

Object-Oriented Programming

Lab #02

Department: 화학공학과

Student ID: 2019101074

Name: 안용상

* 코드를 작성하는 문제는 1) 주석(설명)이 포함된 소스코드와 2) 실행 결과를 포함 (모두 text copy)

1. C++에서 정수는 bool 타입으로 어떻게 평가되는가?

0 은 False

1 은 True로 평가된다.

2. 각 변수가 아래와 같이 초기화되었을 때,

```
int x = 3, y = 5, z = 7;
```

```
bool b1 = true, b2 = false, b3 = x == 3, b4 = y < 3;
```

아래의 결과를 출력해보고, 그 결과가 나오는 이유를 설명하라.

(a) `x == 3`

(b) `x < y`

(c) `x >= y`

(d) `x <= y`

(e) `x != y - 2`

(f) `x < 10`

(g) `x >= 0 && x < 10`

(h) `x < 0 && x < 10`

(i) `x >= 0 && x < 2`

(j) `x < 0 || x < 10`

(k) `x > 0 || x < 10`

(l) `x < 0 || x > 10`

(m) `b1`

(n) `!b1`

(o) `!b2`

(p) `b1 && b2`

(q) `b3`

(r) `b4`

출력을 하려면 다음과 같이 코드를 짜야한다

<코드>

```

#include <iostream>
int main() {
    int x = 3, y = 5, z = 7;
    bool b1 = true, b2 = false, b3 = x == 3, b4 = y < 3;
    std::cout << (x == 3) << std::endl; // 1 출력
    std::cout << (x < y) << std::endl; // 1 출력
    std::cout << (x >= y) << std::endl; // 0 출력
    std::cout << (x <= y) << std::endl; // 1 출력
    std::cout << (x != y - 2) << std::endl; // 1 출력
    std::cout << (x < 10) << std::endl; // 1 출력
    std::cout << (x >= 0 && x < 10) << std::endl; // 1 출력
    std::cout << (x < 0 && x < 10) << std::endl; // 0 출력
    std::cout << (x >= 0 && x < 2) << std::endl; // 0 출력
    std::cout << (x < 0 || x < 10) << std::endl; // 1 출력
    std::cout << (x > 0 || x < 10) << std::endl; // 1 출력
    std::cout << (x < 0 || x > 10) << std::endl; // 0 출력
    std::cout << b1 << std::endl; // 1 출력
    std::cout << !b1 << std::endl; // 0 출력
    std::cout << !b2 << std::endl; // 1 출력
    std::cout << (b1 && b2) << std::endl; // 0 출력
    std::cout << b3 << std::endl; // 1 출력
    std::cout << b4 << std::endl; // 0 출력
    return 0;
}

```

위와 같이 코드를 짜서 출력을 하면

1. `x == 3`

출력결과 1

이유 : `x`는 `int`형 3이니까 `x`가 3과 일치하는지 여부는 `True`로 판정되고 이는 다시 `True`를 나타내는 1로 반환된다

2. `x < y`

출력결과 1

이유 : `x`는 3이고 `y`는 5니까 5가 3보다 크다는 명제는 참값을 가져서 `True`가 되고 그 값인 1이 반환된다

3. `x >= y`

출력결과 0

이유 : `x`가 `y`보다 크거나 같다는 명제는 , 3이 5보다 크거나 같다는 명제이고 이는 거짓으로 `False` 즉 0이 반환된다

4. `x <= y`

출력결과 1

이유 : `x`가 `y`보다 작거나 같다는 명제는 3이 5보다 작거나 같다 라는 명제이므로 참

값을 가져 1이 반환된다

5. $x \neq y - 2$

출력결과 0

이유 : x 가 y 에서 2를 뺀 값과 같지 않다는 명제는, 3이 $(5-2)$ 와 같지 않다는 명제이므로 이는 계산했을 때 거짓값을 가지므로 `False`, 즉 0이 반환된다.

6. $x < 10$

출력결과 1

이유 : x 가 10보다 작다는 명제는 3이 10보다 작다는 명제이므로 참값을 가져서 1이 반환된다

7. $x \geq 0 \ \&\& \ x < 10$

출력결과 1

이유 : x 가 0보다 같고 x 가 10보다 작다는 합성명제는, `&&`연산자에 의해 논리 AND연산이 되는데, 왼쪽 명제와 오른쪽 명제가 둘 다 참이므로 1이 반환된다

8. $x < 0 \ \&\& \ x < 10$

출력결과 0

이유 : x 가 0보다 작고 x 가 10보다 작다는 합성명제이다. 이때 왼쪽 명제가 거짓이므로 이는 AND연산에 의해 `FALSE`, 즉 0이 반환된다.

