

REPORT



과 목 명 : 프로그래밍

담당교수 : 이원진 교수님

소 속 : 소프트웨어학과

학 번 : 32151671

이 름 : 박민혁



단국대학교
Dankook University

프로그래밍 제 2차 과제

I. 과제 개요

1. 과제 내용

-첨부된 과제 관련 파일을 확인 후, 작성하여 제출

2. 과제 기본 개념

(1). 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)

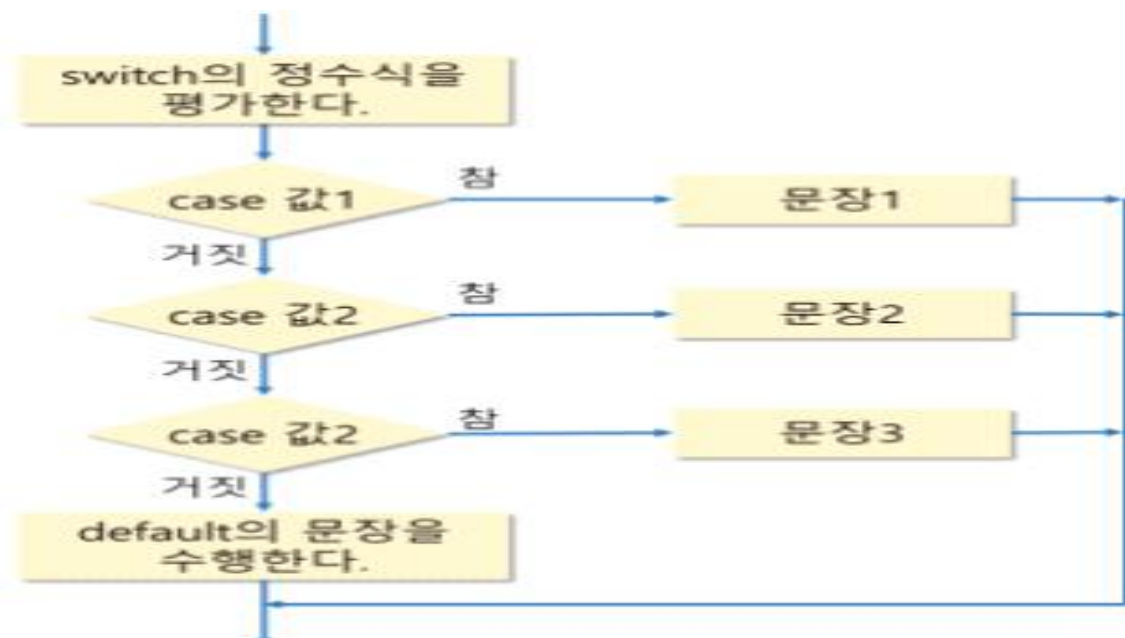
- 조건문 if else문으로서, else 이후에 if else를 필요한 횟수만큼 반복 가능

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

(2). 6장 - 반복문

반복문

- 조건이 만족하는 동안 주어진 문장을 반복해서 수행 -> 루프(loop)
- 반복문을 이용하면 사용자가 원하는 만큼 반복적으로 수행하게 만들 수 있음

반복문의 종류	구문	특징
for	for (초기화; 조건검사; 증감연산) { for 문 몸체(body); };	일정한 반복 횟수를 이용하는 반복문에 적합.
while	while (조건검사) { while 문 몸체(body); };	구문이 간단하며, 검사부분이 처음에 있음.
do while	do { do while 문 몸체(body); } while (조건검사);	반복 몸체를 1번은 실행하며, 검사부분이 뒤에 있음.

