

Euidong Lee
Junseok Lee
Minhyo Jeong
(snucsl.ta@gmail.com)

Systems Software &
Architecture Lab.

Seoul National University

Spring 2022

4190.103A-001: Programming Practice Lab. 2



샘플 코드 설명

샘플 코드 설명

- 전처리기(Preprocessor)
 - Input / Output 을 위한 라이브러리를 추가
 - std : standard, io: input/output, .h: header file

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

샘플 코드 설명

- main 함수
 - 작성한 프로그램이 실행될 때 처음 불리는 함수
 - 중괄호('{', '}')로 함수가 수행할 내용의 범위를 표현

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

샘플 코드 설명

- printf
 - 화면에 큰따옴표(“, ”)로 둘러 쌓인 문자를 출력
 - \n은 개행문자(엔터)를 의미

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

샘플 코드 설명

- return
 - 함수가 종료될 때 만들어내는 결과물
 - 여기에서는 프로그램이 정상적으로 종료되었다는 의미를 전달

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

변수 – Quick Overview

변수 - Quick Overview

- 정보를 저장하는 역할을 수행

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


변수 – Quick Overview

- 선언(Declare)
 - 변수(Variable)를 생성
 - int: 자료형(Data Type)
 - 실수: float
 - 문자: char
 - short, long, double ...
 - a: 변수 이름

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

변수 - Quick Overview

- 할당(Assignment)
 - 변수에 정보를 저장

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

printf / scanf

printf / scanf

- stdio.h(Standard Input/Output Library)에 구현
- printf
 - 문자열을 화면에 출력(Output)
- scanf
 - 사용자로부터 입력을 받음(Input)

실습 1

실습 1 - 설명

- 변수와 printf를 연습해봅시다
 - 정수를 저장하고 출력
 - 실수를 저장하고 출력
 - 문자를 저장하고 출력

실습 1 - 설명

- 정수를 저장하고 출력하는 프로그램
 - 원하는 10이 나오나요?
 - 아니면 a가 그대로 나오나요?

```
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
```

실습 1 - 설명

- 변수에 저장된 값을 출력하려면 별도의 방법이 필요합니다
 - ‘%’(서식지정자)를 사용해봅시다
 - ‘%’는 변수의 값이 표현될 위치를 지정해줍니다

```
int num = 3;
```

```
printf("%d", num);
```

“3” → 표준 출력(화면)

- 이전의 코드를 수정해보세요

실습 1 - 설명

- 원하는 datatype에 맞는 서식지정자를 사용해야합니다
 - %d : 정수
 - %f : 실수
 - %c : 문자

```
main.c x
1  #include <stdio.h>
2
3  int main(void){
4      int var_i = 3;
5      float var_f = 1.8;
6      char var_c = 'a';
7
8      printf("%d\n%f\n%c\n", var_i, var_f, var_c);
9
10     return 0;
11 }
```



```
❯ make -s
❯ ./main
3
1.800000
a
❯
```

실습 1

- 다음 조건을 만족하는 프로그램을 만들어봅시다
 - 점수, 성적, 석차를 저장하고 출력합니다
 - 점수: 98.25
 - 성적: A
 - 석차: 1
 - 각 항목은 알맞은 datatype의 변수에 저장합니다
 - 저장한 변수를 바탕으로 다음과 동일한 결과가 나올 수 있도록 프로그램을 만들어보세요!

```
➤ make -s
➤ ./main
Score: 98.250000
Grade: A
Rank: 1
➤ █
```

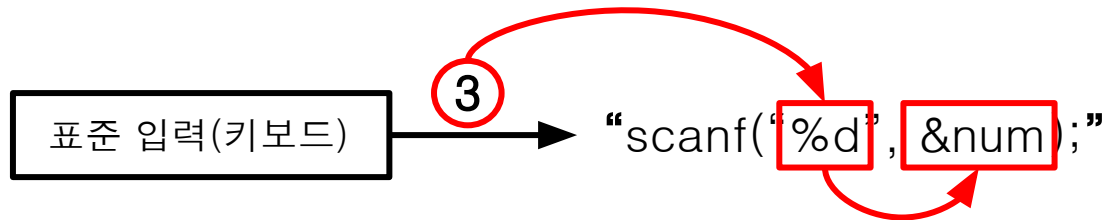
실습 2

실습 2 - 설명

- scanf를 연습해봅시다
 - 정수를 입력받는 프로그램
 - 실수를 입력받는 프로그램
 - 문자를 입력받는 프로그램

실습 2 - 설명 (scanf)