9. $x \geq 0 \ \&\& \ x < 2$

출력결과 0

이유 : x 가 0보다 크거나 작고, x 가 2보다 작다는 합성명제이다. 이 때, 오른쪽 명제인 $x < 2$ 가 거짓 `False`이므로 0이 반환된다.

10. $x < 0 \ || \ x < 10$

출력결과 1

이유 : x 가 0보다 작거나 x 가 10보다 크다는 OR연산이다. 이때 오른쪽 명제가 참이므로 왼쪽명제에 상관없이 이 연산값은 `True`가 되고 1이 반환된다.

11. $x > 0 \ || \ x < 10$

출력결과 1

x 가 0보다 크거나 x 가 10보다 작다는 양쪽 명제가 다 `True`이므로 OR연산 결과는 1이 반환된다

12. $x < 0 \ || \ x > 10$

출력결과 0

양쪽 명제가 모두 거짓이므로 OR연산 결과는 0이 반환된다

13. b1

출력결과 1

이유 : b1은 그 자체로 True인 명제로, 이는 출력 시 1로 반환된다

14. !b1

출력결과 0

True를 not 연산해주었으므로 이는 False가 되는 명제이고 따라서 0이 반환된다.

15. !b2

출력결과 1

false를 not 연산 해주었으므로 이는 True가 되고 따라서 1이 반환된다

16. b1 && b2

출력결과 0

True와 False가 AND 연산 되었으므로 이는 0이 반환된다

17. b3

출력결과 1

b3는 $x == 3$ 이라는 명제를 가지고 있다. x는 int형 3이 맞으므로 이 명제는 True가 되고 따라서 이는 1을 반환한다.

18. b4

출력결과 0

b4는 y가 3보다 작다는 명제인데 y는 5로 3보다 크므로 이는 거짓인 명제이고 따라서 b4는 그자체로 False인 명제이며 결과값은 0이 반환된다.

3. 아래의 수식을 간략하게 표현하라.

(a) $!(x == 2)$

$x != 2$

(b) $x < 2 \ || \ x == 2$

$x <= 2$

(c) $!(x < y)$

$x >= y$

(d) $!(x <= y)$

$x > y$

```
(e) x != 0
    x != 0
(f) x == 0
    x == 0
```

4. 정수를 입력받아서, 2의 배수, 3의 배수, 6의 배수, 9의 배수를 판별(복수 판별)하는 프로그램을 작성하라.

<코드>

```
#include <iostream>
#include <stdio.h>
int main() {
    int user_input;
    std::cout << "배수 판별 대상 숫자를 입력해주세요~" << std::endl;
    std::cin >> user_input;
    if (std::cin.fail() == 1) { //입력받은 수가 정수형이 아닐 때를 체크
        while (std::cin.fail() == 1) { //입력받은 수가 정수형이지 않으면 계속 해서
            //정수형이 나올 때까지 cin을 다시받는 코드
            std::cin.clear();
            std::cin.ignore(10000, 'Wn'); //cin의 입력버퍼를 지우기
            std::cout << "다시 입력하세요~" << std::endl;
            std::cin >> user_input;
        }
    }
    printf("당신이 입력한 수는 %d Wn-----Wn", user_input);
    int myArray[4] = { 2, 3, 6, 9 }; //문제에서 주어진 나눌 수에 대한
        //정보를 어레이에 순차적으로 담음
    int result[4]; //결과값을 담을 길이 4인 빈 어레이를 만들
    int count_0 = 0; //결과값이 0인 경우를 카운트하는 초기값이 0인 정수 정의
    for (int i = 0; i <= 3; i++) {
        result[i] = user_input % myArray[i]; //for문을 돌리면서 나눌 수를 순차적으로
        //바꾸고 입력받은 수를 나눴을 때, 나머지를 담음
        if (result[i] == 0) {
            count_0 += 1; //나머지가 0이면, 즉 배수이면 카운트를 셈
            printf("%d의 배수 ", myArray[i]);
        }
    }
    if (count_0 == 0) { //카운트가 하나도 증가하지 않았다면
        //나머지가 0인경우가 한번도 없었다는 것이므로
        //이때는 제시된 수들 중, 어떤 수의 배수도 아님
        printf("2, 3, 6, 9의 배수가 아닙니다!!!");
    }
    else {
        printf("입니다!!!");
    }
}
```

<출력결과>

배수 판별 대상 숫자를 입력해주세요~

36

당신이 입력한 수는 36

2의 배수 3의 배수 6의 배수 9의 배수 입니다!!!

