

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

Systems Software &
Architecture Lab.

Seoul National University

Spring 2022

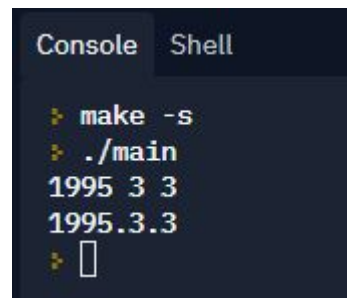
4190.103A-001: Programming Practice Lab. 3



Lab. 2 과제 풀이

Lab.2) 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
❯
```

Lab.2) 과제 1 - 풀이

```
#include <stdio.h>

int main(void){
    int year, month, day;

    scanf("%d %d %d", &year, &month, &day);
    printf("%d.%d.%d\n", year, month, day);

    return 0;
}
```

Lab.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
❯
```

Lab.2) 과제 2 - 풀이

```
#include <stdio.h>

int main(void) {
    int velocity;

    scanf("%d", &velocity);

    if(velocity > 100){
        printf("Exceeded\n");
    }
    else{
        printf("Acceptable\n");
    }

    return 0;
}
```

주식

주석

- 프로그래머의 편의를 위한 메모기능
- 프로그램 수행에는 아무런 영향을 주지 않음
- // : 하나의 행을 주석으로 처리
- /* ~ */ : 시작과 끝의 모든 행을 주석으로 처리

```
/*  
This Program is for ...  
blah,blah,blah...  
.....  
*/  
  
// if A is positive ...  
if (a > 0)  
    printf("%d is a positive number\n", a);
```


실습 1

실습 - 1 설명

- 관계 연산자
 - 두개의 피연산자를 대소 비교 후 참이면 1, 거짓이면 0

$X == Y$	X와 Y의 값이 같다
$X != Y$	X와 Y의 값이 다르다
$X < Y$	X가 Y보다 작다
$X <= Y$	X가 Y보다 작거나 같다
$X > Y$	X가 Y보다 크다
$X >= Y$	X가 Y보다 크거나 같다

실습 - 1 설명

- 아래 코드를 작성하고 실행해보세요
 - 여러 경우의 수들을 입력받아 각 연산의 결과를 확인해보세요

```
#include <stdio.h>
int main(void) {
    int a, b;
    scanf("%d %d", &a, &b);

    printf("a == b : %d \n", a == b);
    printf("a != b : %d \n", a != b);
    printf("a > b : %d \n", a > b);
    printf("a < b : %d \n", a < b);
    printf("a >= b : %d \n", a >= b);
    printf("a <= b : %d \n", a <= b);
    return 0;
}
```

```
➤ make -s
➤ ./main
1000 1000
a == b : 1
a != b : 0
a > b : 0
a < b : 0
a >= b : 1
a <= b : 1
```

실습 - 1 설명

- 산술연산자의 우선순위
 - +, - 보다 *, /, % 가 우선
 - $a+b*2 \rightarrow \underline{a+(b*2)}$

- 산술연산자와 관계연산자의 우선순위
 - 관계연산자보다 산술연산자의 우선순위가 더 높음
 - $a + b <= a * b \rightarrow \underline{(a+b) <= (a * b)}$

