

REPORT



과 목 명 : 프로그래밍

담당교수 : 이원진 교수님

소 속 : 소프트웨어학과

학 번 : 32151671

이 름 : 박민혁



단국대학교
Dankook University

프로그래밍 제 1차 과제

I. 과제 개요

1. 과제내용

- 강의자료 제 4장 연산자 : 붉은색 예제 부분
- 강의자료 제 5장 조건문 : C Programming – 실습문제 8개(강의자료 뒤쪽 참고)

** 위 과제에 대하여 아래와 같이 작성하여 제출 바람

- 1) 모든 예제는 실습하고, 소스코드와 결과를 캡처 해서 문서로 작성하여 제출
- 2) 모든 예제에 대한 정확한 분석한 내용을 작성(전체 프로그램의 개요, 구조 동작 원리 등등)
- 3) 모든 예제에 대한 실습 및 결과 부분에 레포트 작성자가 누구인지 표기해야 함
(주석, printf문 등을 이용)

2. 과제범위

- 강의자료 제4장
 - * 예제 4-1, 4-2, 4-4, [복습] 증감 연산자의 이해(페이지 23~24), 예제 4-6, 예제 4-7, [복습] 관계 연산자 이해(페이지 38), 예제 4-11, 예제 4-12, 예제 4-16
- 강의자료 제5장
 - * 강의자료 뒤쪽에 별도 예제 8개

3. 과제 기본 개념

(1). 4장 - 연산자

연산자의 기본 개념

수식(expression) : 연산자와 피연산자의 조합

- 연산자(operator) : 연산에 사용되는 기호
- 피연산자(operand) : 연산의 대상이 되는 값 (변수나 상수)
- 모든 수식에는 반드시 값이 있음
- 수식의 평가 : 수식의 값을 구하는 것

피연산자의 개수로 분류한 연산자

종류	연산자의 의미	연산자	
단항 연산자	1개의 피연산자	+a -a a++ ++a a-- --a !a &a ~a sizeof a	
이항 연산자	2개의 피연산자	산술	a+b a-b a*b a/b a%b
		대입	a=b a+=b a-=b a*=b a/=b a%=b a>>=b a<<=b a&=b a =b a^=b
		관계	a>b a<b a>=b a<=b a==b a!=b
		논리	a&&b a b
		비트	a&b a b a^b a<<b a>>b
삼항 연산자	3개의 피연산자	a?b:c	

연산자의 기능에 따라 분류한 연산자

연산자의 종류	연산자
산술 연산자	a+b a-b a*b a/b a%b +a -a
증감 연산자	a++ ++a a-- --a
관계 연산자	a>b a<b a>=b a<=b a==b a!=b
논리 연산자	a&&b a b !a
비트 연산자	a&b a b a^b ~a a<<b a>>b
대입 연산자	a=b a+=b a-=b a*=b a/=b a%=b a&=b a =b a^=b a<<=b a>>=b
멤버 접근 연산자	*a &a a[b] a.b a->b
그 밖의 연산자	a?b:c a,b sizeof a (type)a

산술 연산자

산술 연산자	의미
+a	플러스(부호)
-a	마이너스(부호)
a + b	더하기
a - b	빼기
a * b	곱하기
a / b	나누기
a % b	나머지 구하기

복합 대입 연산자

복합 대입 연산자	의미	복합 대입 연산자	의미
$a += b$	$a = a + b$	$a \&= b$	$a = a \& b$
$a -= b$	$a = a - b$	$a = b$	$a = a b$
$a *= b$	$a = a * b$	$a ^= b$	$a = a ^ b$
$a /= b$	$a = a / b$	$a \ll= b$	$a = a \ll b$
$a \% = b$	$a = a \% b$	$a \gg= b$	$a = a \gg b$

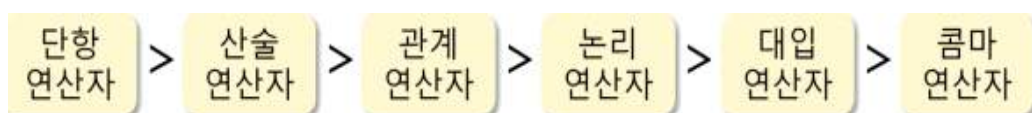
관계 연산자

관계 연산자	의미	a = 1, b = 2일 때 연산의 결과
$a > b$	a가 b보다 큰가?	0
$a \geq b$	a가 b보다 크거나 같은가?	0
$a < b$	a가 b보다 작은가?	1
$a \leq b$	a가 b보다 작거나 같은가?	1
$a == b$	a가 b와 같은가?	0
$a != b$	a가 b와 다른가?	1

논리 연산자

논리 연산자	부울 대수	의미
$a \&\& b$	논리 AND	a와 b가 둘 다 0이 아니면 1 a와 b중 하나만 0이면 0
$a b$	논리 OR	a와 b중 하나만 0이 아니면 1 a와 b가 둘 다 0이면 0
$! a$	논리 NOT	a가 0이면 1, a가 0이 아니면 0

연산자 우선 순위



비트 연산자

구분	비트 연산자	의미
비트 논리 연산자	$a \& b$	a와 b의 각 비트 단위로 논리 AND 연산
	$a b$	a와 b의 각 비트 단위로 논리 OR 연산
	$a \wedge b$	a와 b의 각 비트 단위로 논리 XOR 연산
	$\sim a$	a의 각 비트 단위로 논리 NOT 연산
비트 이동 연산자	$a \ll b$	a의 각 비트를 b개만큼 왼쪽으로 이동
	$a \gg b$	a의 각 비트를 b개만큼 오른쪽으로 이동

(2). 5장 - 조건문

제어문

- 프로그램의 수행 순서를 제어하기 위한 목적의 문장

제어문의 종류	C 구문	설명
조건문	if	조건식이 참이면 문장을 수행한다.
	switch	정수식의 값에 따라 수행할 문장을 선택한다.
반복문	for	조건식이 참인 동안 문장을 반복 수행한다.
	while	
	do while	
분기문	break	switch나 반복문을 빠져나간다.
	continue	반복문의 시작으로 돌아간다.
	goto	지정된 레이블의 문장으로 이동한다.
	return	함수를 호출한 곳으로 돌아간다.