반복문 종류

- while

- (1). 반복문 while은 단순한 숫자의 반복이 아니라 반복할 때마다 조건을 따지는 반복문
- (2). 조건식이 반복몸체 앞에 위치

- do while

- (1). 조건식이 반복몸체 뒤에 위치하므로 처음에 조건을 검사 할 수는 없음
- (2). 무조건 한 번 실행 한 후 조건을 검사하고 이때 조건식이 참(0이 아니면)이면 반복을 더 실행

- for

- (1). 반복문 for는 숫자로 반복하는 횟수를 제어하는 반복문
- (2). 명시적으로 반복 횟수를 결정할 때 주로 사용

반복문 필요성

- 일정한 문장의 반복

- (1). 여러 줄의 동일한 패턴의 출력
- (2). 1부터 n까지의 합

while문

- for 문보다 구조가 간단한 반복문으로 다음과 같이 조건검사 부분인 (expression)을 while 다음에 괄호로 구성

- (1). 조건검사(expression) 부분을 평가하여 0이 아니면(참) 반복문체인 statement를 실행
- (2). 반복이 실행되는 statement를 반복문체(repetition body)라 부름
- (3). while문은 for나 do while 반복문 보다 간단

while문 실행순서

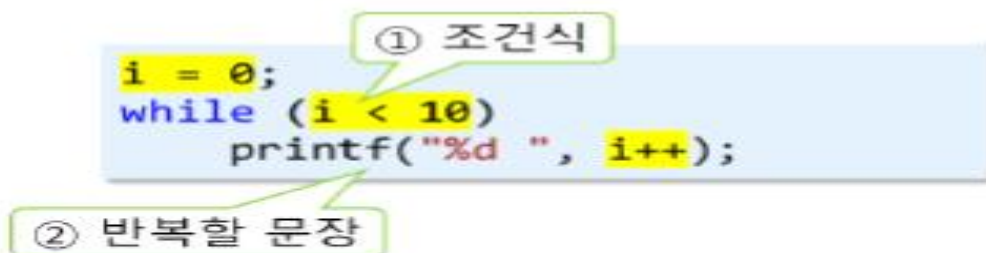
- (1). 조건문인 expression을 검사하여 0이 아니면(참) 반복문의 몸체에 해당하는 문장이나 블록을 실행함
- (2). 몸체를 실행하면 다시 조건검사를 조사하여 1의 과정을 다시 실행함
- (3). 조건검사가 거짓이면 while 문장을 종료함

형식

while (조건식)
반복할문장;

사용예

```
i = 0;  
while (i < 10)  
    printf("%d ", i++);
```



while의 수행 순서

①→②→①→②→...→①

do while문

- while문과 유사하나 반복 조건이 몸체의 마지막에 있는 것이 차이
- 가장 먼저 statement를 실행한 이후 반복 조건(expression)를 평가

- (1). 0(거짓)이면 do while 문을 종료
- (2). 반복몸체에 특별히 분기 구문이 없는 경우, do while의 몸체는 적어도 한 번은 실행
- (3). while 이후의 세미콜론(;)은 반드시 필요