C:\Users\Wqhfdk\Project1\wx64\Debug\Project1.exe(프로세스 24520개)이(가)
 종료되었습니다(코드: 0개) .
 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이
 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
 이 창을 닫으려면 아무 키나 누르세요...

```

Microsoft Visual Studio 디버그 콘솔

배수 판별 대상 숫자를 입력해주세요~
36
당신이 입력한 수는 36
-----
2의 배수 3의 배수 6의 배수 9의 배수 입니다!!!
C:\Users\Wqhfdk\Project1\wx64\Debug\Project1.exe(프로세스 24520개)이(가)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
  
```

5. Dangling else problem에 대해 설명하라.

if / else 문에서 else는 if가 먼저 나오고 후속으로 남은 경우의 수를 처리해주는 구문인데,

이때 else는 짝 if문을 찾아서 그 if에 해당되지 않는 경우를 처리해준다

프로그래머가 if / else문을 짤 때, 의도치 않은 결과를 반환 받지 않으려면, if /else가 어떻게 짝지어지는지를 알아야 한다

중괄호로 구분하지 않은 else는 짝이 되는 if를 가장 가까운 위치에서 찾는다

따라서, 가장 가까운 if의 else구문이 맞는 상황일 때는, 그렇게 두어도 괜찮다. 하지만 그게 아니라 가장 가까운 if를 넘어서서 조금 더 먼 위치에 있는 if문의 짝이 되는 else라면 이는 반드시 중괄호를 통해서 그 if의 짝임을 명시해줘야 한다. 같은 스코프에 두어야 한다는 것이다.

이때, 그렇게 하지 않고 의도치 않은 결과를 받게 되는 문제상황을 Dangling else problem이라고 한다.

예를들면 다음과 같다.

```

#include <iostream>
#include <stdio.h>
int main() {
    if (1 == 0)
        if (1 == 1)
            printf("case 01");
    else
        printf("case02");
}
  
```

위 경우에 들여쓰기 상으로는 else가 첫번째 if와 연결되는게 맞다면, case02가 출력되어야 하겠지만, 사실은 else는 두번째 if와 짝지어서 결과는 아무것도 출력하지 않게된다.

이런 Dangling else problem을 방지하려면 { }를 써서 어떤 if의 else문인지 명확하게 명시해줘야한다.

6. 아래 코드의 동작을 설명하라.

```
(a) int x;
    std::cin >> x;
    if(x%2) std::cout << x << std::endl;
```

>>>가장 먼저 사용자에게 입력을 받아서 int형 변수 x에 값을 담는다.

그리고 그것을 2로 나눈 나머지가 1이라면, 즉 그 입력받은 정수가 홀수라면 if문 속의 내용이 실행되고, 그 홀수 정수가 출력이 된다. 그리고 줄바꿈을 출력한다

```
(b) int x;
    std::cin >> x;
    if(!(x%3)) std::cout << x << std::endl;
```

>>>가장 먼저 사용자에게 입력을 받아서 int형 변수 x에 값을 담는다.

그리고 그것을 3으로 나눴을 때의 나머지인 정수 값을 not연산 해준다

이때 not연산의 결과는 나머지 정수 값이 0이었던 경우만 1이되어 True값으로 인식되고

따라서 3의 배수인 경우에만 나머지가 0인경우가 되므로, 3의 배수인 경우에 그 입력정수를 출력하게 된다 그리고 줄바꿈을 출력한다

```
(c) int x;
    std::cin >> x;
    x = (x>=0?1:-1);
```

>>>가장 먼저 사용자에게 입력을 받아서 int형 변수 x에 값을 담는다.

연산자 우선 순위에 의해서 괄호안의 삼항 연산이 먼저 실행된다.