- `scanf("%d", &num);`
 - 표준 입력(키보드)으로 받은 입력을 대응되는 자료형으로 변환하여 변수에 저장합니다
 - `printf`와는 다르게 변수 앞에 ‘&’(Ampersand)를 적어야합니다



실습 2 - 설명

- 실습 예시 코드

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

실습 2 - 설명

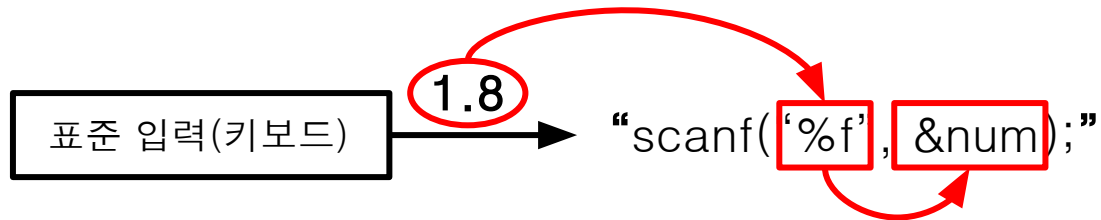
- 프로그램을 실행해봅시다
 - 이전과 다르게 프로그램이 끝나지 않고 무언가를 기다리고 있는 것을 확인할 수 있습니다
 - **콘솔창**에 원하는 정수를 적어봅시다

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

```
➤ make -s
➤ ./main
10
10
➤
```

실습 2 - 설명

- scanf(“%f”, &num);
 - printf와 마찬가지로 대응하는 서식지정자를 사용하여 다양한 datatype으로 입력 받을 수 있습니다
 - int: %d, float: %f, char: %c



실습 2 - 설명

- 예제 코드

```
main.c x
1  #include <stdio.h>
2
3  int main(void){
4      float var_f;
5      char var_c;
6
7      scanf("%f", &var_f);
8      printf("%f\n", var_f);
9
10     scanf("%c", &var_c);
11     printf("%c\n", var_c);
12
13     return 0;
14 }
15
```

Input

```
make -s
./main
1.8
1.800000
j
j
```

Output

실습 2

- 실습 1을 scanf를 이용해서 변경해봅시다
 - 실습 1에서 만든 변수에 data를 scanf로 받아 저장하여 출력합니다
 - 받아야 하는 data의 종류와 type은 동일합니다
 - 점수: 98.25
 - 성적: A
 - 석차: 1
 - 이를 바탕으로 다음과 같이 동작하는 프로그램을 작성해보세요

```
❯ make -s
❯ ./main
98.25
Score: 98.250000
A
Grade: A
1
Rank: 1
❯
```

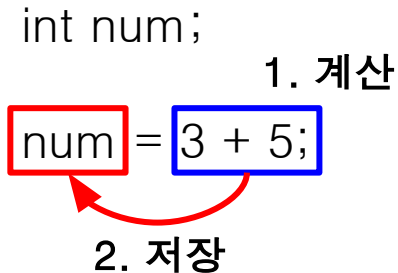
실습 2

- Tip
 - scanf를 여러번 사용하게 되면 입력을 받지 않고 그냥 넘어가는 경우가 있습니다
 - 이때는 scanf 시작 부분에 ‘ ’(띄어쓰기)를 넣어주세요
 - 예: scanf(“ %d”, &num);

실습 3

실습 3 - 설명

- 값을 계산해서 변수에 저장해봅시다
 - '='을 기준으로 좌우가 나뉩니다
 - 왼쪽은 값을 저장하는 변수
 - 오른쪽은 계산식
 - 따라서 오른쪽의 계산식을 계산하여 왼쪽에 저장합니다



실습 3 - 설명

- 예제 코드와 함께 연습해봅시다

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

실습 3 - 설명

- 변수끼리 더하는 것도 가능합니다

```
main.c ×  
1  #include <stdio.h>  
2  
3  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
```

실습 3 - 설명

- 다양한 연산자가 가능합니다
 - + : 더하기
 - - : 빼기
 - * : 곱하기
 - / : 나누기
 - 연산자 우선순위가 적용되고, 괄호로 묶는 것도 가능합니다!

num = 3 - 4 * (5 + 6);