do ~ while문 실행순서

- (1). 문장들이 실행 됨.
- (2). 조건식이 계산 됨
- (3). 결과가 참이면 돌아감
- (4). 결과가 거짓이면 종료

형식

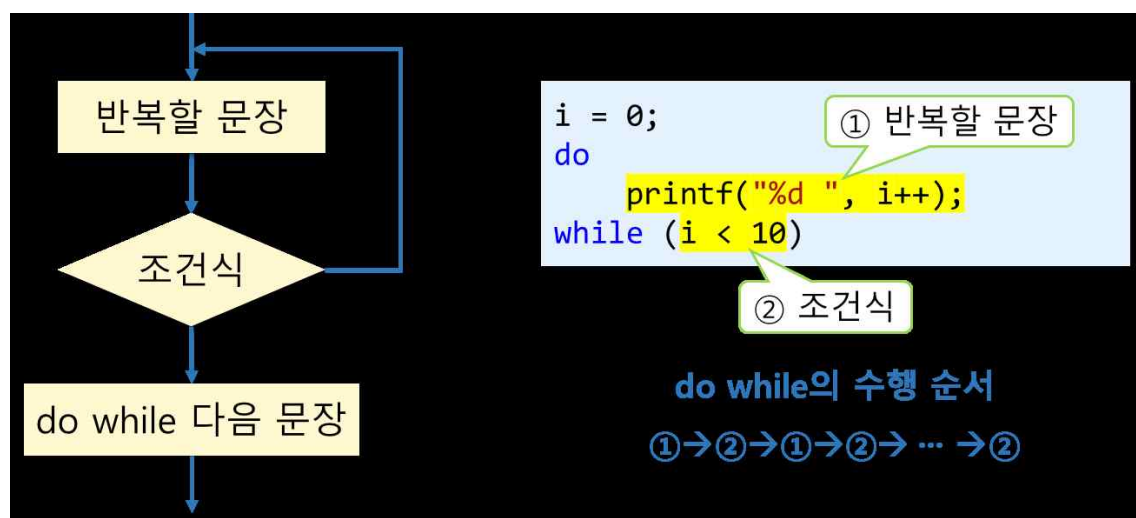
do

반복할문장;

while (조건식);