입력 받은 정수가 0과 같거나 큰 정수인지 아닌지를 판별하고, 0과 같거나 큰 정수일 경우 1을 그렇지 않을 때는 -1을 반환하고 이를 다시 x변수에 재할당한다.

```
(d) int x = 10, y;
    y = ((x%2)?++x:--x);
    std::cout << x << ", " << y << std::endl;
```

>>>가장 먼저 사용자에게 입력을 받아서 int형 변수 x에 값을 담는다.

그리고 연산자 우선순위에 의해서 입력 받은 정수를 2로 나눈 나머지가 1인 경우, `++x` 연산을 해주고, 그렇지 않을 때는 `--x` 연산을 해준다. 즉 홀수일때 `x`값에 1을 더해주고 짝수일 때 `x` 값에 1을 빼준다는 소리

그리고 `++x` 혹은 `--x` 연산값을 `y` 변수에 할당해준다.

그리고 나서 다시

`(x의 값), (y값)\n`

의 출력을 한다

```
(e) int x = 11, y;  
    y = ((x%2)?++x:--x);  
    std::cout << x << ", " << y << std::endl;
```

>>> 가장 먼저 사용자에게 입력을 받아서 `int`형 변수 `x`에 값을 담는다.

연산자 우선 순위에 의해서, 삼항연산이 먼저 시작된다. 11로 초기화된 `x`를 2로 나눈 나머지를 판단한다. 만약 1이면 `++x`를 0이면 `--x`를 수행해 그 값으로 `y`를 초기화한다

여기서 11은 홀수이므로 2로 나눈 나머지가 1이고 따라서 `++x` 연산이 수행되어 11+1인 12가 `y`에 할당된다. 이때 `x`값 자체도 1이 더해진다

그리고 출력값은

`12,12\n`

이 된다

```
(f) int x, y;  
    std::cin >> x >> y;  
    if(x>y);  
    else std::cout << x << std::endl;
```

>>> 가장 먼저 사용자에게 입력을 받아서 `int`형 변수 `x`와 `y`에 정수를 차례로 담는다

이때 `x`가 `y`보다 큰 경우 아무것도 실행되지 않고, `x`가 `y`보다 작은 `else`의 경우 `x`를 출력한다.

-
7. 두 정수형 변수 `a`와 `b`를 입력을 받고, `flag`를 입력받아서, 1) `flag`가 0이면 합, 2) `flag`가 1이면 평균(실수), 3) `flag`가 2이면 큰 값, 4) `flag`가 3이면 작은 값을 출력하는 프로그램을 작성하라. (모두 `if`문으로 작성)

<코드>

```
#include <iostream>
```



```

#include <stdio.h>
int main() {
    int a, b, flag;
    std::cout << "변수 a를 입력하십시오";
    std::cin >> a; // a입력받기
    //정수형에 실수를 입력받을 때, 정수부분만 취하고, 소수점 아래부분을 입력버퍼에
    남겨두게되어서
    //추후 문제가 생긴다. 그리고 정수를 받기로 했는데, 실수를 받고 그냥 넘어가면
    안된다
    //따라서 std::cin.peek()가 10을 반환하지 않으면 반복해서 다시 변수 a를 입력받게
    했다.
    //10을 반환할때를 기준으로 삼은 이유는, 정수를 올바르게 받으면 입력버퍼에 딱 Wn만
    남기때문에
    //10을 반환하지만, 실수를 입력받으면 입력버퍼에 다른 숫자들도 남기때문에 10이 아닌
    숫자가 반환되기 때문이다.
    /// 그리고 std::cin.fail() == 1은 실수, 정수도 아닌 문자같은 것을 입력받았을 때,
    입력fail의 여부를 판단하기 위해 넣었다
    //그래서 정수가 아닌것을 입력받았을 모든 경우를 std::cin.fail() == 1 ||
    std::cin.peek() != 10로 양쪽 변을 OR연산하며
    //표현했다.
    while (std::cin.fail() == 1 || std::cin.peek() != 10) {
        std::cin.clear();
        std::cin.ignore(10000, 'Wn');
        printf("다시 변수 a를 입력해주세요!");
        std::cin >> a;
    }
    //b도 a와 동일하게 처리했다,
    std::cout << "변수 b를 입력하십시오";
    std::cin >> b;
    while (std::cin.fail() == 1 || std::cin.peek() != 10) {
        std::cin.clear();
        std::cin.ignore(10000, 'Wn');
        printf("다시 변수 b를 입력해주세요!");
        std::cin >> b;
    }
    int checked = 0; //check라는 변수를 flag를 올바르게 입력받았는지를 판별하기 위한
    변수로 사용하기 위해 0으로 초기화한다.
    while (checked == 0) {
        printf("a와 b를 어떻게 처리할까요?Wn합 : 0입력Wn평균 : 1입력WnMin값 : 2입력Wn-
        -----Wn");
        std::cin >> flag;
        while (std::cin.fail() == 1 || std::cin.peek() != 10) { //정수가 아닌 것을
        입력받은 경우를 판단한다. 그렇지 않은경우 계속 재입력받는다.
            std::cin.clear();
            std::cin.ignore(10000, 'Wn');
            std::cout << "다시 입력해주세요!Wna와 b를 어떻게 처리할까요?Wn합 :
            0입력Wn평균 : 1입력WnMin값 : 2입력Wn-----Wn";
            std::cin >> flag;
        }
        //이번에는 flag에 할당된 정수가 0,1,2중의 한 숫자가 맞는지 판단한다.
        //만약 맞다면 checked는 0이 아닌 양의 정수가 된다,
        //그리고 이경우 while문이 끝나게 된다.
        for (int i = 0; i <= 2; i++) {
            if (i == flag) {
                checked += 1;
            }
        }
    }
}