중첩 if문

- if문 안에 포함된 if문

if ~ else if문

- 반복된 조건에 따른 선택(연속적인 if문)
- 조건문 if else문으로서, else 이후에 if else를 필요한 횟수만큼 반복 가능

다중 if

- 서로 독립적인 조건을 여러 개 비교하는 경우
- 각각의 if문은 else if로 연결되지 않는다.

switch문

- 정수식(조건식)의 값에 따라서 여러 가지 경우 중 하나로 분기(선택하는데 사용)



switch 이용 주의

- Keyword : switch 문에서 switch, case, break, default는 keyword이며, expression과 value는 반드시 정수나 정수 수식이어야 한다. 또한 case 다음의 value 값은 중복되어서는 안 된다.
- switch 문의 정확한 실행 순서는
 - (1). switch 문의 표현식을 평가
 - (2). 위 표현식에서 계산된 값과 일치하는 상수 값을 갖는 case의 값을 위에서부터 찾는다. 일치된 case 값을 만나면 case 내부의 문장을 실행
 - (3). break를 만나면 switch 문을 종료, 또는 switch 몸체와 마지막 문장을 수행하면 switch 문을 종료
 - (4). 일치된 case 값을 만나지 못하여 default를 만나면 default 내부의 문장을 실행

II. 과제 소스 코드와 결과

1. 예제 4-1 : 정수의 산술 연산.

// 예제 4-1 : 정수의 산술 연산. 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : a1sgur9784@naver.com

```
#include<stdio.h>

int main(void)
{
    int num1 = 0;           // 정수 num1 변수 초기화.
    int num2 = 0;           // 정수 num2 변수 초기화.

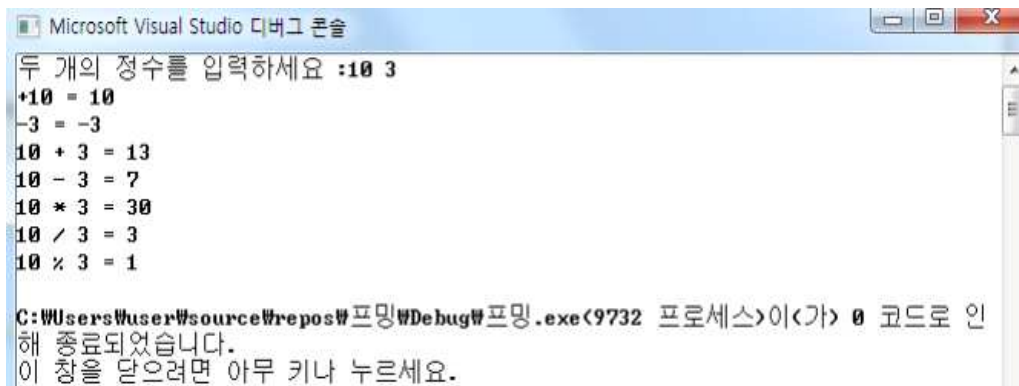
    printf("두 개의 정수를 입력하세요 :");
    scanf_s("%d %d", &num1, &num2);

    // 두개의 정수인 num1과 num2의 값 입력.

    printf("+%d = %d\n", num1, +num1);
    printf("-%d = %d\n", num1, -num2);
    printf("%d + %d = %d\n", num1, num2, num1 + num2);
    printf("%d - %d = %d\n", num1, num2, num1 - num2);
    printf("%d * %d = %d\n", num1, num2, num1*num2);
    printf("%d / %d = %d\n", num1, num2, num1 / num2);
    printf("%d %% %d = %d\n", num1, num2, num1%num2);

    // 입력 받은 정수 num1의 플러스 부호.
    // 입력 받은 정수 num2의 마이너스 부호.
    // 입력 받은 정수 num1과 num2의 합.
    // 입력 받은 정수 num1과 num2의 차.
    // 입력 받은 정수 num1과 num2의 곱.
    // 입력 받은 정수 num1과 num2의 나눗셈 몫.
    // 입력 받은 정수 num1과 num2의 나눗셈 나머지.

    return 0;
}
```



2. 예제 4-2 : 실수의 산술 연산.

// 예제 4-2 : 실수의 산술 연산. 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : a1sgur9784@naver.com

```
#include<stdio.h>

int main(void)
{
    double num1 = 0;        // 실수 num1 변수 초기화.
    double num2 = 0;        // 실수 num2 변수 초기화.

    printf("두 개의 실수를 입력하세요 :");
    scanf_s("%lf %lf", &num1, &num2);

    // 두개의 실수 num1과 num2의 값 입력.

    printf("+%f = %f\n", num1, +num1);
    printf("-%f = %f\n", num1, -num2);
    printf("%f + %f = %f\n", num1, num2, num1 + num2);
    printf("%f - %f = %f\n", num1, num2, num1 - num2);
    printf("%f * %f = %f\n", num1, num2, num1*num2);
    printf("%f / %f = %f\n", num1, num2, num1 / num2);
    // printf("%f %% %f = %f\n", num1, num2, num1%num2);
    // Compile Error.

    return 0;
}
```

```

Microsoft Visual Studio 디버그 콘솔

두 개의 실수를 입력하세요 :10 3
+10.000000 = 10.000000
-3.000000 = -3.000000
10.000000 + 3.000000 = 13.000000
10.000000 - 3.000000 = 7.000000
10.000000 * 3.000000 = 30.000000
10.000000 / 3.000000 = 3.333333

C:\Users\User\source\repos\프밍\Debug\프밍.exe(6632 프로세스)이<가> 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.