실습 - 1 설명

- 아래 코드를 작성하고 실행해보세요

```
#include <stdio.h>
int main(void) {
    int a, b;
    scanf("%d %d", &a, &b);

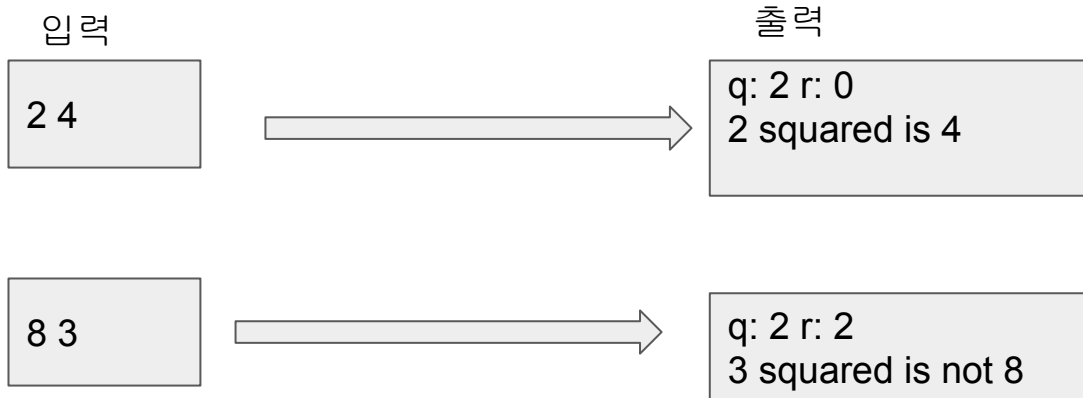
    printf("a + b * 2 : %d\n", a + b * 2);
    printf("a + (b * 2) : %d\n", a + (b * 2));
    printf("(a + b) * 2 : %d\n", (a + b) * 2);
    printf("a + b <= a * b : %d\n", a + b <= a * b);
    printf("(a + b) <= (a * b) : %d\n", (a + b) <= (a * b));
    printf("a + (b <= a) * b : %d\n", a + (b <= a) * b);

    return 0;
}
```

```
➤ ./main
2 2
a + b * 2 : 6
a + (b * 2) : 6
(a + b) * 2 : 8
a + b <= a * b : 1
(a + b) <= (a * b) : 1
a + (b <= a) * b : 4
```

실습 1 - 제출

- 두 정수를 입력받고 큰 수를 작은 수로 나눴을 때의 몫과 나머지를 출력하세요. 그리고 두 수가 제곱의 관계인지도 출력하세요
 - 제곱 관계 : $a \times a$ 가 b 와 같음



```
➤ make -s
➤ ./main
2 4
q: 2 r: 0
2 squared is 4
➤
➤ ./main
8 3
q: 2 r: 2
3 squared is not 8
```

실습 2

실습 2 - 설명

- else if문
 - else if를 통해 if, else 사이에 또 다른 조건을 만들어낼 수 있음
 - 여러 개의 else if를 사용할 수 있음

```
if (조건식) {  
}  
else if (조건식){  
}  
else if (조건식){  
}  
...  
else {  
}
```


실습 2 - 설명

- 0, 양수, 음수를 구분하는 프로그램을 작성하고 실행해보세요

```
#include <stdio.h>
int main(void) {
    int a;
    scanf("%d", &a);

    if (a > 0) {
        printf("%d is a positive number.\n", a);
    }
    else if (a < 0) {
        printf("%d is a negative number.\n", a);
    }
    else {
        printf("%d is zero.\n", a);
    }

    return 0;
}
```

```
➤ make -s
➤ ./main
-100
-100 is a negative number.
```

실습 2 -제출

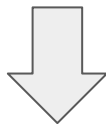
- 정수 형태의 성적을 입력받아 학점으로 변환하는 프로그램을 만들어 보세요
 - 90점 이상 100점 이하 : A
 - 80점 이상 : B
 - 70점 이상 : C
 - 60점 이상 : D
 - 60점 미만 : F
 - 100점 초과 : 입력오류
 - 음수 : 입력오류

```
➤ make -s
➤ ./main
80
Grade: B
➤ ./main
59
Grade: F
➤ ./main
110
Invalid score: 110
➤ ./main
-10
Invalid score: -10
```

실습 2 - Tip

- if 문에서 실행될 명령문이 하나라면 중괄호 { }를 생략할 수 있음

```
if (a > 0) {  
    printf("%d is a positive number.\n", a);  
}
```



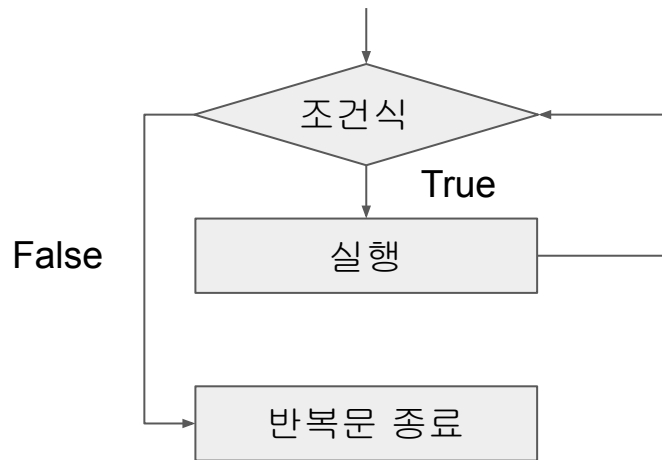
```
if (a > 0)  
    printf("%d is a positive number.\n", a);
```

