



C프로그래밍

Lecture 08. 조건에 따른 흐름의 분기

동덕여자대학교
데이터사이언스 전공
권 범

목차

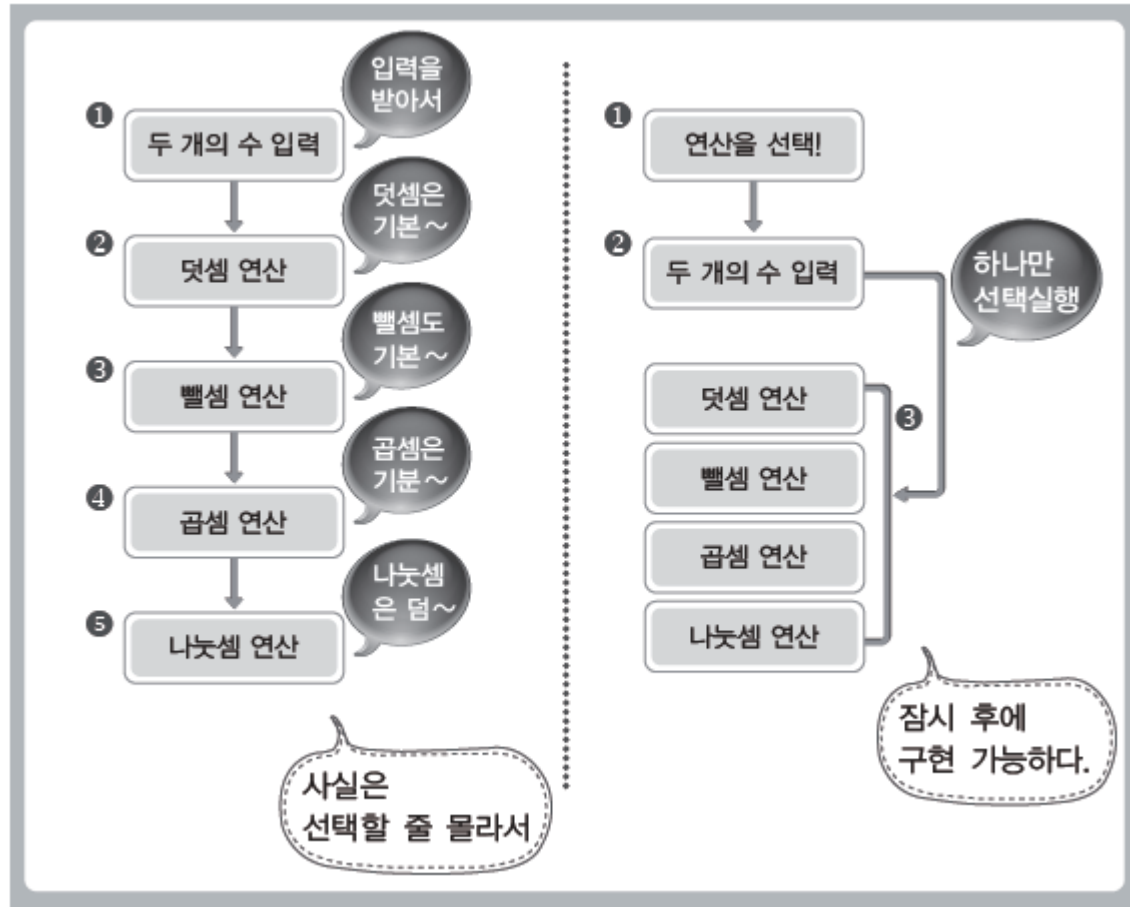
- ❖ 01. 조건적 실행과 흐름의 분기
- ❖ 02. 반복문의 생략과 탈출: continue & break
- ❖ 03. switch문에 의한 선택적 실행과 goto문
- ❖ 04. 연습 문제

01. 조건적 실행과 흐름의 분기

- 02. 반복문의 생략과 탈출: continue & break
- 03. switch문에 의한 선택적 실행과 goto문
- 04. 연습 문제

01. 조건적 실행과 흐름의 분기

❖ 흐름의 분기가 필요한 이유



프로그램을 구현하다 보면 상황에 따라서 선택적으로 실행해야 하는 영역도 존재하기 마련입니다!

분기하지 못하면 프로그램 사용자는 사칙연산 중 하나를 선택하지 못합니다.

01. 조건적 실행과 흐름의 분기

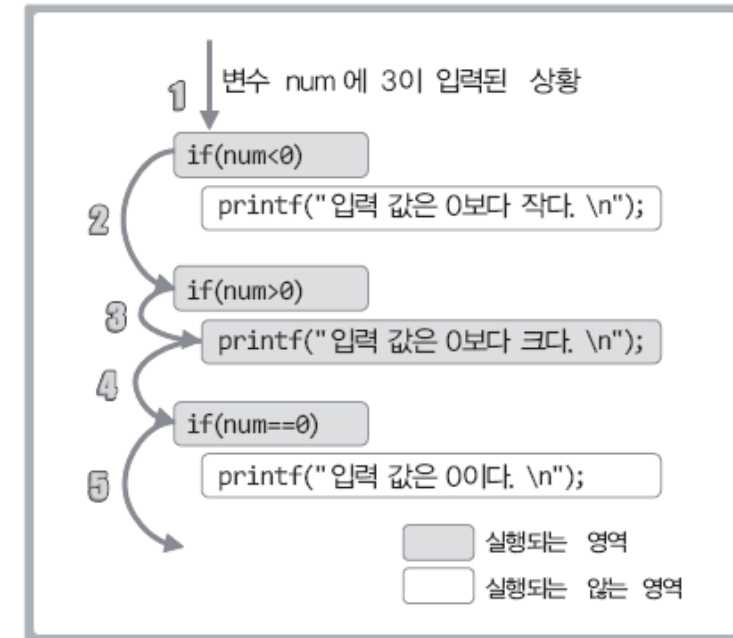
❖ if문을 이용한 조건적 실행 (1/2)

```
1  /* basic_if1.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4  int main(void)
5  {
6      int num;
7      printf("정수 입력: ");
8      scanf("%d", &num);
9
10     if (num < 0)    // num이 0보다 작으면 아래의 문장 실행
11         printf("입력 값은 0보다 작다.\n");
12     if (num > 0)    // num이 0보다 크면 아래의 문장 실행
13         printf("입력 값은 0보다 크다.\n");
14     if (num == 0)   // num이 0이면 아래의 문장 실행
15         printf("입력 값은 0이다.\n");
16
17     return 0;
18 }
```

정수 입력: 3
입력 값은 0보다 크다.

정수 입력: -2
입력 값은 0보다 작다.

정수 입력: 0
입력 값은 0이다.



01. 조건적 실행과 흐름의 분기

❖ if문을 이용한 조건적 실행 (2/2)

```
1  /* basic_if2.c */
2  #include <stdio.h>
3  int main(void)
4  {
5      int num1 = 22;
6      int num2 = 15;
7
8      if (num1 > num2) // num1이 num2보다 크면 실행
9      {
10         printf("num1이 num2보다 큽니다.\n");
11         printf("%d > %d\n", num1, num2);
12     }
13
14     return 0;
15 }
```

```
1  /* basic_if3.c */
2  #include <stdio.h>
3  int main(void)
4  {
5      int num1 = 22;
6      int num2 = 15;
7
8      if(num1 > num2) // num1이 num2보다 크면 실행
9          printf("num1이 num2보다 큽니다.\n");
10
11     return 0;
12 }
```

한 줄이면 중괄호 { }를 생략하는 것이 가능합니다.