```

3. 예제 4-4 : 정수와 실수의 혼합 연산.

// 예제 4-4 : 정수와 실수의 혼합 연산, 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : alsur9784@naver.com

```

#include<stdio.h>

#define PI 3.141592 // 매크로 상수 PI 정의.

int main(void)
{
    int radius = 0; // 정수 radius(반지름) 변수 초기화.
    double area=0; // 실수 area(면적) 변수 초기화.
    double perimeter=0; // 실수 perimeter(둘레) 변수 초기화.

    printf("반지름? "); // 정수 radius(반지름) 입력.
    scanf_s("%d", &radius);

    area=PI*radius*radius; // PI가 이미 실수이므로 radius를 double형으로 변환해서 연산 해줌.
    printf("원의 면적 : %.2f\n", area);

    perimeter=2*PI*radius; // PI가 이미 실수이므로 radius를 double형으로 변환해서 연산 해줌.
    printf("원의 둘레: %.2f\n", perimeter);

    return 0;
}

```

```

Microsoft Visual Studio 디버그 콘솔

반지름? 5
원의 면적: 78.54
원의 둘레: 31.42

C:\Users\User\source\repos\프밍\Debug\프밍.exe(9712 프로세스)이<가> 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.

```


4. [복습] 증감 연산자의 이해.

```
// 증감 연산자의 이해, 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : a1sgur9784@naver.com

#include<stdio.h>

int main(void)
{
    int a = 10;           // 변수 a=10 선언.
    int b = 20;           // 변수 b=20 선언.
    int c = 30;           // 변수 c=30 선언.

    printf("a = %d, b = %d, c = %d\n", a, b, c);           // 초기 변수 a b c를 출력 한다.
    printf("a + b++의 결과 = %d\n", a + b++);              // b++은 문장이 실행된 후에 b++가 되므로 10+20 = 30 출력.
    printf("a = %d, b = %d, c = %d\n", a, b, c);           // b++ 되서 b=21.
    printf("a + ++b * 3의 결과 = %d\n", a + ++b * 3);        // 현재 a=10 b=21 c=30을 출력 한다.
    printf("a = %d, b = %d, c = %d\n", a, b, c);           // ++이 앞에 붙는 경우 연산자 우선 순위가 가장 높다. 그러므로 10 + (22) * 3 = 10 + 66 = 76.
    printf("a + ++b / 2 + c++의 결과 = %d\n", a + ++b / 2 + c++); // 현재 a=10 b=22 c=30을 출력 한다.
    printf("a = %d, b = %d, c = %d\n", a, b, c);           // ++이 앞에 붙는 경우 연산자 우선 순위가 가장 높다. 그리고 c++은 문장이 실행된 후에 c++가 되므로 30을 그대로 사용 한다.
    printf("a + ++b / 2 + c++의 결과 = %d\n", a + ++b / 2 + c++); // 그러므로 11 / 2 + 30 = 5 + 30 = 35.
    printf("a = %d, b = %d, c = %d\n", a, b, c);           // c++ 되서 c=31.
    printf("a + ++b * 3 + ++c의 결과 = %d\n", a + ++b * 3 + ++c); // 현재 a=11 b=22 c=31을 출력 한다.
    printf("a + ++b * 3 + ++c의 결과 = %d\n", a + ++b * 3 + ++c); // a++은 문장이 실행된 후에 a++이 되므로 a=11, ++이 앞에 붙는 경우는 문장이 실행되기 전 b=1를 해주므로 b=23, c도 마찬가지로 c=32.
    printf("a = %d, b = %d, c = %d\n", a, b, c);           // 따라서 11 + 23 + 32 = 66.
    printf("a = %d, b = %d, c = %d\n", a, b, c);           // a=1 실행, 따라서 a=12 b=23 c=32이다.

    return 0;
}
```

5. 예제 4-6 : 대입 연산식의 값.

```
// 예제 4-6 : 대입 연산식의 값, 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : a1sgur9784@naver.com

#include<stdio.h>

int main(void)
{
    int a = 0;           // 변수 a 변수 초기화.
    double b = 0;        // 실수 b 변수 초기화.
    int c = 0;           // 변수 c 변수 초기화.

    a = 123;             // a에 123 저장.

    printf("a=%d\n", a); // a=123 출력.
    printf("a=%d\n", a = 456); // a에 456 저장 후 a=456 출력.
    printf("b=%lf\n", b = a + 0.5); // b에 a + 0.5 저장 후 b=456.5 출력.
    printf("c=%d\n", c = printf("ABC\n")); // printf 함수의 return 값을 c에 저장하고 c=4 출력.

    return 0;
}
```

```
Microsoft Visual Studio 디버그 콘솔

a=123
a=456
b=456.500000
ABC
c=4

C:\Users\user\source\repos\프밍\Debug\프밍.exe(10020 프로세스)이<가> 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

6. 예제 4-7 : 복합 대입 연산자의 활용.

// 예제 4-7 : 복합 대입 연산자의 활용. 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : a1sgur9784@naver.com

```
#include<stdio.h>

int main(void)
{
    int items = 0;           // 정수 items(전체 항목 수) 변수 초기화.
    int pages = 0;          // 정수 pages 변수 초기화.
    int items_per_page = 0; // 정수 items_per_page(1 페이지 당 항목 수) 변수 초기화.

    printf("항목수 ? ");
    scanf_s("%d", &items); // 정수 items 입력.

    printf("한 페이지 당 항목수 ? ");
    scanf_s("%d", &items_per_page); // 정수 items_per_page 입력.

    pages = items / items_per_page; // 페이지 수.
    items %= items_per_page;        // 남은 항목 수.

    printf("%d page와 %d 항목\n", pages, items); // 나머지 연산의 결과를 다시 items에 저장하므로 따로 변수가 필요 없음.

    return 0;
}
```

```
Microsoft Visual Studio 디버그 콘솔

항목수 ? 55
한 페이지 당 항목수 ? 20
2 page와 15 항목

C:\Users\user\source\repos\프밍\Debug\프밍.exe(9828 프로세스)이<가> 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

7. [복습] 관계 연산자의 이해.