실습 3

실습 3 - 설명

- while 문
 - 조건식이 참일 경우 계속 반복하는 문법

```
while(조건식) {  
    실행문 1;  
    실행문 2;  
    ...  
}
```



실습 3 - 설명

- 입력받은 숫자만큼 * 을 출력하는 프로그램을 작성하고 실행해보세요

```
#include <stdio.h>
int main(void) {
    int i, n;
    scanf("%d", &n);

    i = 0;
    while(i < n){
        printf("*");
        i = i + 1;
    }

    return 0;
}
```

```
#include <stdio.h>
int main(void) {
    int n;
    scanf("%d", &n);

    while(n > 0){
        printf("*");
        n = n - 1;
    }

    return 0;
}
```

```
➤ make -s
➤ ./main
10
*****➤
```

실습 3 - 설명

- 횟수를 증가시키는 부분을 ++, -- 연산자로 대체가능
 - `i = i + 1 -> i++`
 - `i = i - 1 -> i--`

```
#include <stdio.h>
int main(void) {
    int i, n;
    scanf("%d", &n);

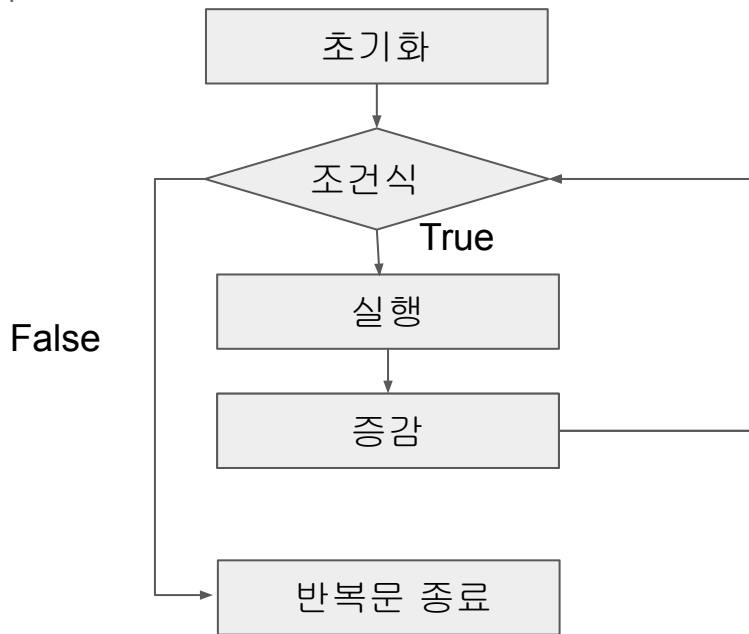
    i = 0;
    while(i < n){
        printf("*");
        i++;
    }

    return 0;
}
```

실습 3 - 설명

- for문
 - while문과 동일한 용도로 반복을 하기 위한 문법
 - 초기화 및 증감식을 함께 선언할 수 있음

```
for (초기화 ; 조건식 ; 증감) {  
    실행문 1;  
    실행문 2;  
    ...  
}
```



실습 3 - 설명

- 입력받은 숫자만큼 * 을 출력하는 프로그램을 for문으로 작성해보세요

```
#include <stdio.h>
int main(void) {
    int i, n;
    scanf("%d", &n);

    for (i = 0; i < n; i++){
        printf("*");
    }

    return 0;
}
```

실습 3 - 설명

- break문
 - 반복문을 탈출하기 위함
 - for, while문 안에서 사용 가능

```
while(조건식) {  
    실행문 1;  
    실행문 2;  
    ...  
    break;  
}
```

반복문탈출

실습 3 - 설명

- 입력받은 숫자만큼 * 을 출력하는 프로그램을 while문과 break를 이용해서 작성해보세요

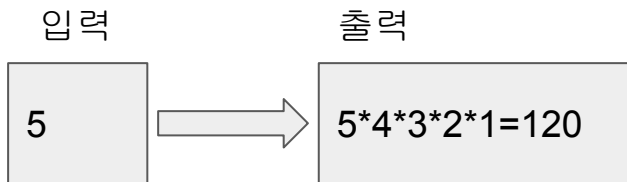
```
#include <stdio.h>
int main(void) {
    int n;
    scanf("%d", &n);

    while(1){
        if (n <= 0)
            break;

        printf("*");
        n--;
    }
    return 0;
}
```