01. 조건적 실행과 흐름의 분기

❖ if문을 이용한 계산기 프로그램 (1/2)

```
1  /* calculator_version1.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4  int main(void)
5  {
6      int opt;
7      double num1, num2;
8      double result;
9
10     printf("1. 덧셈 2. 뺄셈 3. 곱셈 4. 나눗셈\n");
11     printf("몇 번을 선택하시겠습니까?");
12     scanf("%d", &opt);
13     printf("두 개의 실수 입력: ");
14     scanf("%lf %lf", &num1, &num2);
```

```
15
16     if (opt == 1)
17         result = num1 + num2;
18     if (opt == 2)
19         result = num1 - num2;
20     if (opt == 3)
21         result = num1 * num2;
22     if (opt == 4)
23         result = num1 / num2;
24
25     printf("결과: %f\n", result);
26     return 0;
27 }
```

1. 덧셈 2. 뺄셈 3. 곱셈 4. 나눗셈
몇 번을 선택하시겠습니까? 3
두 개의 실수 입력: 2.14 5.12
결과: 10.956899

01. 조건적 실행과 흐름의 분기

❖ if문을 이용한 계산기 프로그램 (2/2)

```
1  /* calculator_version1.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4  int
5  {
6
7
8      double result;
9
10
11
12      scanf("%d", &opt);
13      printf("두 개의 실수 입력: ");
14      scanf("%lf %lf", &num1, &num2);
```

✓ 프로그램 구성상 사칙연산 중 하나만 실행이 됩니다.
✓ 그럼에도 불구하고 프로그램 사용자가 덧셈 연산을 선택할지라도 총 4번의 조건 검사(if문을 통한)를 진행한다는 비효율성이 존재합니다.

이러한 비효율성의 해결에 사용되는 것이 if~else문입니다.

```
15
16
17      result = num1 + num2;
18      if (opt == 2)
19          result = num1 - num2;
20      if (opt == 3)
21          result = num1 * num2;
22      if (opt == 4)
23          result = num1 / num2;
24
25      printf("결과: %f\n", result);
26      return 0;
27 }
```

1. 덧셈 2. 뺄셈 3. 곱셈 4. 나눗셈
몇 번을 선택하시겠습니까? 3
두 개의 실수 입력: 2.14 5.12
결과: 10.956899

01. 조건적 실행과 흐름의 분기

❖ if~else문을 이용한 흐름의 분기 (1/2)

```
1  /* basic_if_else1.c */
2  #include <stdio.h>
3  int main(void)
4  {
5      int num1 = 22;
6      int num2 = 15;
7
8      if (num1 > num2) // num1이 num2보다 크면 실행
9      {
10         printf("num1이 num2보다 큽니다.\n");
11         printf("%d > %d\n", num1, num2);
12     }
13     else // num1이 num2보다 크지 않으면 실행
14     {
15         printf("num1이 num2보다 크지 않습니다.\n");
16         printf("%d < %d\n", num1, num2);
17     }
18     return 0;
19 }
```

```
1  /* basic_if_else2.c */
2  #include <stdio.h>
3  int main(void)
4  {
5      int num1 = 22;
6      int num2 = 15;
7
8      if (num1 > num2)
9          printf("num1이 num2보다 큽니다.\n");
10     else
11         printf("num1이 num2보다 크지 않습니다.\n");
12
13     return 0;
14 }
```

한 줄이면 중괄호 { }를 생략하는 것이 가능합니다.

- else: (부사) (이미 언급된 것에 덧붙여) 그 밖의 다른 (부사) 다른

01. 조건적 실행과 흐름의 분기

❖ if~else문을 이용한 흐름의 분기 (2/2)

```
1  /* basic_if_else3.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int num;
8      printf("정수 입력: ");
9      scanf("%d", &num);
10     if (num < 0)
11         printf("입력 값은 0보다 작다.\n");
12     else
13         printf("입력 값은 0보다 작지 않다.\n");
14
15     return 0;
16 }
```

정수 입력: -2
입력 값은 0보다 작다.

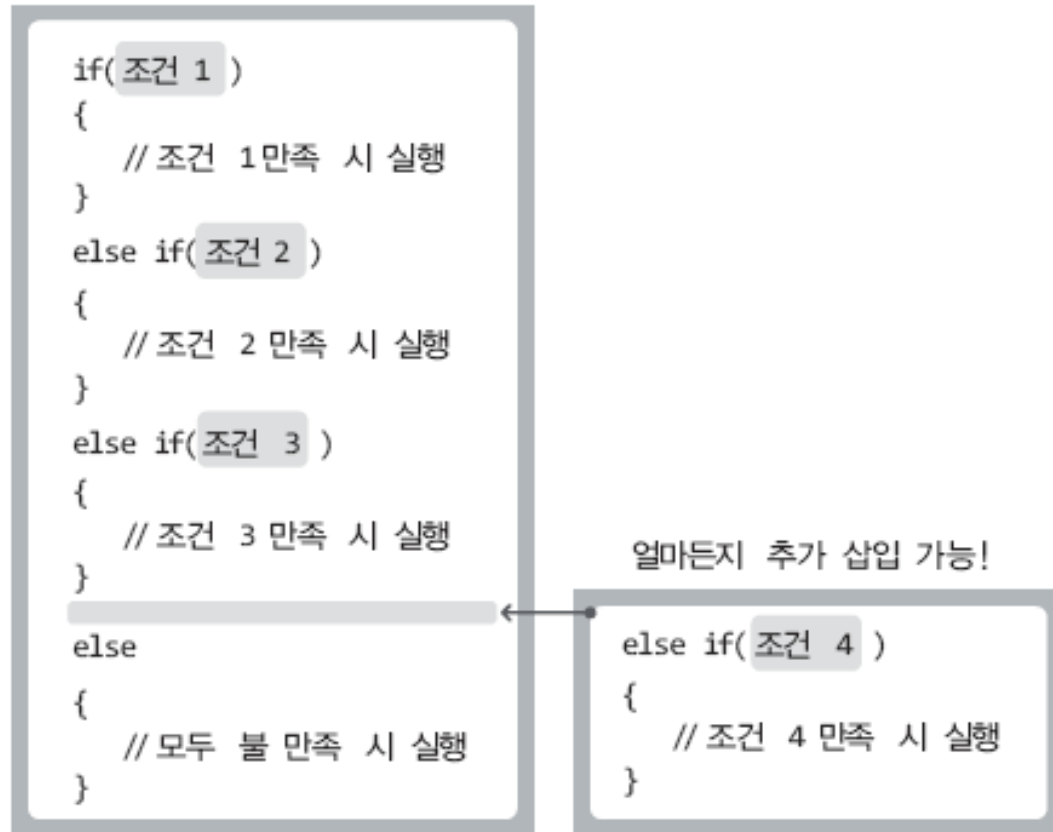
정수 입력: 3
입력 값은 0보다 작지 않다.