실습 3 - 설명

- 한 변의 길이를 입력받아 정사각형의 넓이를 구해보시다
 - scanf를 이용해서 사용자에게 한 변의 길이를 입력 받습니다
 - 계산을 통해서 정사각형의 넓이를 구합니다
 - 계산한 결과를 출력해보세요

실습 3 - 설명

- 예시 코드

```
main.c x
1  #include <stdio.h>
2
3  int main(void){
4      int side;
5      int area;
6
7      printf("Side: ");
8      scanf("%d", &side);
9
10     area = side * side;
11     printf("Area: %d\n", area);
12
13     return 0;
14 }
15
```



```
➤ make -s
➤ ./main
Side: 14
Area: 196
➤
```

실습 3 - 제출

- 반지름을 입력 받아 원의 둘레와 넓이를 출력하는 프로그램을 만들어보세요
 - scanf를 이용해서 반지름을 입력 받습니다
 - 반지름은 실수의 형태입니다
 - 원 둘레, 넓이를 계산해서 차례로 출력합니다
 - π 는 3.14로 계산합니다

```
✦ make -s
✦ ./main
Radius: 1.34
Perimeter: 8.415200
Area: 5.638184
✦ □
```

실습 4

실습 4 - 설명

- 조건에 따라서 달라지는 프로그램을 만들어봅시다
 - 다음과 같은 프로그램은 조건에 따라서 결과가 달라집니다
 - 짝/홀 판단 프로그램
 - 음수/양수 판단 프로그램
 - ...
 - 프로그램에 조건을 부여하는 방법을 살펴봅시다

실습 4 - 설명

- if문
 - 다음과 같은 부분이 필요합니다
 - 경우를 나누기 위한 조건을 적는 곳
 - 조건에 부합했을 때 실행하는 곳

조건: num이 0보다 크니? `int num = 3;`

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

조건에 맞았을 때 수행하는 동작

실습 4 - 설명

- 예제 코드와 함께 연습해봅시다
 - 조건을 'num > 0'에서 'num < 0'으로 바꿔서도 실행해보세요

```
main.c x
1  #include <stdio.h>
2
3  int main(void){
4      int num = 3;
5
6      if(num > 0){
7          printf("Positive\n");
8      }
9
10     return 0;
11 }
12
```

실습 4 - 설명

- ‘else’를 이용해서 ‘조건이 아니면’의 경우를 추가할 수도 있습니다
 - ‘else’는 조건부가 없습니다(‘if 조건이 아니면’의 의미를 갖고 있기 때문)

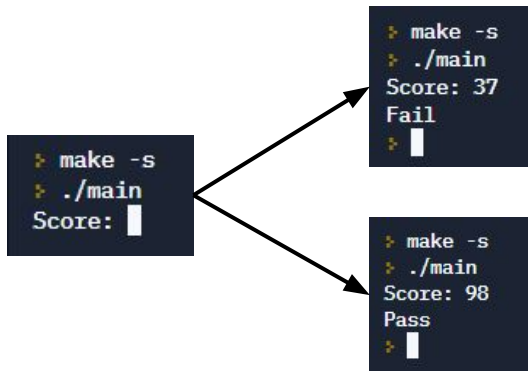
```
main.c ×
1  #include <stdio.h>
2
3  int main(void){
4      int num = -3;
5
6      if(num > 0){
7          printf("Positive\n");
8      }
9      else{
10         printf("Negative\n");
11     }
12
13     return 0;
14 }
```



```
❯ make -s
❯ ./main
Negative
❯
```


실습 4

- 점수를 입력받아 pass/fail을 판단하는 프로그램을 만들어봅시다
 - scanf를 이용해서 정수로 이루어진 점수를 입력 받습니다
 - 점수의 범위는 0~100 사이라고 가정합니다
 - 입력받은 점수가 50점 미만이면 fail, 이상이면 pass를 출력합니다



실습 5

실습 5 - 설명

- 랜덤 숫자를 만들어봅시다
 - 프로그램을 여러 번 실행해 보세요

```
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
```



```
✦ make -s
✦ ./main
441353124
✦ █
```

실습 5 - 설명

- 랜덤 숫자를 만들어봅시다

```
main.c x
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
```

← 랜덤 숫자를 만들기 위한 씨앗(Seed)

실습 5 - 설명

- 랜덤 숫자를 만들어봅시다

```
main.c x
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
```