실습 3 - 제출

- $n!$ (Factorial) 계산하는 프로그램을 만들어보세요
 - 양의 정수 n 을 입력받음
 - $n! = (n) \times (n-1) \times (n-2) \cdots \times (1)$
 - for문, while문 중 하나를 사용해서 구현해보고, 완료되었다면 다른 하나도 이용해보세요



```
➤ make -s
➤ ./main
6
6*5*4*3*2*1=720
```

실습 3

- 문자를 하나씩 반복하여 입력받는 프로그램을 작성해보세요
 - 문자를 한번에 하나씩 입력받음
 - 'z' 가 총 3번 입력이 되면 프로그램을 종료함
 - 프로그램이 종료될때 "End" 를 출력함

```
> make -s
> ./main
a
b
z
a
b
z
a
b
z
End
```

실습 4

실습 4 - 제출

- 양의 정수를 분석하는 프로그램을 만들어보세요
 - 하나의 양의 정수를 입력받음
 - 양의 정수가 아니면 더 이상 진행하지 않고 종료함
 - 짝수, 홀수인지 출력함
 - even, odd
 - 자리수를 출력함
 - x-digit number
 - 소수인지 아닌지 확인하여 출력함
 - 소수 : 약수가 1과 자기 자신뿐인 수
 - 1은 소수가 아님
 - for문과 %(modulo)를 사용하여 구현

```
> make -s
> ./main
7
7 is a odd number
7 is a 1-digit number
7 is a prime number
>
> ./main
1214562
1214562 is a even number
1214562 is a 7-digit number
1214562 is not a prime number
>
> ./main
1
1 is a odd number
1 is a 1-digit number
1 is not a prime number
>
> ./main
-109
ERROR : -109 is not a positive number
```

음수를
입력받을 경우
예외처리

실습 4

- Up & Down 게임을 만들어 보세요
 - 컴퓨터가 $0 \leq X < 32$ 사이의 임의의 숫자를 하나 선정 (Random 함수사용)
 - 사용자에게 5번의 기회를 주고 각 기회마다 하나의 숫자를 입력
 - 입력받은 숫자와 답을 비교해서 Up, Down 중 하나를 출력
 - 답이 맞으면 Success를 출력하고 종료하고 정해진 횟수에 끝나지 않았을 경우 Fail을 출력

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

int main(void) {
    srand(time(NULL));
    int random_value = rand();
```

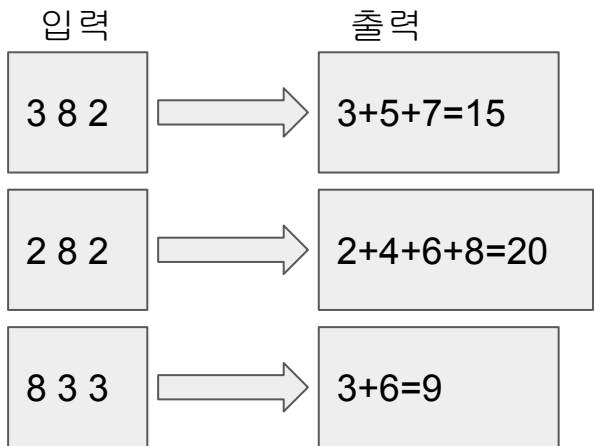
```
➤ make -s
➤ ./main
1: 16
Down
2: 8
Down
3: 4
Up
4: 6
Success➤
```

```
➤ make -s
➤ ./main
1: 16
Up
2: 17
Up
3: 18
Up
4: 19
Up
5: 20
Up
Fail. The answer is 26
```


과제 1

과제 1

- 양의 정수를 세개(n, m, k) 입력받고 n 에서 m , 혹은 m 에서 n 으로 k 만큼 증가시켜가며 더하는 프로그램
 - 양의 정수 세개(n, m, k)를 입력
 - n, m 의 대소관계를 파악하여 작은수에서 큰수까지 k 씩 증가시켜 더함