**if~else문은 하나의 문장이기 때문에
if와 else 사이에 다른 문장이 삽입될 수 없습니다.**

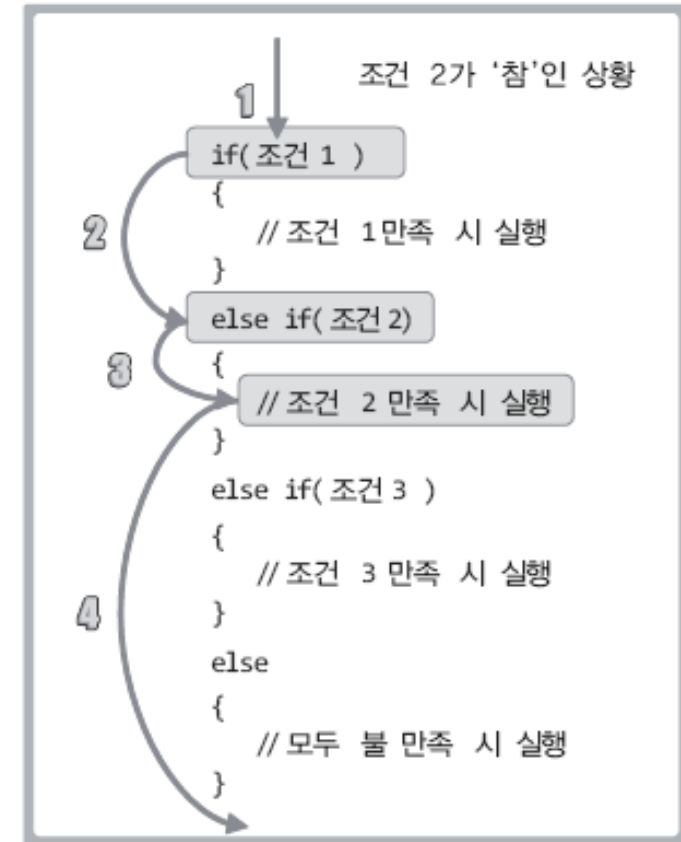
01. 조건적 실행과 흐름의 분기

❖ if ... else if ... else문의 구성과 흐름

if ... else if ... else문의 구성



if ... else if ... else문의 흐름



01. 조건적 실행과 흐름의 분기

❖ if ... else if ... else문의 적용

```
1  /* calculator_version2.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4  int main(void)
5  {
6      int opt;
7      double num1, num2;
8      double result;
9
10     printf("1. 덧셈 2. 뺄셈 3. 곱셈 4. 나눗셈\n");
11     printf("몇 번을 선택하시겠습니까?");
12     scanf("%d", &opt);
13     printf("두 개의 실수 입력: ");
14     scanf("%lf %lf", &num1, &num2);
```

```
15
16     if (opt == 1)
17         result = num1 + num2;
18     else if (opt == 2)
19         result = num1 - num2;
20     else if (opt == 3)
21         result = num1 * num2;
22     else
23         result = num1 / num2;
24
25     printf("결과: %f\n", result);
26     return 0;
27 }
```

1. 덧셈 2. 뺄셈 3. 곱셈 4. 나눗셈
몇 번을 선택하시겠습니까? 3
두 개의 실수 입력: 2.14 5.12
결과: 10.956899


01. 조건적 실행과 흐름의 분기

❖ if ... else if ... else문의 진실

```
if (num < 0)
    printf("입력 값은 0보다 작다.\n");
else if (num > 0)
    printf("입력 값은 0보다 크다.\n");
else
    printf("입력 값은 0이다.\n");
```

```
if (num < 0)
    printf("입력 값은 0보다 작다.\n");
else
{
    if (num > 0)
        printf("입력 값은 0보다 크다.\n");
    else
        printf("입력 값은 0이다.\n");
}
```

if~else문은 하나의 문장임을 생각하세요!



```
if(num<0)
    printf("입력 값은 0보다 작다. \n");
else
    if(num>0)
        printf("입력 값은 0보다 크다. \n");
    else
        printf("입력 값은 0이다. \n");
```

- ✓ else에 하나의 if~else문이 속한 상황입니다.
- ✓ 속한 문장이 하나일 때에는 중괄호 { }를 생략할 수 있습니다.

01. 조건적 실행과 흐름의 분기

❖ 조건 연산자: 피연산자가 세 개인 삼항 연산자 (1/2)

(조건) ? data1 : data2

조건이 참이면 data1 반환,
거짓이면 data2를 반환합니다.

```
int num3;  
  
if (num1 > num2)  
    num3 = num1;  
else  
    num3 = num2;
```




```
int num3 = (num1 > num2) ? (num1) : (num2);
```

01. 조건적 실행과 흐름의 분기

❖ 조건 연산자: 피연산자가 세 개인 삼항 연산자 (2/2)

```
1  /* absolute_value.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int num, abs;
8
9      printf("정수 입력: ");
10     scanf("%d", &num);
11
12     abs = (num > 0) ? num : num * (-1);
13     printf("절댓값: %d\n", abs);
14
15     return 0;
16 }
```



정수 입력: -15
절댓값: 15

02. 반복문의 생략과 탈출: continue & break

- 01. 조건적 실행과 흐름의 분기
- 03. switch문에 의한 선택적 실행과 goto문
- 04. 연습 문제

02. 반복문의 생략과 탈출: continue & break

❖ break! 이제 그만 빠져나가자!

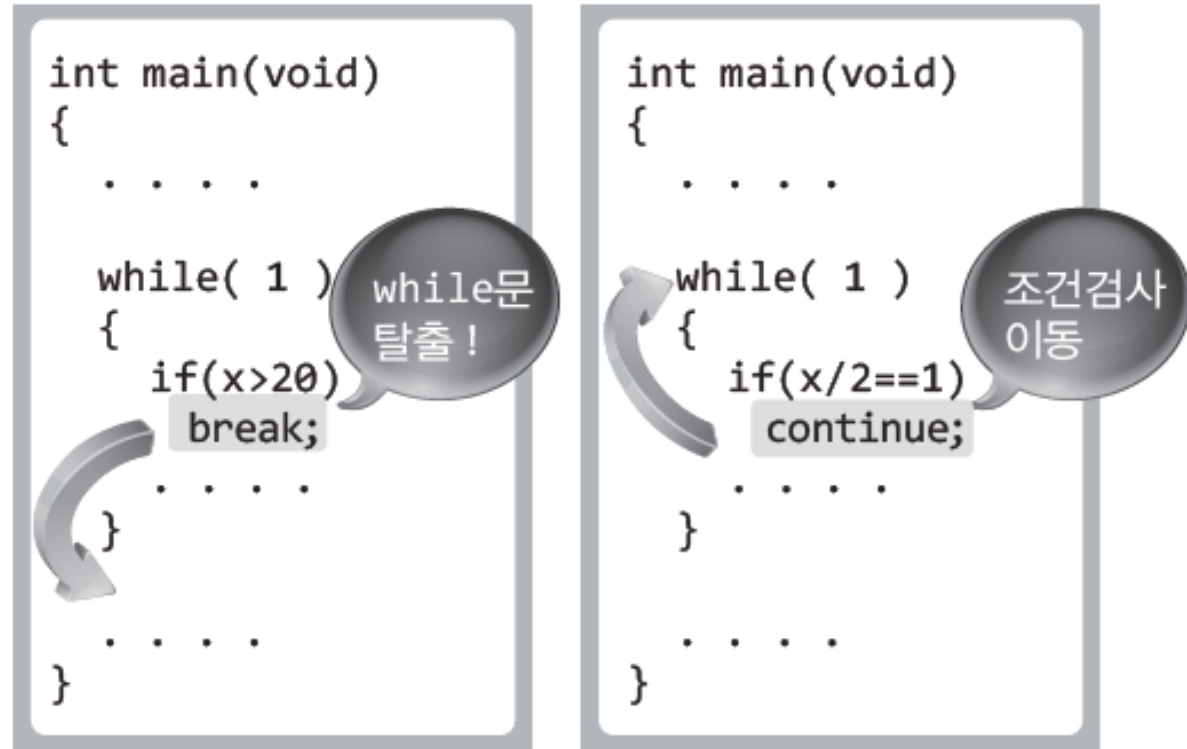
```
1  /* until_sum.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int sum = 0, num = 0;
7
8      while (1)
9      {
10         sum += num;
11         if (sum > 5000)
12             break;    // break문 실행! 반복문 탈출!
13         num++;
14     }
15
16     printf("sum: %d\n", sum);
17     printf("num: %d\n", num);
18     return 0;
19 }
```