// 관계 연산자 이해. 작성자 : 소프트웨어학과 32151671 박민혁. 이메일 : a1sgur9784@naver.com

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

```
int main(void)
{
```

```
    int x = 10, y = 20;
    int r1 = 0, r2 = 0, r3 = 0, r4 = 0;
```

```
    r1 = (x == y);
    r2 = (x != y);
    r3 = (x >= y);
    r4 = (x <= y);
```

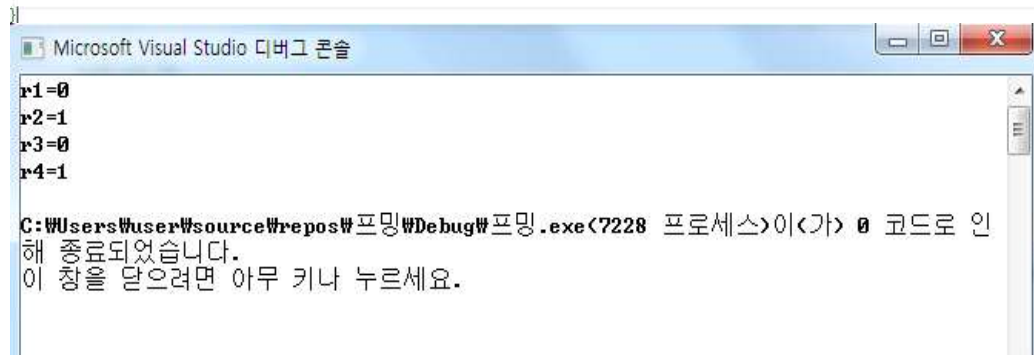
```
    printf("r1=%d\n", r1);
    printf("r2=%d\n", r2);
    printf("r3=%d\n", r3);
    printf("r4=%d\n", r4);
```

```
    return 0;
```

```
// 정수 x = 10 선언, 정수 y = 20 선언.
// 정수 r1 r2 r3 r4 초기화.
```

```
// r1에 x와 y가 같으면 1을 저장. 아니면 0을 저장.
// r2에 x와 y가 같지 않으면 1을 저장. 아니면 0을 저장.
// r3에 x가 y보다 크거나 같으면 1을 저장. 아니면 0을 저장.
// r4에 x가 y보다 작거나 같으면 1을 저장. 아니면 0을 저장.
```

```
// 출력.
```



8. 예제 4-11 : 비트 이동 연산자의 사용 예

// 예제 4-11 : 비트 이동 연산자의 사용 예 작성자 : 소프트웨어학과 32151671 박민혁. 이메일 : a1sgur9784@naver.com

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

```
int main(void)
{
```

```
    int x = 0x00000012;
    int y = x << 4;
    int z = x >> 4;
```

```
    printf("x=%#08x, %d\n", x, x);
    printf("x=%#08x, %d\n", y, y);
    printf("x=%#08x, %d\n", z, z);
```

```
    return 0;
```

```
// x를 16진수로 표현.
// y에 x의 비트를 왼쪽으로 4칸 옮김.
// z에 x의 비트를 오른쪽으로 4칸 옮김.
```

```
// 0x00000012 18.
// 0x00000120 288 (18 * 16).
// 0x00000001 1 (18 / 16).
```

```

Microsoft Visual Studio 디버그 콘솔

x=0x000012, 18
x=0x000120, 288
x=0x000001, 1

C:\Users\User\source\repos\프밍\Debug\프밍.exe(7156 프로세스)이(가) 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.

```

9. 예제 4-12 : 조건 연산자의 사용 예

// 예제 4-12 : 조건 연산자의 사용 예 작성자 : 소프트웨어학과 92151671 박민혁, 이메일 : alsur9784@naver.com

```

#include<stdio.h>

int main(void)
{
    int a = 0, b = 0;           // 정수 a b 변수 초기화.
    int result = 0;           // 결과 값을 저장 할 result 초기화.
    int max = 0;

    printf("2개의 정수 ? ");
    scanf_s("%d %d", &a, &b);   // 정수 a b 입력.

    printf("%d은 ", a);
    a % 2 != 0 ? printf("홀수") : printf("짝수");           // a % 2 != 0 이면 홀수, 아니면 짝수
    printf("입니다\n");

    result = a > 0 ? a : -a;           // a > 0 이면 a 값을 그대로 result에 저장, 아니면 -불러서 양의 정수 형태로 result에 저장.
    printf("a의 절대값 : %d\n", result);

    result = b > 0 ? b : -b;           // b > 0 이면 b 값을 그대로 result에 저장, 아니면 -불러서 양의 정수 형태로 result에 저장.
    printf("b의 절대값 : %d\n", result);

    max = a > b ? a : b;           // a > b 이면 max에 a 저장, 아니면 b 저장.
    printf("a,b 중 큰 값: %d\n", max);
    return 0;
}

```

```

Microsoft Visual Studio 디버그 콘솔

2개의 정수 ? 3 -5
3은 홀수입니다
a의 절대값 : 3
b의 절대값 : 5
a,b 중 큰 값: 3

C:\Users\User\source\repos\프밍\Debug\프밍.exe(8884 프로세스)이(가) 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.

```

10. 예제 4-16 : 연산자의 우선순위와 결합 규칙.