```
❖ make -s
❖ ./main
3 8 2
3+5+7=15
❖
❖ ./main
2 8 2
2+4+6+8=20
❖
❖ ./main
8 3 3
3+6=9
```

과제 1 - 평가 기준

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

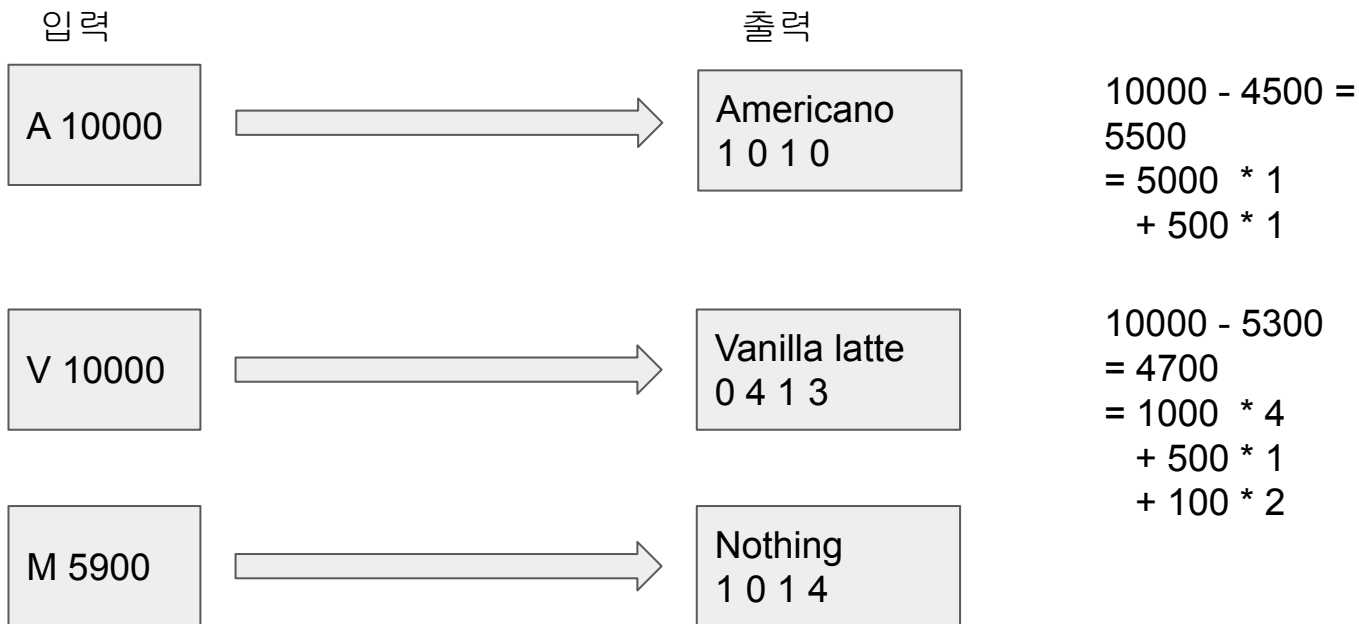
과제 2

과제 2

- 자판기에서 음료의 이름과 잔돈의 개수를 출력하는 프로그램
 - 음료를 나타내는 문자 N과 투입한 금액을 나타내는 정수 M(100의 배수)를 입력
 - 음료
 - 'A' = Americano(4500원), 'L' = Caffe latte(4800원),
 - 'M' = Caffe mocha(6000원), 'V' = Vanilla latte(5300원)
 - 잔돈
 - 5000원, 1000원, 500원, 100원
 - 단위가 높은 돈을 우선적으로 지급
 - 주문한 음료와 잔돈의 개수를 출력
 - 잔돈의 개수는 큰 단위부터 공백을 두고 출력
 - 투입한 금액이 선택한 음료의 가격보다 낮다면 'Nothing'을 출력하고 투입한 금액 전부를 반환

과제 2

- 출력 예시



과제 2 - 평가 기준

- 기준 1
 - 프로그램이 동작하는가? - 1점
- 기준 2
 - 정해진 형식대로 입력을 받는가? - 1점
- 기준 3
 - 정해진 형식대로 결과를 출력하는가? - 3점