- ✓ break문은 자신을 감싸는 반복문 하나를 빠져 나갑니다.
- ✓ if문과 함께 사용이 되어서 특정 조건이 만족될 때, 반복문을 빠져나가는 용도로 주로 사용됩니다.

sum: 5050
num: 100

02. 반복문의 생략과 탈출: continue & break

❖ continue! 나머지 생략하고 반복조건 확인하러 (1/2)




- `continue`
(동사) (쉬지 않고) 계속되다
(동사) (쉬지 않고) 계속하다

- ✓ `continue`문은 반복문을 빠져나가지 않습니다.
- ✓ 다만 반복조건을 확인하러 올라갈 뿐입니다.
- ✓ 그리고 반복조건이 여전히 참(True)이라면 반복영역을 처음부터 실행하게 됩니다.

02. 반복문의 생략과 탈출: continue & break

❖ continue! 나머지 생략하고 반복조건 확인하러 (2/2)

```
1  /* skip_continue.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int num;
7      printf("start! ");
8
9      for (num = 1; num < 20; num++)
10     {
11         if (num % 2 == 0 || num % 3 == 0)
12             continue;
13         printf("%d ", num);
14     }
15
16     printf("end!\n");
17     return 0;
18 }
```



start! 1 5 7 11 13 17 19 end!