```

```

    }
}
if (checked > 0) {
    printf("입력을 받았습니다.\n입력모드 %d\n\n=====Wn\n", flag);
}
else {
    printf("다시 입력하세요!\n");
}
}

//flag가 최종적으로 0일때 printf에 포맷해서 합을 출력한다.
if (flag == 0) {
    printf("합을 구해드리겠습니다!");
    printf("합 : %d", a + b);
}

//flag가 최종적으로 1일때 printf에 포맷해서 정수를 출력한다.
//이때 분모를 실수 2.0으로 뒤서, 출력포맷 %f에 적절히 실수로써 들어가게 한다.
//왜냐하면 분자가 정수이기 때문에, 분모를 실수로해야 연산결과도 실수로 표현되기 때문이다.
else if (flag == 1) {
    printf("평균을 구해드리겠습니다!");
    printf("평균 : %f", (a + b)/2.0);
}

//flag가 최종적으로 2일때, Min값을 구해준다
// min이라는 변수를 만들고, a와 b를 if문으로 비교해서 작은값을 min변수에 할당 후
//min을 출력한다.
else if (flag == 2) {
    printf("Min을 구해드리겠습니다!");
    int min;
    if (a > b) {
        min = b;
    }
    else {
        min = a;
    }
    printf("Min : %d", min);
}
}
}

```

<출력결과 : flag가 0(합)인 경우 >

변수 a를 입력하시오12345

변수 b를 입력하시오54321

a와 b를 어떻게 처리할까요?

합 : 0입력

평균 : 1입력

Min값 : 2입력

0

입력을 받았습니다.

입력모드 0

=====

합을 구해드리겠습니다!합 : 66666

<출력결과 : flag가 1(평균)인 경우>

변수 a를 입력하시오12345
변수 b를 입력하시오54321
a와 b를 어떻게 처리할까요?
합 : 0입력
평균 : 1입력
Min값 : 2입력

1
입력을 받았습니다.
입력모드 1

=====

평균을 구해드리겠습니다!평균 : 33333.000000

<출력결과 : flag가 2(Min값)인 경우>

변수 a를 입력하시오12345
변수 b를 입력하시오54321
a와 b를 어떻게 처리할까요?
합 : 0입력
평균 : 1입력
Min값 : 2입력

2
입력을 받았습니다.
입력모드 2

=====

Min을 구해드리겠습니다!Min : 12345

8. 7번 문제에서 flag의 판별을 switch문을 사용하여 수정하라.