// 예제 4-16 : 연산자의 우선순위와 결합 규칙 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : a1sgur9784@naver.com

```
#include<stdio.h>

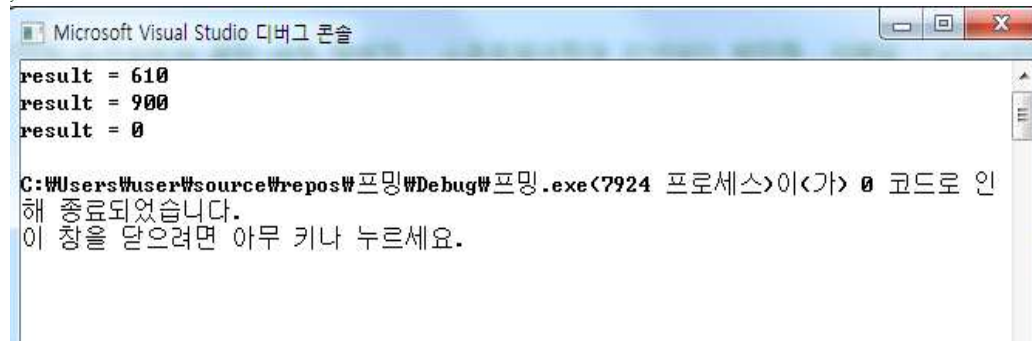
int main(void)
{
    int a = 10;           // 정수 a 변수에 10 저장.
    int b = 20;           // 정수 b 변수에 20 저장.
    int c = 30;           // 정수 c 변수에 30 저장.
    int result = 0;       // a b c 연산 결과를 저장 할 result 변수 초기화.

    result = a + b * c;    // 10 + 20 * 30 = 610을 result에 저장.
    printf("result = %d\n", result);

    result = (a + b)*c;    // (10 + 20) * 30 = 900을 result에 저장.
    printf("result = %d\n", result);

    result = a < b && c < 0; // a < b 이고, c < 0 이면 1을 result에 1을 저장, 아니면 0을 저장.
    printf("result = %d\n", result);

    return 0;
}
```



11. Example – 01

// Example – 01 : 연령에 따른 요금제 출력문, 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : a1sgur9784@naver.com

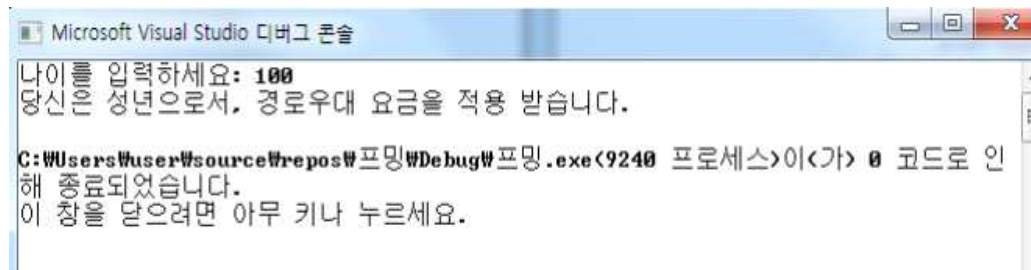
```
#include<stdio.h>

int main(void)
{
    int age = 0;           // 정수 age 변수 초기화.

    printf("나이를 입력하세요: ");
    scanf_s("%d", &age);   // 정수 age 입력.

    if (age < 18)           // age가 18보다 작으면.
        printf("당신은 미성년자로서, 청소년 요금을 적용 받습니다.\n"); // 미성년자 출력.
    else                   // 그 외.
    {
        printf("당신은 성년으로서, ");
        if (age < 65)       // age가 18보다 크거나 같고, 65보다 작으면.
            printf("일반 요금을 적용 받습니다.\n"); // 일반 요금.
        else               // age가 18보다 크거나 같고, 65보다 크면.
            printf("경로우대 요금을 적용 받습니다.\n"); // 경로우대 요금.
    }

    return 0;
}
```



```
Microsoft Visual Studio 디버그 콘솔
나이를 입력하세요: 100
당신은 성년으로서, 경로우대 요금을 적용 받습니다.

C:\Users\User\source\repos\프밍\Debug\프밍.exe(9240 프로세스)이<가> 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

12. Example – 02

// Example - 02 : 연령에 따른 요금제 출력문, 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : alsaur9784@naver.com

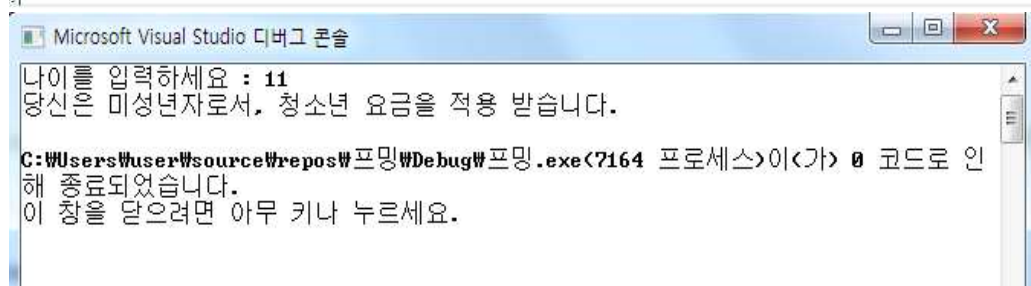
```
#include<stdio.h>

int main(void)
{
    int age = 0; // 정수 age 변수 초기화.

    printf("나이를 입력하세요 : ");
    scanf_s("%d", &age); // 정수 age 입력.

    if (age < 18) // age가 18보다 작으면.
        printf("당신은 미성년자로서, 청소년 요금을 적용 받습니다.\n"); // 미성년자 출력.
    else if (age < 65) // age가 18보다 크거나 같고, 65보다 작으면.
        printf("당신은 성년으로서, 일반 요금을 적용 받습니다.\n"); // 일반 요금.
    else // age가 18보다 크거나 같고, 65보다 크면.
        printf("당신은 성년으로서, 경로우대 요금을 적용 받습니다.\n"); // 경로우대 요금.

    return 0;
}
```



```
Microsoft Visual Studio 디버그 콘솔
나이를 입력하세요 : 11
당신은 미성년자로서, 청소년 요금을 적용 받습니다.