랜덤 숫자를 만들어 줌

실습 5 - 설명

- 0~9 사이의 숫자로 만들어 봅시다
 - 나머지의 원리를 이용합니다
 - 10으로 나눈 나머지는 0에서 9 사이의 정수입니다
 - '%' (modulo) 연산이 나머지를 구합니다

1. 랜덤 숫자 생성

```
rand_num = rand() % 10;
```

2. 10으로 나눈 나머지

실습 5 - 설명

- 예시 코드

```
main.c x
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() % 10;
11     printf("%d\n", rand_num);
12
13     return 0;
14 }
15
```



```
➤ make -s
➤ ./main
4
➤
```

실습 5 - 제출

- 랜덤 숫자를 맞추는 프로그램을 만들어 봅시다
 - 랜덤 함수를 이용해서 컴퓨터가 0~4 사이의 수를 생성하여 변수에 저장합니다
 - 사용자에게 숫자를 입력 받습니다
 - 화면에 랜덤 숫자를 공개하며 정답 여부를 출력합니다

```
➤ make -s
➤ ./main
3
Correct!!
Answer: 3
➤ █
```

```
➤ make -s
➤ ./main
1
Wrong
Answer: 0
➤ █
```


추가 문제 (optional)

추가 문제

- 추가적인 프로그래밍을 하고 싶으시다면 다음 문제들을 해결해보세요
 - 숫자 3개를 입력받아 평균을 구해주는 프로그램
 - 학생의 나이, 성별을 입력 받아 출력하는 프로그램
 - 거리를 입력 받아 10보다 멀면 'far', 그렇지 않으면 'close'를 출력하는 프로그램

- 이 문제들은 제출하지 않으셔도 됩니다

과제 설명

과제 설명

- 내용
 - 당일 진행한 실습과 관련된 문제
- 문제 수
 - 2 ~ 3 문제
- 제출 기한
 - 다음 실습 수업 전날 23:59 까지
- 제출 방법
 - Replit의 Submit 기능 활용

과제 설명

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

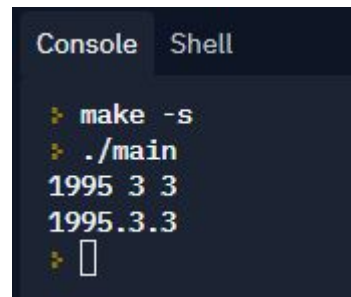
과제 설명

- 배점
 - 기본 과제(매주마다 부여되는 과제)
 - 각 10점
 - 10번의 과제, 총 100점
 - 프로젝트(학기 말 1번)
 - 50점
 - 총 점수 * $\frac{2}{3}$ * 0.4 의 방식으로 최종 점수 산출

과제 1

과제 1 - 문제

- 사용자에게 생년월일을 입력 받아 출력하는 프로그램
- 조건 1
 - 사용자는 다음과 같이 입력한다고 가정 : [년] [월] [일]
 - 반드시 정수로 입력하는 것으로 가정
 - 년, 월, 일은 띄어쓰기로 구분하여 입력
 - 예시) 1995 3 3
- 조건 2
 - 출력은 다음과 같은 형태를 만족하며 출력
 - [년].[월].[일]
 - 1995.3.3



```
Console Shell
❯ make -s
❯ ./main
1995 3 3
1995.3.3
❯
```


과제 1 - 평가 기준

- 기준 1
 - 프로그램이 동작하는가? - 1점
- 기준 2
 - 정상적으로 3개의 정수를 입력 받는가? - 2점
- 기준 3
 - 정해진 형식대로 결과를 출력하는가? - 2점

과제 2

과제 2 - 문제

- 과속 여부를 판단하는 프로그램
- 조건 1
 - 과속 기준은 100km/h
 - 사용자가 정수 형태의 속력을 입력
 - 입력 받은 값을 변수에 저장
- 조건 2
 - 입력 받은 값과 과속 기준을 비교
 - 초과 시 “Exceeded” 출력
 - 초과하지 않았으면 “Acceptable” 출력

```
❯ make -s
❯ ./main
96
Acceptable
❯
```

```
❯ make -s
❯ ./main
105
Exceeded
❯
```

과제 2 - 평가 기준

- 기준 1
 - 프로그램이 동작하는가? - 1점
- 기준 2
 - 정상적으로 속력을 입력 받는가? - 2점
- 기준 3
 - 기준에 부합하여 결과를 출력하는가? - 2점