<코드>

```
#include <iostream>
#include <stdio.h>
int main() {
    int a, b, flag;
    std::cout << "변수 a를 입력하시오";
    std::cin >> a; // a입력받기
    //정수형에 실수를 입력받을 때, 정수부분만 취하고, 소수점 아래부분을 입력버퍼에
    남겨두게되어서
    //추후 문제가 생긴다. 그리고 정수를 받기로 했는데, 실수를 받고 그냥 넘어가면
    안된다
    //따라서 std::cin.peek()가 10을 반환하지 않으면 반복해서 다시 변수 a를 입력받게
    했다.
    //10을 반환할때를 기준으로 삼은 이유는, 정수를 올바르게 받으면 입력버퍼에 딱 \n만
    남기때문에
    //10을 반환하지만, 실수를 입력받으면 입력버퍼에 다른 숫자들도 남기때문에 10이 아닌
    숫자가 반환되기 때문이다.
    /// 그리고 std::cin.fail() ==1는 실수, 정수도 아닌 문자같은 것을 입력받았을 때,
    입력fail의 여부를 판단하기 위해 넣었다
```

```

//그래서 정수가 아닌것을 입력받았을 모든 경우를 std::cin.fail() == 1 ||
std::cin.peek() != 10로 양쪽 변을 OR연산하며
//표현했다.
while (std::cin.fail() == 1 || std::cin.peek() != 10) {
    std::cin.clear();
    std::cin.ignore(10000, '\n');
    printf("다시 변수 a를 입력해주세요!");
    std::cin >> a;
}
//b도 a와 동일하게 처리했다,
std::cout << "변수 b를 입력하십시오";
std::cin >> b;
while (std::cin.fail() == 1 || std::cin.peek() != 10) {
    std::cin.clear();
    std::cin.ignore(10000, '\n');
    printf("다시 변수 b를 입력해주세요!");
    std::cin >> b;
}
int checked = 0; //check라는 변수를 flag를 올바르게 입력받았는지를 판별하기 위한
변수로 사용하기 위해 0으로 초기화한다.
while (checked == 0) {
    printf("a와 b를 어떻게 처리할까요?\n합 : 0입력\n평균 : 1입력\nMin값 : 2입력\n-
-----\n");
    std::cin >> flag;
    while (std::cin.fail() == 1 || std::cin.peek() != 10) { //정수가 아닌 것을
입력받은 경우를 판단한다. 그렇지 않은경우 계속 재입력받는다.
        std::cin.clear();
        std::cin.ignore(10000, '\n');
        std::cout << "다시 입력해주세요!\na와 b를 어떻게 처리할까요?\n합 :
0입력\n평균 : 1입력\nMin값 : 2입력\n-----\n";
        std::cin >> flag;
    }
    //이번에는 flag에 할당된 정수가 0,1,2중의 한 숫자가 맞는지 판단한다.
    //만약 맞다면 checked는 0이 아닌 양의 정수가 된다,
    //그리고 이경우 while문이 끝나게 된다.
    for (int i = 0; i <= 2; i++) {
        if (i == flag) {
            checked += 1;
        }
    }
    if (checked > 0) {
        printf("입력을 받았습니다.\n입력모드 %d\n\n===== \n", flag);
    }
    else {
        printf("다시 입력하세요!\n");
    }
}
//여기부터 바뀐부분
switch (flag) {
case 0: //flag가 0일때
    printf("합을 구해드리겠습니다!");
    printf("합 : %d", a + b);
    break;
case 1: //flag가 1일 때
    printf("평균을 구해드리겠습니다!");

```

```

        printf("평균 : %f", (a + b) / 2.0);
        break;
    case 2: //flag가 2일 때
        printf("Min을 구해드리겠습니다!");
        int min;
        if (a > b) {
            min = b;
        }
        else {
            min = a;
        }
        printf("Min : %d", min);
        break;
    default:
        printf("잘못된 입력입니다.");
    }

    return 0;

}

```

<출력결과는>

<출력결과 : flag가 0(합)인 경우 >

변수 a를 입력하시오12345

변수 b를 입력하시오54321

a와 b를 어떻게 처리할까요?

합 : 0입력

평균 : 1입력

Min값 : 2입력

0

입력을 받았습니다.

입력모드 0

=====

합을 구해드리겠습니다!합 : 66666

<출력결과 : flag가 1(평균)인 경우>

변수 a를 입력하시오12345

변수 b를 입력하시오54321

a와 b를 어떻게 처리할까요?

합 : 0입력

평균 : 1입력

Min값 : 2입력

1

입력을 받았습니다.

입력모드 1

=====

평균을 구해드리겠습니다!평균 : 33333.000000

<출력결과 : flag가 2(Min값)인 경우>

변수 a를 입력하시오12345

변수 b를 입력하시오54321

a와 b를 어떻게 처리할까요?