C:\Users\User\source\repos\프밍\Debug\프밍.exe(7164 프로세스)이<가> 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```


13. Example – 03

//Example - 03 : 달에 따른 계절 출력문. 작성자 : 소프트웨어학과 32151671 박민혁. 이메일 : aalsgur9784@naver.com

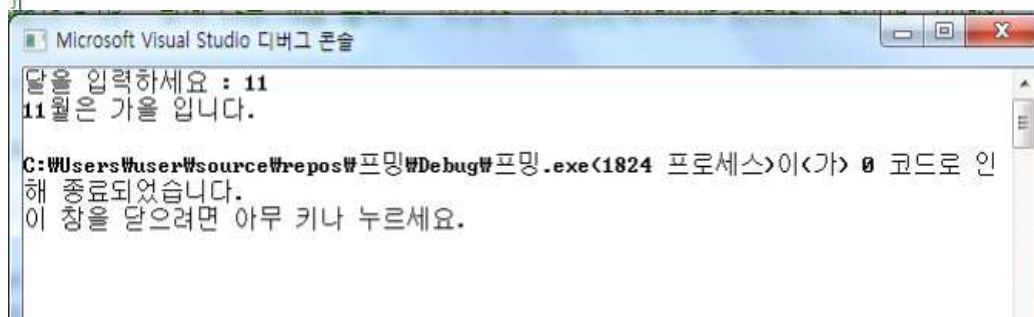
```
#include<stdio.h>

int main(void)
{
    int month = 0; // 정수 month 변수 초기화.

    printf("달을 입력하세요 : ");
    scanf_s("%d", &month); // 정수 month 입력.

    if (month == 12 || month == 1 || month == 2) // month가 12 1 2 세개 중 하나 이면.
        printf("%d월은 겨울 입니다.\n", month); // 겨울 출력.
    else if (month == 3 || month == 4 || month == 5) // month가 3 4 5 세개 중 하나 이면.
        printf("%d월은 봄 입니다.\n", month); // 봄 출력.
    else if (month == 6 || month == 7 || month == 8) // month가 6 7 8 세개 중 하나 이면.
        printf("%d월은 여름 입니다.\n", month); // 여름 출력.
    else if (month == 9 || month == 10 || month == 11) // month가 9 10 11 세개 중 하나 이면.
        printf("%d월은 가을 입니다.\n", month); // 가을 출력.
    else // 1 ~ 12 숫자가 아니면.
        printf("잘못 입력하셨습니다.\n"); // 잘못 입력.

    return 0;
}
```



14. Example – 04

//Example - 04 : if문을 이용한 연산식. 작성자 : 소프트웨어학과 32151671 박민혁. 이메일 : aalsgur9784@naver.com

```
#include<stdio.h>

int main(void)
{
    int invalid_operator = 0; // 프로그램이 잘 수행 됐는지 확인 하는 정수 invalid_operator 변수 초기화.
    char op; // 연산자를 입력 받을 문자 op 변수 초기화.
    float number1=0.0f, number2=0.0f, result=0.0f; // 실수 number1 number2 변수 초기화, 두개의 변수 연산을 저장할 실수 result 초기화.

    printf("두 수를 다음과 같은 형태로 입력하세요.\n");
    printf("연산자는 네 가지(+, -, *, /) 중의 하나여야 합니다.\n");
    printf("number1 연산자 number2\n");
    scanf("%f %c %f", &number1, &op, &number2); // 실수 number1 문자 op 실수 number2 입력.

    if (op == '+') // 문자 op가 '+' 이면.
        result = number1 + number2; // 곱하기 연산 수행 후 result에 저장.
    else if (op == '/') // 문자 op가 '/' 이면.
        result = number1 / number2; // 나누기 연산 수행 후 result에 저장.
    else if (op == '*') // 문자 op가 '*' 이면.
        result = number1 * number2; // 덧셈 연산 수행 후 result에 저장.
    else if (op == '-') // 문자 op가 '-' 이면.
        result = number1 - number2; // 마이너스 연산 수행 후 result에 저장.
    else // 그 외(오류 처리)
        invalid_operator = 1; // invalid_operator 변수가 1이 증가하지 않았으면.

    if (invalid_operator != 1) // invalid_operator 변수가 1이 증가하지 않았으면.
    {
        printf("결과\n");
        printf("X5.2f Xc X5.2f = X5.2f\n", number1, op, number2, result); // 결과 값 출력.
    }
    else // 오른쪽에서 5칸 부터 시작 후 소수점 둘째 자리 까지 나타낸다.
        printf("연산자가 잘못 입력 되었습니다.\n"); // 오류 처리.

    return 0;
}
```

```
Microsoft Visual Studio 디버그 콘솔
두 수를 다음과 같은 형태로 입력하세요.
연산자는 네 가지(+, -, *, /) 중의 하나여야 합니다.
number1 연산자 number2
3.492 / 2.12

>>>>>>>>> 결과는
3.49 / 2.12 = 1.65

C:\Users\82104\source\repos\Project49\Debug\Project49.exe(11340 프로세스)이(가) 0 코드로 인해 종료되었습니다.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디버깅]->[디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

15. Example – 05

```
// Example - 05 : Switch를 이용한 연산식. 작성자 : 소프트웨어학과 32151671 박민혁. 이메일 : alsqur9784@naver.com