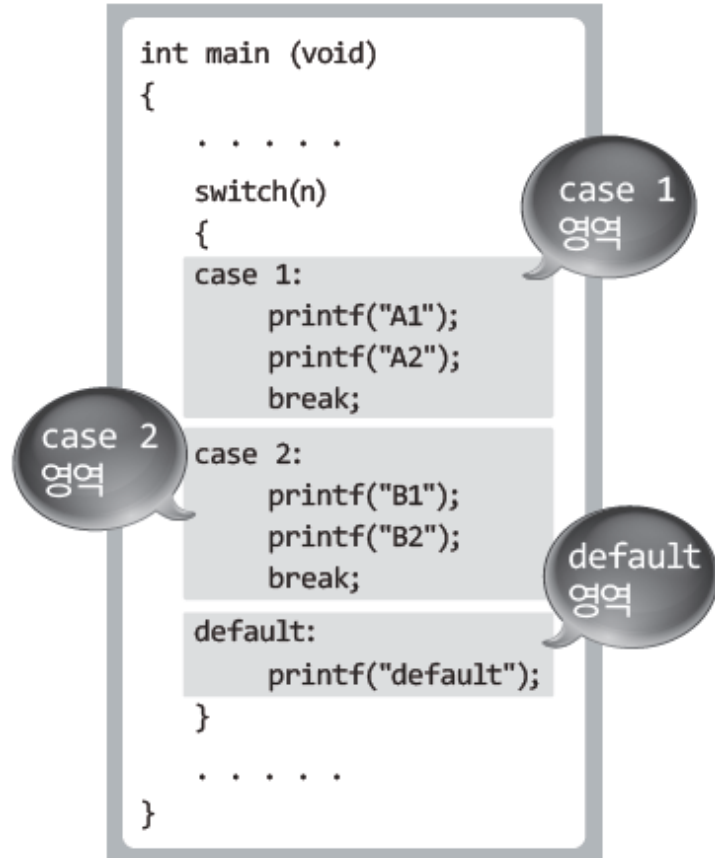
03. switch문에 의한 선택적 실행과 goto문

- 01. 조건적 실행과 흐름의 분기
- 02. 반복문의 생략과 탈출: continue & break
- 04. 연습 문제

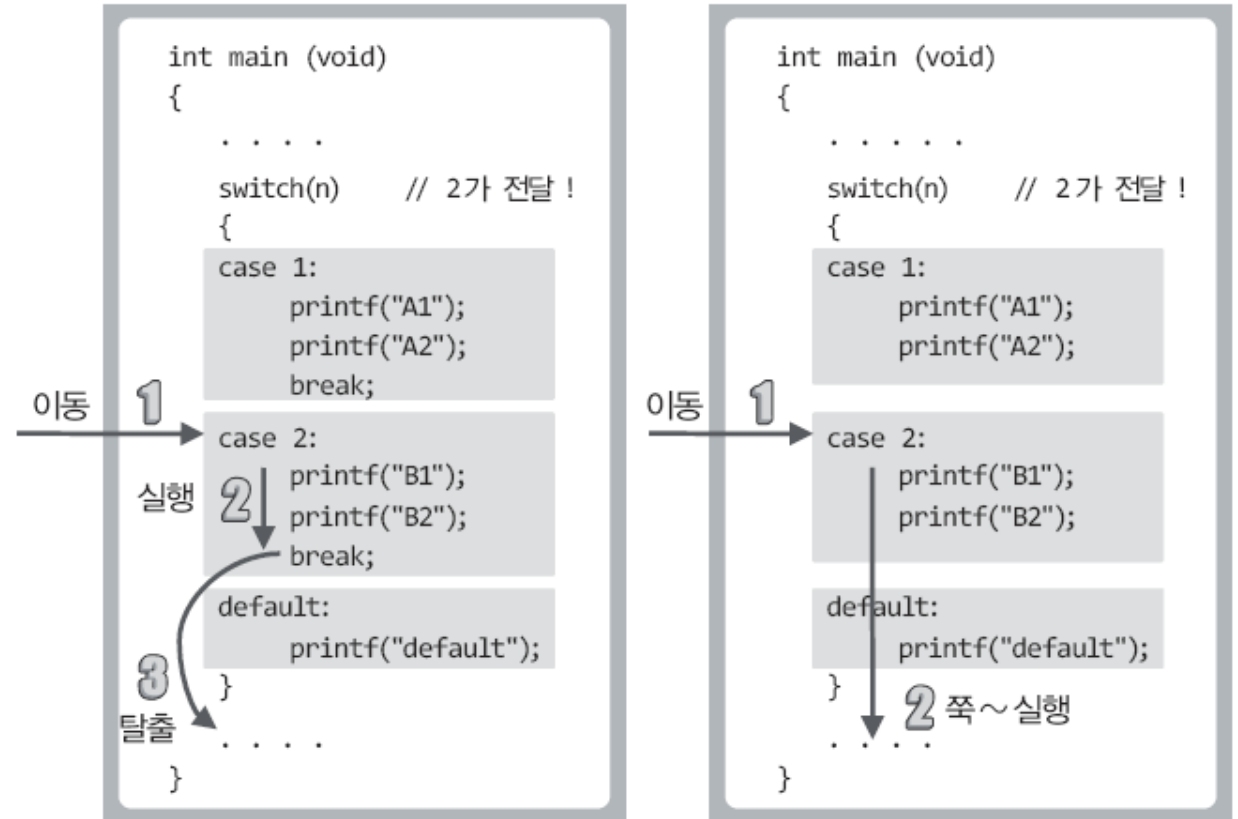
03. switch문에 의한 선택적 실행과 goto문

❖ switch문의 구성과 기본기능

switch문의 구성



삽입되어 있는 break문이 갖는 의미



03. switch문에 의한 선택적 실행과 goto문

❖ switch문 관련 예제

```
1  /* english_school.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4  int main(void)
5  {
6      int num;
7      printf("1이상 5이하의 정수 입력: ");
8      scanf("%d", &num);
9
10     switch (num)
11     {
12     case 1:
13         printf("1은 ONE\n");
14         break;
15     case 2:
16         printf("2는 TWO\n");
17         break;
```

```
18     case 3:
19         printf("3은 THREE\n");
20         break;
21     case 4:
22         printf("4는 FOUR\n");
23         break;
24     case 5:
25         printf("5는 FIVE\n");
26         break;
27     default:
28         printf("I don't know!\n");
29     }
30     return 0;
31 }
```

1이상 5이하의 정수 입력: 3
3은 THREE

03. switch문에 의한 선택적 실행과 goto문

❖ break문을 생략한 형태의 switch문 구성 (1/2)

```
1  /* break_switch.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4  int main(void)
5  {
6      char sel;
7      printf("M 오전, A 오후, E 저녁\n");
8      printf("입력: ");
9      scanf("%c", &sel);
10
11     switch (sel)
12     {
13     case 'M':
14     case 'm':
15         printf("Morning\n");
16         break;
```

```
17     case 'A':
18     case 'a':
19         printf("Afternoon\n");
20         break;
21     case 'E':
22     case 'e':
23         printf("Evening\n");
24         break;    // 사실 불필요한 break문!
25     }
26     return 0;
27 }
```

M 오전, A 오후, E 저녁
입력: M
Morning

03. switch문에 의한 선택적 실행과 goto문

❖ break문을 생략한 형태의 switch문 구성 (2/2)

```
1  /* break_switch.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4  int main(void)
5  {
6      char sel;
7      printf("M 오전, A 오후, E 저녁\n");
8      printf("입력: ");
9      scanf("%c", &sel);
10
11     switch (sel)
12     {
13     case 'M':
14     case 'm':
15         printf("Morning\n");
16         break;
```

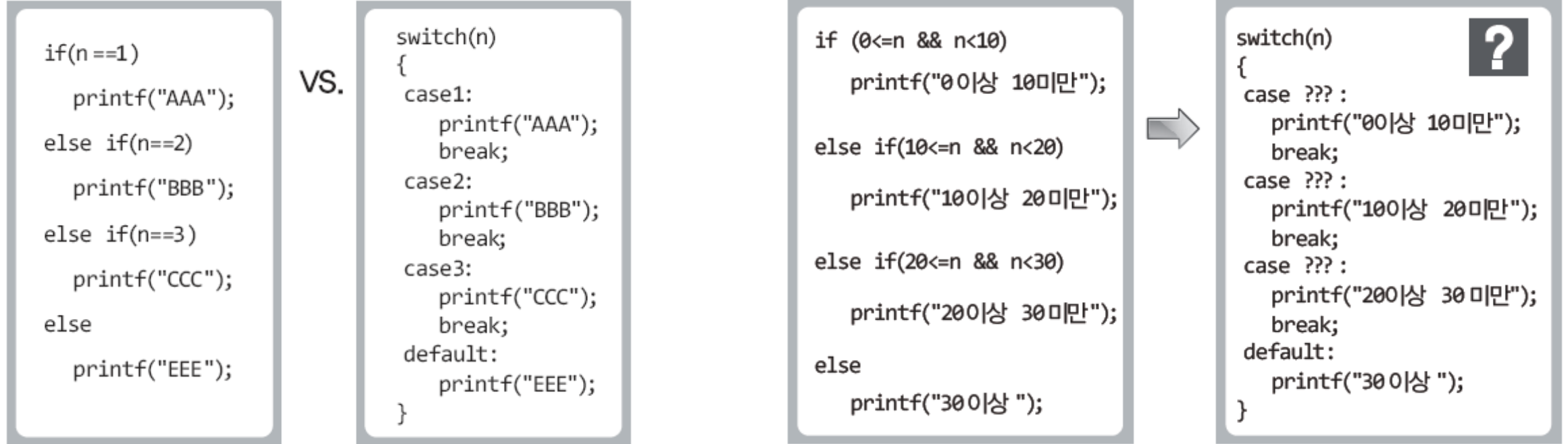
```
17     case 'A':
18     case 'a':
19         printf("Afternoon\n");
20         break;
21     case 'E':
22     case 'e':
23         printf("Evening\n");
24         break;    // 사실 불필요한 break문!
25     }
```

본 예제와 같은 경우, 다음과 같이
두 case 레이블을 한 줄에 같이 표시하기도 합니다.

```
case 'M': case 'm':
...
case 'A': case 'a':
...
case 'E': case 'e':
...
```


03. switch문에 의한 선택적 실행과 goto문

❖ switch vs. if ... else if ... else



if ... else if ... else문보다 switch문을 선호합니다.
switch문이 더 간결해 보이기 때문입니다.

모든 if ... else if ... else문을
switch문으로 대체할 수 있는 것은 아닙니다.

03. switch문에 의한 선택적 실행과 goto문

❖ 마지막으로 goto에 대해서 소개합니다.

```
int main(void)
{
    ...
    rabbit:           // 위치를 표시하는 rabbit 레이블
    ...
    goto rabbit;      // 레이블 rabbit으로 무조건 이동!
    ...
}
```

**goto는 프로그램의 흐름을
복잡하게 만드는 단점이 있습니다.**

이해는 하되, 가급적이면 활용은 하지 마세요.

03. switch문에 의한 선택적 실행과 goto문

❖ goto문의 사용법

```
1  /* goto.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4  int main(void)
5  {
6      int num;
7      printf("자연수 입력: ");
8      scanf("%d", &num);
9
10     if (num == 1)
11         goto ONE;
12     else if (num == 2)
13         goto TWO;
14     else
15         goto OTHER;
16
```

```
17 ONE:
18     printf("1을 입력하셨습니다!\n");
19     goto END;
20 TWO:
21     printf("2를 입력하셨습니다!\n");
22     goto END;
23 OTHER:
24     printf("3 혹은 다른 값을 입력하셨습니다!\n");
25     goto END;
26 END:
27     return 0;
28 }
```

자연수 입력: 2
2를 입력하셨습니다!