합 : 0입력

평균 : 1입력

Min값 : 2입력

2

입력을 받았습니다.

입력모드 2

=====

Min을 구해드리겠습니다!Min : 12345

9. 7번 문제에서 if문을 가능한한 조건연산자(conditional operator)로 수정하라.

7번문제에서 flag를 판별하고 그에 맞는 연산을 출력하는 로직을

조건연산자만으로 짜게되면 다음과 같이 짤 수 있다.

<부분코드>

```
(flag==0)? printf("합 : %d", a + b):(flag==1)? printf("평균 : %f", (a + b) / 2.0):(a>b)?  
printf("Min : %d", b): printf("Min : %d", a);
```

위와 같이 코드를 수정해도 똑같이 목적이 달성이 된다.

<전체 실행 코드>

```
#include <iostream>  
#include <stdio.h>  
int main() {  
    int a, b, flag;  
    std::cout << "변수 a를 입력하시오";  
    std::cin >> a; // a입력받기  
    //정수형에 실수를 입력받을 때, 정수부분만 취하고, 소수점 아래부분을 입력버퍼에  
    남겨두게되어서  
    //추후 문제가 생긴다. 그리고 정수를 받기로 했는데, 실수를 받고 그냥 넘어가면  
    안된다  
    //따라서 std::cin.peek()가 10을 반환하지 않으면 반복해서 다시 변수 a를 입력받게  
    했다.  
    //10을 반환할때를 기준으로 삼은 이유는, 정수를 올바르게 받으면 입력버퍼에 딱 Wn만  
    남기때문에  
    //10을 반환하지만, 실수를 입력받으면 입력버퍼에 다른 숫자들도 남기때문에 10이 아닌  
    숫자가 반환되기 때문이다.  
    /// 그리고 std::cin.fail() == 1은 실수, 정수도 아닌 문자같은 것을 입력받았을 때,  
    입력fail의 여부를 판단하기 위해 넣었다  
    //그래서 정수가 아닌것을 입력받았을 모든 경우를 std::cin.fail() == 1 ||  
    std::cin.peek() != 10로 양쪽 변을 OR연산하며  
    //표현했다.  
    while (std::cin.fail() == 1 || std::cin.peek() != 10) {  
        std::cin.clear();
```

```

        std::cin.ignore(10000, '\n');
        printf("다시 변수 a를 입력해주세요!");
        std::cin >> a;
    }
    //b도 a와 동일하게 처리했다,
    std::cout << "변수 b를 입력하십시오";
    std::cin >> b;
    while (std::cin.fail() == 1 || std::cin.peek() != 10) {
        std::cin.clear();
        std::cin.ignore(10000, '\n');
        printf("다시 변수 b를 입력해주세요!");
        std::cin >> b;
    }
    int checked = 0; //check라는 변수를 flag를 올바르게 입력받았는지를 판별하기 위한
    변수로 사용하기 위해 0으로 초기화한다.
    while (checked == 0) {
        printf("a와 b를 어떻게 처리할까요?Wn합 : 0입력Wn평균 : 1입력WnMin값 : 2입력Wn-
        -----Wn");
        std::cin >> flag;
        while (std::cin.fail() == 1 || std::cin.peek() != 10) { //정수가 아닌 것을
        입력받은 경우를 판단한다. 그렇지 않은경우 계속 재입력받는다.
            std::cin.clear();
            std::cin.ignore(10000, '\n');
            std::cout << "다시 입력해주세요!Wna와 b를 어떻게 처리할까요?Wn합 :
            0입력Wn평균 : 1입력WnMin값 : 2입력Wn-----Wn";
            std::cin >> flag;
        }
        //이번에는 flag에 할당된 정수가 0,1,2중의 한 숫자가 맞는지 판단한다.
        //만약 맞다면 checked는 0이 아닌 양의 정수가 된다,
        //그리고 이경우 while문이 끝나게 된다.
        for (int i = 0; i <= 2; i++) {
            if (i == flag) {
                checked += 1;
            }
        }
        if (checked > 0) {
            printf("입력을 받았습니다.Wn입력모드 %dWnWn=====WnWn", flag);
        }
        else {
            printf("다시 입력하세요!Wn");
        }
    }
    //여기부터 바뀐부분
    (flag==0)? printf("합 : %d", a + b):(flag==1)? printf("평균 : %f", (a + b) /
    2.0):(a>b)? printf("Min : %d", b): printf("Min : %d", a);

    return 0;
}

```

<출력결과>

Flag가 0일 때

변수 a를 입력하시오123

변수 b를 입력하시오312

a와 b를 어떻게 처리할까요?