#include<stdio.h>

int main(void)
{
    int invalid_operator = 0;
    char op = 0;
    float number1 = 0.0f;
    float number2 = 0.0f;
    float result = 0.0f;

    printf("두 수를 다음과 같은 형태로 입력하세요.\n");
    printf("연산자는 네 가지(+, -, *, /) 중의 하나여야 합니다.\n");
    printf("number1 연산자 number2\n");
    scanf_s("%f %c %f", &number1, &op, &number2);

    switch (op)
    {
        case '+':
            result = number1 + number2;
            break;
        case '/':
            result = number1 / number2;
            break;
        case '-':
            result = number1 - number2;
            break;
        case '*':
            result = number1 * number2;
            break;
        default:
            invalid_operator = 1;
    }

    switch (invalid_operator)
    {
        case 1:
            printf("연산자가 잘못 입력 되었습니다.\n");
            break;
        default:
            printf("\n>>>>>>>>> 결과는 \n");
            printf("%5.2f %c %5.2f = %5.2f\n", number1, op, number2, result);
            break;
    }

    return 0;
}
```



```
Microsoft Visual Studio 디버그 콘솔
두 수를 다음과 같은 형태로 입력하세요.
연산자는 네 가지(+, -, *, /) 중의 하나여야 합니다.
number1 연산자 number2
28.123 * 5.32

>>>>>>>> 결과는
28.12 * 5.32 = 149.61

C:\Users\82104\source\repos\Project49\Debug\Project49.exe(2564 프로세스)이(가) 0 코드로 인해 종료되었습니다.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디버깅]->[디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도
록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요.
```

16. Example – 06

```
// Example - 06 : if문과 Switch문 중첩 사용. 작성자 : 소프트웨어학과 82151571 박인혁, 이메일 : alsgr3754@naver.com
#include <stdio.h>

int main(void)
{
    int year = 0, month = 0, maxday = 30, validinput = 0; // 년도 및 max 날짜 오류 처리를 위한 변수 초기화.

    printf("년과 월을 다음과 같이 입력하면 그 달의 일수를 알려드립니다.\n");
    printf("2003 2월입력>");
    scanf_s("%d %d", &year, &month); // 년도와 월 입력.

    switch (month) // month에 대한 조건문.
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12: // 1 3 5 7 8 10 12월은 maxday가 31일. 그리고 오류 처리를 위한 변수 1 증가.
            maxday = 31;
            validinput = 1;
            break;

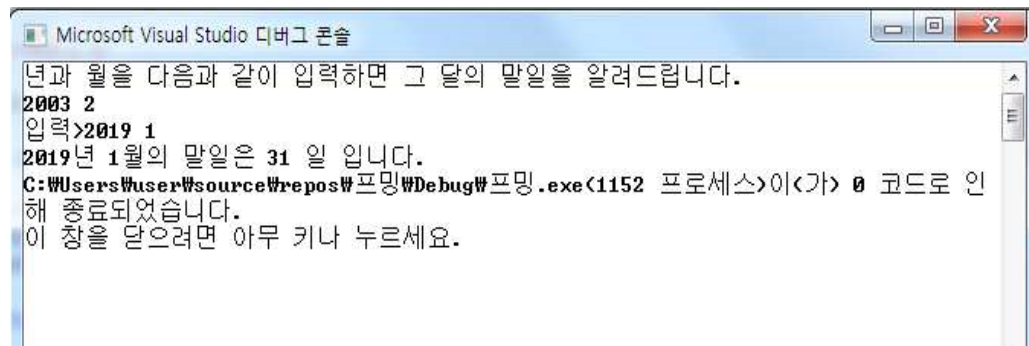
        case 4:
        case 6:
        case 9:
        case 11: // 4 6 9 11월은 maxday가 30일. 그리고 오류 처리를 위한 변수 1 증가.
            maxday = 30;
            validinput = 1;
            break;

        case 2: // 2월은 윤년 계산.
            if ((year % 4 == 0 && (year % 100 != 0) || (year % 400 == 0)) // 윤년 계산 식
                maxday = 29; // 윤년이면 maxday가 29일.
            else // 아니면
                maxday = 28; // maxday는 28일.
            validinput = 1; // 오류 처리를 위한 변수 1 증가.
            break;

        default:
            validinput = 0;
    }

    if (validinput)
        printf("%d년 %d월의 일수는 %d 일 입니다.", year, month, maxday); // validinput(1)이면 대한 조건문.
    else // 값과 같지 않음
        printf("입력이 잘못 되었습니다.\n"); // validinput(0)이면.

    return 0;
}
```



```
Microsoft Visual Studio 디버그 콘솔
년과 월을 다음과 같이 입력하면 그 달의 말일을 알려드립니다.
2003 2
입력>2019 1
2019년 1월의 말일은 31 일 입니다.
C:\Users\User\source\repos\프밍\Debug\프밍.exe<1152 프로세스>이<가> 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

17. Example – 07

// Example - 07 : Switch문을 이용한 가위바위보, 작성자 : 소프트웨어학과 32151671 박민혁, 이메일 : alsgur9784@naver.com

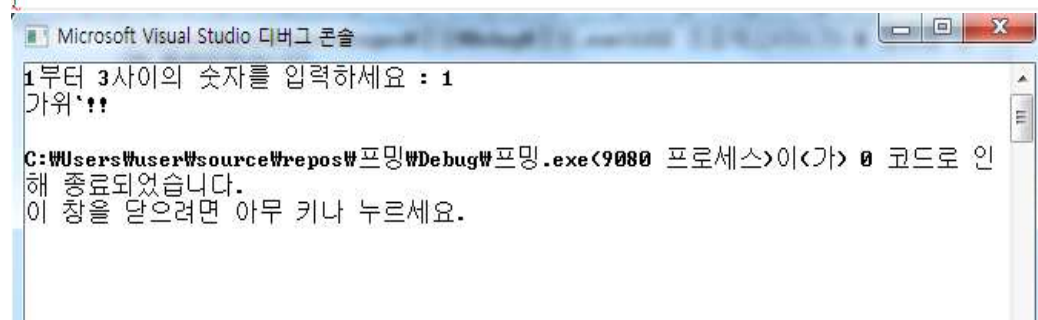
```
#include<stdio.h>

int main(void)
{
    int toss = 0; // 정수 toss 변수 초기화.

    printf("1부터 3사이의 숫자를 입력하세요 : ");
    scanf_s("%d", &toss); // 정수 toss 입력.

    switch (toss) // toss 변수에 대한 조건문.
    {
        case 1: // toss가 1이면,
                printf("가위\n"); // 가위 출력.
                break;
        case 2: // toss가 2이면,
                printf("바위\n"); // 바위 출력.
                break;
        case 3: // toss가 3이면,
                printf("보\n"); // 보 출력.
                break;
        default : // 그 외(오류).
                  printf("1부터 3사이의 숫자가 아닙니다.\n"); // 오류 처리.
    }

    return 0;
}
```



```
Microsoft Visual Studio 디버그 콘솔
1부터 3사이의 숫자를 입력하세요 : 1
가위\n
C:\Users\User\source\repos\프밍\Debug\프밍.exe<9080 프로세스>이<가> 0 코드로 인
해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.
```