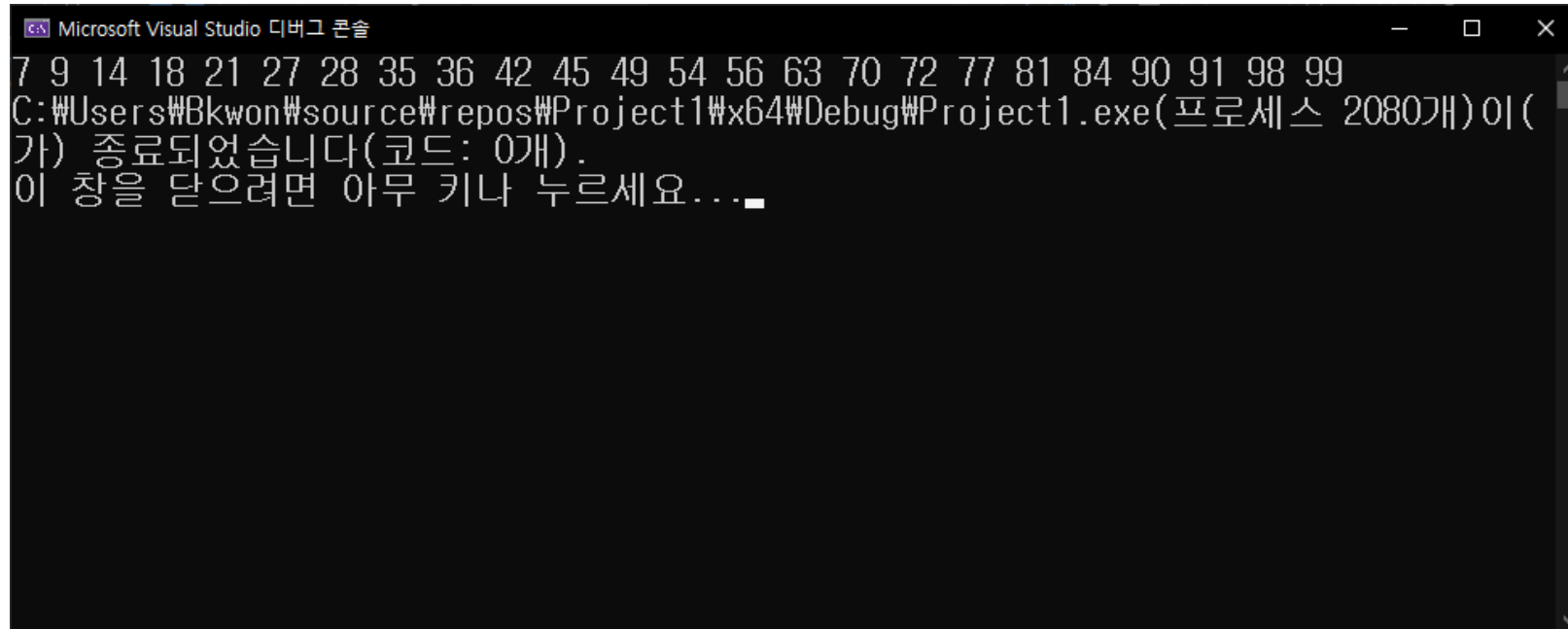
04. 연습 문제

- 01. 조건적 실행과 흐름의 분기
- 02. 반복문의 생략과 탈출: continue & break
- 03. switch문에 의한 선택적 실행과 goto문

04. 연습 문제

❖ (if문) 연습 문제 1.

- 1부터 99사이에 있는 정수 중에서 7의 배수이거나 9의 배수인 정수를 출력하는 프로그램을 작성해 보세요. 단, 7의 배수이자 동시에 9의 배수인 수는 한 번만 출력해야 합니다.



```
Microsoft Visual Studio 디버그 콘솔
7 9 14 18 21 27 28 35 36 42 45 49 54 56 63 70 72 77 81 84 90 91 98 99
C:\Users\Bkwon\source\repos\Project1\Debug\Project1.exe(프로세스 2080개) 이(
가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요..._
```

04. 연습 문제

❖ (if문) 연습 문제 1. 정답 및 해설

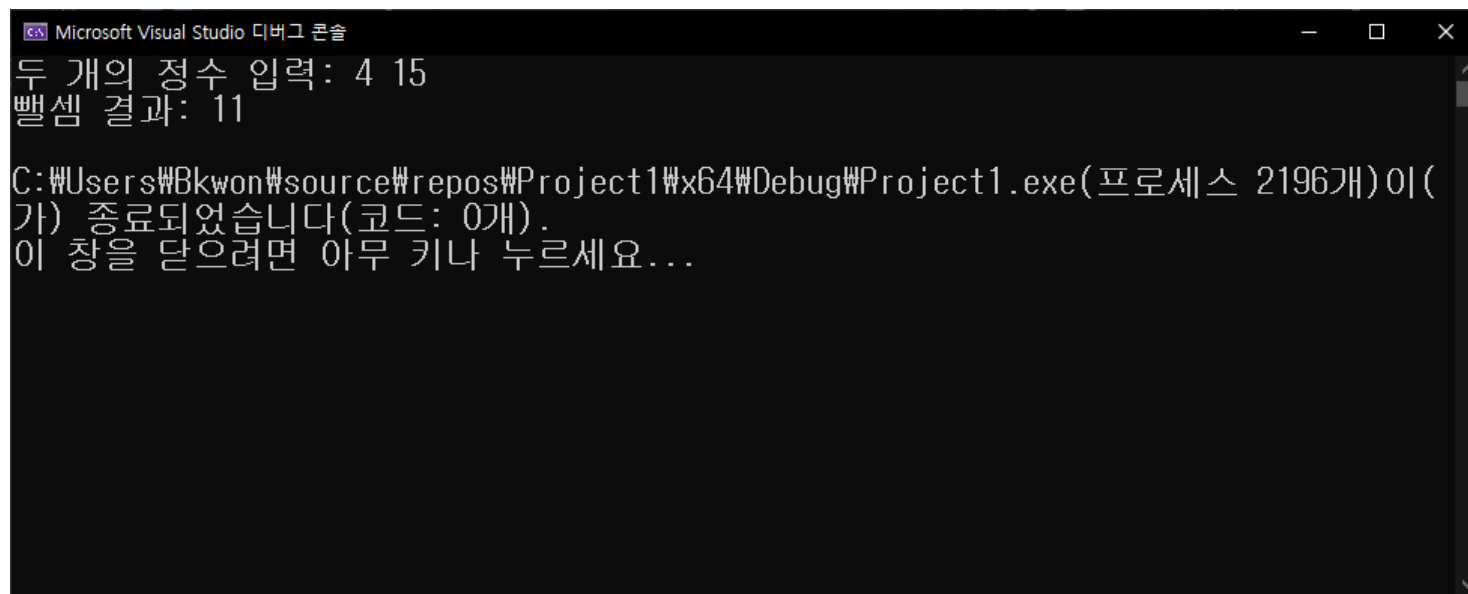
```
1  /* example1.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int j;
7
8      for (j = 1; j < 100; j++)
9      {
10         if ((j % 7 == 0) || (j % 9 == 0))
11             printf("%d ", j);
12     }
13
14     return 0;
15 }
```

7 9 14 18 21 27 28 35 36 42 45 49 54 56 63 70 72 77 81 84 90 91 98 99

04. 연습 문제

❖ (if문) 연습 문제 2.