합 : 0입력

평균 : 1입력

Min값 : 2입력

0

입력을 받았습니다.

입력모드 0

=====

합 : 435

Flag가 1일 때

변수 a를 입력하시오123

변수 b를 입력하시오321

a와 b를 어떻게 처리할까요?

합 : 0입력

평균 : 1입력

Min값 : 2입력

1

입력을 받았습니다.

입력모드 1

=====

평균 : 222.000000

Flag가 2일 때

변수 a를 입력하시오123

변수 b를 입력하시오321

a와 b를 어떻게 처리할까요?

합 : 0입력

평균 : 1입력

Min값 : 2입력

2

입력을 받았습니다.

입력모드 2

=====

Min : 123

10. 100만점의 중간고사, 퀴즈, 기말고사를 입력 받아서, 평균(정수)이 1) 90점 이상이면 A, 2) 90점 미만이고 80점 이상이면 B, 3) 80점 미만이고 70점 이상이면 C, 4) 70점 미만이고 60점 이상이면 D, 5) 60점 미만이면 F를 출력하는 프로그램을 작성하라.

```
/*10. 100만점의 중간고사, 퀴즈, 기말고사를 입력 받아서,
평균(정수)이
1) 90점 이상이면 A,
2) 90점 미만이고 80점 이상이면 B,
3) 80점 미만이고 70점 이상이면 C,
4) 70점 미만이고 60점 이상이면 D,
5) 60점 미만이면 F를 출력하는 프로그램을 작성하라. */
#include <iostream>
```

```

#include <stdio.h>
int main() {
    float middle_exam;
    float final_exam;
    float quiz;
    std::cout << "당신의 중간고사 점수는?";
    std::cin >> middle_exam;
    /*중간고사 기말고사 퀴즈는 점수 범위가 0-100까지의 실수로 기준을 잡았다
    따라서 로직을 입력된 숫자가 0-100사이의 실수일때만 정상적으로 넘어가고
    그렇지 않을때는 추적해서 다시 입력버퍼를 지운뒤 변수에 재입력하게끔 했다*/
    while (std::cin.fail() == 1 || middle_exam < 0 || middle_exam > 100) {
        std::cin.clear();
        std::cin.ignore(10000, '\n');
        std::cout << "다시 중간고사 점수를 입력해주세요";
        std::cin >> middle_exam;
    }
    std::cout << "당신의 퀴즈 점수는?";
    std::cin >> quiz;
    while (std::cin.fail() == 1 || quiz < 0 || quiz > 100) {
        std::cin.clear();
        std::cin.ignore(10000, '\n');
        std::cout << "다시 퀴즈 점수를 입력해주세요";
        std::cin >> quiz;
    }
    std::cout << "당신의 기말고사 점수는?";
    std::cin >> final_exam;
    while (std::cin.fail() == 1 || final_exam < 0 || final_exam > 100) {
        std::cin.clear();
        std::cin.ignore(10000, '\n');
        std::cout << "다시 기말고사 점수를 입력해주세요";
        std::cin >> final_exam;
    }
    //평균을 mean변수에 담아줬다.
    int mean;
    mean = (middle_exam + final_exam + quiz) / 3;
    std::cout << mean;
    printf("당신의 점수는\n");
    //코드 간결화를 위해서 삼항연산자로 많은 if문을 함축해서 한줄로 표현했다
    //필요한 로직이 한 if문에 점수를 학점으로 출력하는 것밖에 없기때문에
    //삼항연산자를 쓴 결과 매우 간단하게 표현이 됐다.
    (mean >= 90) ? printf("A") : (mean >= 80) ? printf("B") : (mean >= 70) ? printf("C") :
    (mean >= 60) ? printf("D") : printf("F");
    printf("입니다");
}

```

출력결과

>>>

당신의 중간고사 점수는?99

당신의 퀴즈 점수는?88

당신의 기말고사 점수는?81

89당신의 점수는

B입니다