18. Example – 08

// Example - 08 : Switch문을 이용한 계절 출력. 작성자 : 소프트웨어학과 32151671 박민혁. 이메일 : a1sgur9784@naver.com

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

```
int main(void)
```

```
{
```

```
    int month = 0; // 달 month 변수 초기화.
```

```
    printf("달을 입력하세요 : ");
```

```
    scanf_s("%d", &month); // 정수 month 변수 입력.
```

```
    switch (month) // month에 대한 조건문.
```

```
    { // month가 12 1 2 이면.
```

```
    case 12:
```

```
    case 1:
```

```
    case 2:
```

```
        printf("%d월은 겨울 입니다.\n", month); // 겨울 출력.
```

```
        break;
```

```
    case 3:
```

```
    case 4:
```

```
    case 5:
```

```
        printf("%d월은 봄 입니다.\n", month); // 봄 출력.
```

```
        break;
```

```
    case 6:
```

```
    case 7:
```

```
    case 8:
```

```
        printf("%d월은 여름 입니다.\n", month); // 여름 출력.
```

```
        break;
```

```
    case 9:
```

```
    case 10:
```

```
    case 11:
```

```
        printf("%d월은 가을 입니다.\n", month); // 가을 출력.
```

```
        break;
```

```
    default :
```

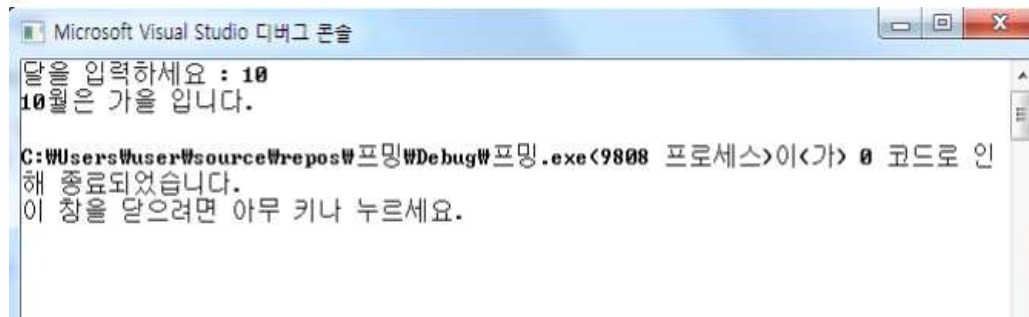
```
        printf("잘못 입력하셨습니다.\n"); // 그 외(오류) 1~12 숫자가 아니면.
```

```
        // 오류 처리.
```

```
    }
```

```
    return 0;
```

```
}
```



Ⅲ. 과제 고찰

사실 3학년이라서 1학년 수업을 듣는 것이 조금 부끄러운 일이다. 그리고 중간고사 점수도 불 만족스러운 점수라서 재수강을 해도 처음 배우는 신입생들에게 점수가 뒤쳐진 것도 부끄럽다. 하지만 과제만이라도 신입생 친구들보다 뛰어나게 하고 싶어서 과제에 노력을 더 기울였다. 확실히 1학년 때 프로그래밍 과제를 했을 때를 떠올리면, 왜 이것을 못해서 친구들과 상의를 했을까, 왜 과제를 보고 제출을 했을까 많이 후회 됐다. 하지만 지금이라도 열심히 해서 성적이 좋게 뜬다면 괜찮다. 일반적인 프로그래밍 코딩은 쉽다. 하지만 과제를 하면서 기초적인 이론이 많이 부족하다는 것을 깨달았다. 당연히 알고 있을 줄 알았던 것들도 많이 헛갈렸다. 헛갈리면 안 되는 시기는 하지만 지금이라도 알아가서 다행이라고 생각한다. 남들이 프로그래밍 재수강 한다고 비웃을지는 모르겠지만, 프로그래밍이라는 과목을 들으면서 얻어 가는 것이 있기에 괜찮은 것 같다. 그래도 과제 제출이나 형식 이런 부분에 있어서는 많이 해봤기 때문에 신입생 친구들 보다는 잘 할 수 있다고 생각한다. 그리고 지난 학기에 웹 프로그래밍이라는 수업을 이원진 교수님께 들었기에 교수님이 어떤 것들을 위주로 내고 무엇을 중요하게 생각 하는지 그래도 다른 친구들 보다는 잘 알고 있다고 생각한다. 앞으로 남은 기말고사는 중간고사 때 보다 조금 더 열심히 하여 A라는 성적을 맞을 것이다. 중간고사는 비록 아쉬운 점수지만 기말고사 때는 중간고사 때 보다 열심히 하여 좋은 성적을 맞을 것이다. 그렇기에 과제에 조금 더 힘을 실었고 무엇보다 과제 점수를 중요시 여기는 교수님이기에 과제 수정을 오래 하였다. 과제를 하면서 일학년 수업이어도 무시하면 안 된다는 것을 느끼게 되었다.