- 두 개의 정수를 입력받아서 두 수의 차를 계산하는 프로그램을 구현해 보세요.
단, 무조건 큰 수에서 작은 수를 뺀 결과를 출력해 줘야 합니다.
예를 들어, 입력된 두 개의 숫자가 순서대로 12, 5라면 $12 - 5$ 를 계산해야 하고,
4, 16이라면 $16 - 4$ 를 계산해야 합니다. 따라서 출력 결과는 무조건 양수가 될 것입니다.



```
Microsoft Visual Studio 디버그 콘솔
두 개의 정수 입력: 4 15
뺄셈 결과: 11

C:\Users\Bkwon\source\repos\Project1\x64\Debug\Project1.exe(프로세스 2196개)이(
가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

04. 연습 문제

❖ (if문) 연습 문제 2. 정답 및 해설

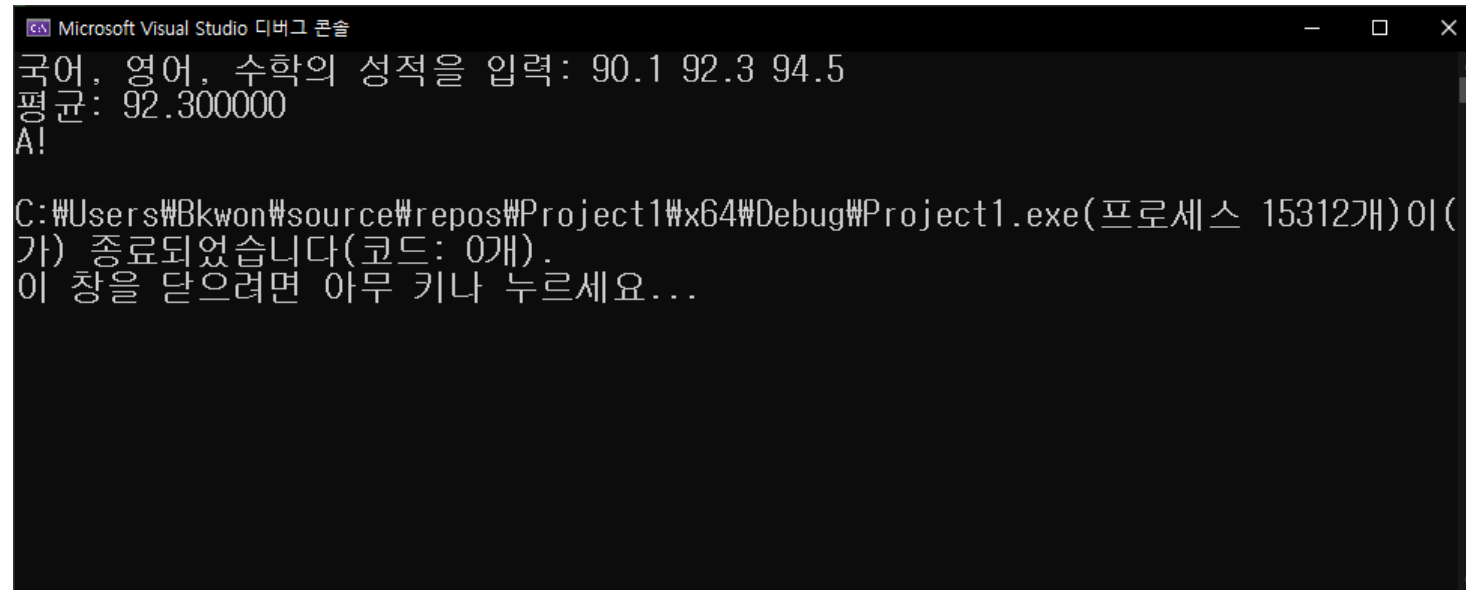
```
1  /* example2.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int a, b, result;
8
9      printf("두 개의 정수 입력: ");
10     scanf("%d %d", &a, &b);
11
12     if (a < b)
13         result = b - a;
14     else if (a > b)
15         result = a - b;
16
17     printf("뺄셈 결과: %d\n", result);
18     return 0;
19 }
```

두 개의 정수 입력 : 4 15
뺄셈 결과: 11

04. 연습 문제

❖ (if문) 연습 문제 3.

- 학생의 최종 학점을 계산하는 프로그램을 작성해 보세요. 학생의 성적이 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 50점 이상이면 D이고 그 미만이면 F입니다. 프로그램 실행 시 순서대로 국어, 영어, 수학의 점수를 입력받습니다. 그리고 평균을 구한 다음 그에 적절한 학점을 출력해 주면 됩니다.



```
Microsoft Visual Studio 디버그 콘솔
국어, 영어, 수학의 성적을 입력: 90.1 92.3 94.5
평균: 92.300000
A!

C:\Users\Bkwon\source\repos\Project1\x64\Debug\Project1.exe(프로세스 15312개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

04. 연습 문제

❖ (if문) 연습 문제 3. 정답 및 해설

```
1  /* example3.c */
2  #define _CRT_SECURE_NO_WARNINGS
3  #include <stdio.h>
4
5  int main(void)
6  {
7      double kor, eng, math;
8      double avg;
9
10     printf("국어, 영어, 수학의 성적을 입력: ");
11     scanf("%lf %lf %lf", &kor, &eng, &math);
12
13     avg = (kor + eng + math) / 3;
14     printf("평균: %f\n", avg);
```

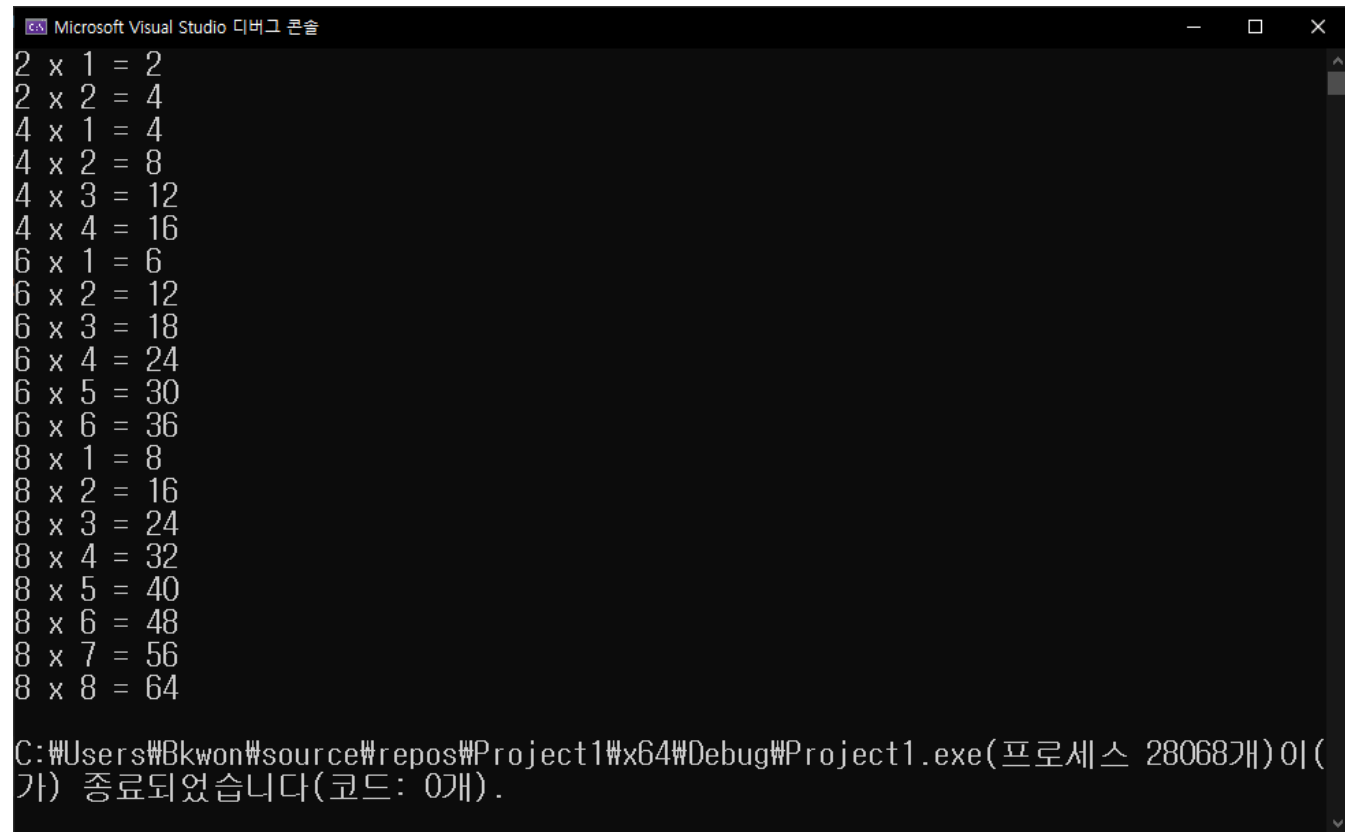
```
15     if (avg >= 90.0)
16         printf("A!\n");
17     else if (avg >= 80.0)
18         printf("B!\n");
19     else if (avg >= 70.0)
20         printf("C!\n");
21     else if (avg >= 50.0)
22         printf("D!\n");
23     else
24         printf("F!\n");
25
26     return 0;
27 }
```

