 10^{38}$
	double	8 바이트	$1.7 \times 10^{-307} \sim 1.7 \times 10^{308}$
	long double	8 바이트 혹은 그 이상	차이를 많이 보임

그림 출처 : <https://slidesplayer.org/slide/17939551/> 10

`printf / scanf`

# printf / scanf

- stdio.h
  - Standard Input/Output Library Header File
  - Header file
    - 함수의 정의(이름은 무엇이고, 어떤 값을 인자로 받으며, 어떤 값을 돌려주는지와 같은 함수에 대한 설명)등을 모아놓은 곳
  - 해당 header file을 통해 printf와 scanf를 사용할 수 있음.

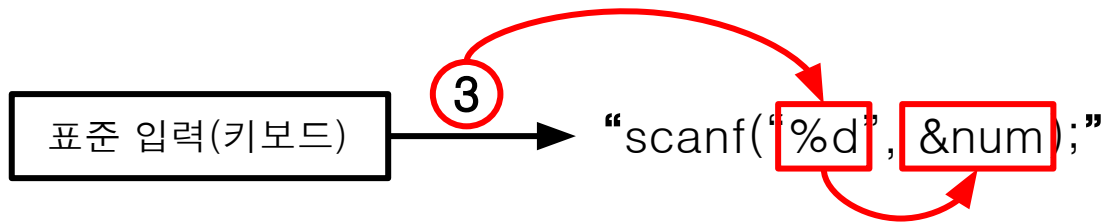
# printf / scanf

- printf
  - 따옴표 안에 명시된 문자열을 화면에 출력할 때 사용
  - \n : 개행 문자
  - 예시에서는 hello, world를 출력하고 다음 줄을 출력할 때 줄 변경을 할 것을 나타냄.

```
main.c ×  
1  #include <stdio.h>  
2  
3  int main(void) {  
4      printf("hello, world\n");  
5      return 0;  
6  }  
7
```

# printf / scanf

- scanf
  - 사용자로부터 입력을 받을 때 사용되는 함수
  - 아래 예시에서는 3이라는 값을 키보드로 치면, 이가 num이라는 변수에 저장됨.
    - & (ampersand)는 주소를 나타냄.
      - &num : num 변수의 주소를 scanf 함수에 전달
      - scanf 함수는 전달된 num 변수의 주소를 통해 num 변수에 키보드로 입력된 3을 저장함.



# printf / scanf

- 왼쪽 아래 예시는 변수 a 값인 10을 출력하는게 아닌, 문자 a를 출력하는 프로그램이다.
- 변수 a의 값을 출력하기 위해서는 우측 아래와 같은 방식으로 출력해야 한다.
  - 정수형 변수 a를 출력하기 위해서는 %d를 사용한다.

```
1  #include <stdio.h>
2
3  int main(void){
4      int a;
5      a = 10;
6
7      printf("a\n");
8
9      return 0;
10 }
11
```

printf("%d", a);

# printf / scanf

- 다음표 안의 출력 문자열에서 변수 값을 출력하고 싶다면 변수의 자료형에 따라 아래 표와 같은 출력 형식을 써야 함.
- 입력의 경우도 마찬가지로, 채우고자 하는 변수의 자료형에 따라 입력 형식이 달라짐.

%c	문자 출력
%s	문자열 출력
%d	10진수 출력
%o	8진수 출력
%x	16진수 출력
%f	실수 출력

```
1 #include <stdio.h>
2
3 int main(void){
4     int a;
5     scanf("%d", &a);
6     printf("%d\n", a);
7
8     return 0;
9
10
11 }
12
```

Input

Output

```
➤ make -s
➤ ./main
10
10
➤
```



# 조건문

# 조건문

- If 문
  - 조건에 해당하면 If 문 내부를 실행.
  - 조건에 해당하지 않으면 무시.

조건: num이 0보다 크니?

```
int num = 3;  
if(num > 0){  
    printf("Positive\n");  
}
```

조건에 맞았을 때 수행하는 동작

# 조건문

- If-Else 문
  - 조건에 해당하면 If 문을, 해당하지 않으면 Else 문을 실행.

조건: num이 0보다 크니?

```
int num = -3;
if(num > 0){
    printf("Positive\n");
} else {
    printf("Negative\n");
}
```

조건에 해당할 때 수행하는 동작

조건에 해당하지 않을 때 수행하는 동작

대입

# 대입

- 변수에 계산한 값을 대입할 수 있음.
  - 연산 우선순위를 정하기 위해서는 괄호를 사용.
    - 연산자 우선순위가 따로 정해져 있지만, 편의를 위해 괄호 사용 권장.
  - = 우측 식을 계산하여 좌측 변수에 대입.
    - 우측 식에는 변수와 숫자를 사용할 수 있음.

```
main.c ×
1 #include <stdio.h>
2
3 v int main(void){
4     int num;
5
6     num = 3 + 5;
7     printf("%d\n", num);
8
9     return 0;
10 }
11
```

```
main.c ×
1 #include <stdio.h>
2
3 v int main(void){
4     int result;
5     int num1 = 3, num2 = 5;
6
7     result = num1 + num2;
8     printf("%d\n", result);
9
10    return 0;
11 }
12
```

# 랜덤 함수

# 랜덤 함수

- rand() 함수가 랜덤 숫자를 반환
  - 하지만 컴퓨터는 알고리즘에 의해 랜덤 숫자를 생성하기 때문에 사실상 랜덤하지 않음!!!
  - 조금 더 랜덤하게 만들고자 시간을 사용하여 랜덤 숫자 생성
    - time 함수를 srand 함수 내부에 작성함으로써 랜덤 숫자를 생성하는 기준을 시간으로 정함.
    - rand() 함수는 stdlib.h에, time 함수는 time.h에서 소환
- 그림에서는 % (모듈로, 나머지 연산)을 통해 10으로 나눈 나머지를 구함.

## 1. 랜덤 숫자 생성

rand\_num = rand() % 10;

## 2. 10으로 나눈 나머지

```
main.c ×
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(void){
6     int rand_num;
7
8     srand(time(NULL));
9
10    rand_num = rand();
11    printf("%d\n", rand_num);
12
13    return 0;
14 }
15
```

# 실습 & 과제



# 과제 / 실습

- 과제 내용
  - 당일 진행한 실습과 관련된 문제
- 제출 기한
  - 실습 : 당일 자정까지, 다음날 자정까지 지각제출 허용 (단, 점수 -30%)
  - 과제 : 다음 실습 수업 전날 23:59 까지
- 제출 방법
  - Elice의 Submit 기능 활용

# 과제

- 평가 방식
  - 매주 과제를 출제하면서 문제 마다 채점 기준을 설명
- 주의점
  - Resubmit 가능
  - Resubmit 시에 마지막 제출 시각을 제출 시점으로 판단