국어, 영어, 수학의 성적을 입력: 90.1 92.3 94.5
평균: 92.300000
A!

04. 연습 문제

❖ (continue & break문) 연습 문제 4.

- 구구단을 출력하되 짝수단(2, 4, 6, 8단)만 출력하게끔 프로그램을 작성해 보세요.
또한, 2단은 2×2까지, 4단은 4×4까지, 6단은 6×6까지, 8단은 8×8까지만 출력하게끔 프로그램을 구현해 보세요.




```
Microsoft Visual Studio 디버그 콘솔
2 x 1 = 2
2 x 2 = 4
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
C:\Users\Bkwon\source\repos\Project1\wx64\Debug\Project1.exe(프로세스 28068개)이(가) 종료되었습니다(코드: 0개).
```

04. 연습 문제

❖ (continue & break문) 연습 문제 4. 정답 및 해설

```
1  /* example4.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int dan, j;
7      for (dan = 2; dan < 10; dan++)
8      {
9          if (dan % 2 != 0)
10             continue;
11          for (j = 1; j < 10; j++)
12          {
13              if (dan < j)
14                  break;
15              printf("%d x %d = %d\n", dan, j, dan * j);
16          }
17      }
18      return 0;
19 }
```



```
2 x 1 = 2
2 x 2 = 4
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
```

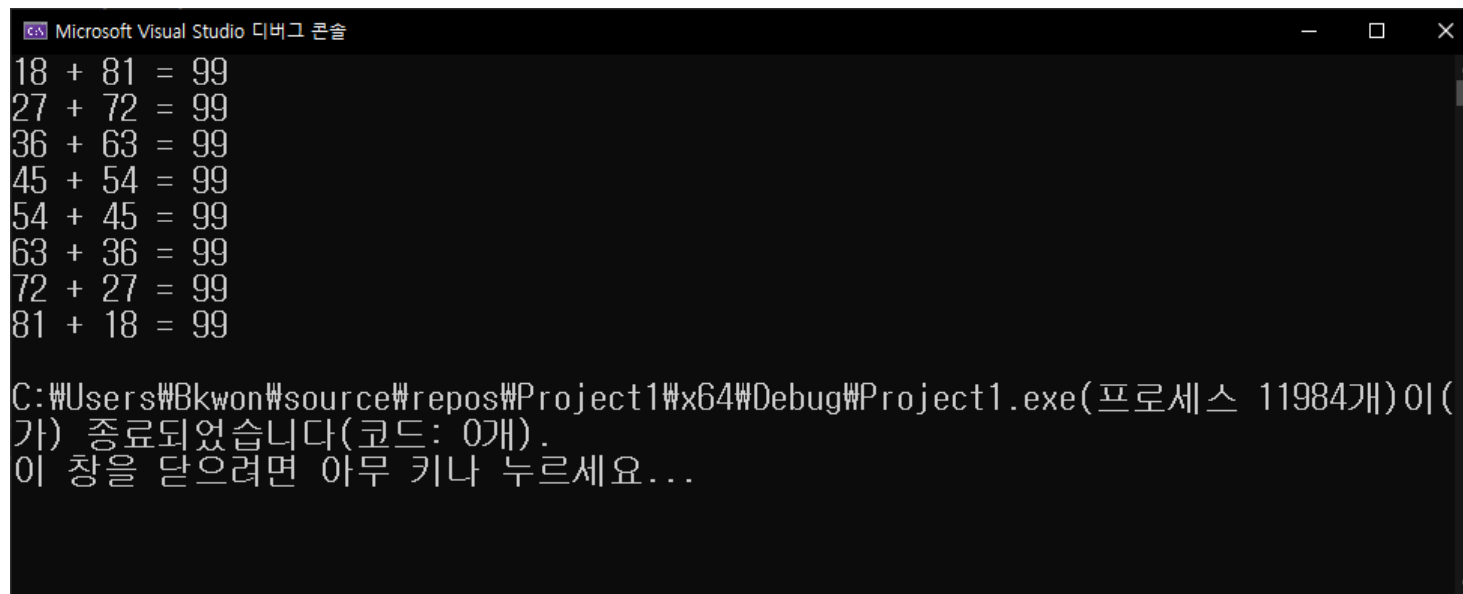
04. 연습 문제

❖ (continue & break문) 연습 문제 5.

- 다음 식을 만족하는 모든 a와 b를 구하는 프로그램을 구현해 보세요.

$$\begin{array}{r} ab \\ +ba \\ \hline 99 \end{array}$$

가능한 모든 수의 조합을 시도해 보기 위해서, for문과 같은 반복문을 이중으로 중첩시켜야 합니다.
출력 결과의 예로는 18+81, 27+72, 36+63, 45+54 등이 있습니다.




```
Microsoft Visual Studio 디버그 콘솔
18 + 81 = 99
27 + 72 = 99
36 + 63 = 99
45 + 54 = 99
54 + 45 = 99
63 + 36 = 99
72 + 27 = 99
81 + 18 = 99

C:\Users\Bkwon\source\repos\Project1\Debug\Project1.exe(프로세스 11984개)이(
가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

04. 연습 문제

❖ (continue & break문) 연습 문제 5. 정답 및 해설

```
1  /* example5.c */
2  #include <stdio.h>
3  int main(void)
4  {
5      int a, b;
6      int result;
7      for (a = 1; a < 10; a++)
8      {
9          for (b = 1; b < 10; b++)
10         {
11             if (a == b)
12                 continue;
13             result = (a * 10 + b) + (b * 10 + a);
14             if (result == 99)
15                 printf("%d%d + %d%d = %d\n", a, b, b, a, result);
16         }
17     }
18     return 0;
19 }
```



18 + 81 = 99
27 + 72 = 99
36 + 63 = 99
45 + 54 = 99
54 + 45 = 99
63 + 36 = 99
72 + 27 = 99
81 + 18 = 99

- ❖ 01. 조건적 실행과 흐름의 분기
- ❖ 02. 반복문의 생략과 탈출: continue & break
- ❖ 03. switch문에 의한 선택적 실행과 goto문
- ❖ 04. 연습 문제

THANK YOU!

Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: bkwon@dongduk.ac.kr