사용예

```
i = 0;
do
    printf("%d ", i++);
while (i < 10);
```



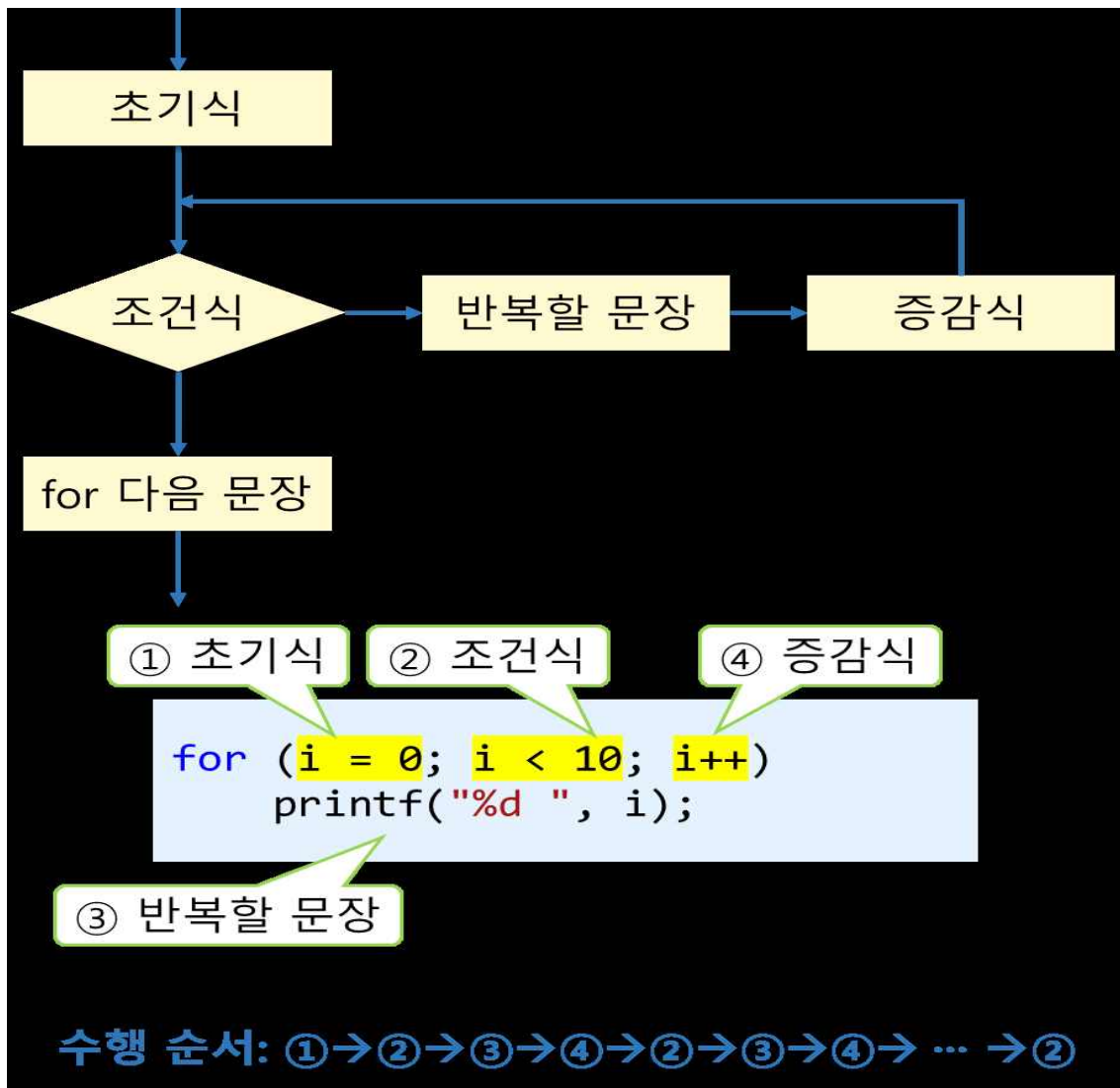
for문

형식

for (초기식; 조건식; 증감식)
반복할문장;

사용예

```
for (i = 0; i < 10; i++)  
    printf("%d ", i);
```



- 초기화(initialization). 반복조건을 검사, 제어변수의 증감(increment)을 수행
- 주의점
 - (1). for(; ;)의 괄호 내부에서 세미콜론으로 구분되는 항목은 모두 생략 가능
 - (2). 2개의 세미콜론은 반드시 필요
 - (3). 반복조건을 제거하면 반복은 무한히 계속
 - (4). 반복할 문장인 반복몸체가 여러 개라면 반드시 블록으로 구성

구조

for(초기화; 조건식; 증감식) 문장;

- (1). 초기화를 실행함
- (2). 반복 조건을 나타내는 조건식을 계산함 (수식의 값이 거짓이면 for문의 실행이 종료)
- (3). 수식의 값이 참이면 문장이 실행
- (4). 증감을 실행하고 2로 돌아감

break문

- switch문 안에 사용하면 제어의 흐름이 switch를 탈출해서 switch의 다음 문장으로 이동함
- for, while, do while 등의 반복문 안에서 사용하면 반복문을 빠져 나가게 됨

continue문

- 루프의 시작이나 끝 부분으로 이동
- for문 안에서 continue를 for의 시작 부분으로 이동해서 루프를 반복함

goto문

- 제어의 흐름을 프로그램의 특정 위치로 이동
- 레이블 : 이동할 문장을 가리킴 (레이블 이름과 콜론 사용)
- 꼭 필요한 경우가 아니면 goto문을 사용하지 않는 것이 좋음
- 한꺼번에 여러 개의 루프를 탈출해야 할 때 유용

return문

- 프로그램 수행 중에 return문을 만나면 함수를 호출한 곳으로 되돌아감
- 리턴 값이 있는 함수에서는 return 다음에 값을 써주고, 리턴 값이 없는 함수에서는 return만 써줌

II. 과제 소스 코드와 결과

1. Example - 01

```
// 1. 다음 코드는 마일을 킬로미터로 변환하는 프로그램이다.
// 주석에 해당되는 문장을 작성하라.
// 작성자 : 소프트웨어학과 32151671 박민혁 이메일 : a1sgur9784@naver.com


#include<stdio.h> // 표준 입출력 함수의 정보를 포함함

int main(void) // main 함수
{
    float mile = 0.0f; // mile 변수를 float형으로 선언
    float km = 0.0f; // km 변수를 float형으로 선언

    printf("마일을 입력하시오 : "); // "마일을 입력하시오."라는 텍스트 출력
    scanf_s("%f", &mile); // 키보드로부터 값을 받아서 mile에 저장

    km = mile * 1.609; // 변수 mile에 1.609를 곱하여 킬로미터를 얻음
    printf("마일을 킬로미터로 변환한 값 : %f\n", km); // 킬로미터를 출력

    return 0; // 0을 반환하여 main() 종료
}
```



Microsoft Visual Studio 디버그 콘솔

마일을 입력하시오 : 10
마일을 킬로미터로 변환한 값 : 16.090000

C:\Users\User\source\repos\프로그래밍과제2\Debug\프로그래밍과제2.exe(9912 프로세스)이<가> 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.

2. Example - 02

```
// 2. 사용자가 아스키 코드값을 입력하면 그 아스키 코드값에 해당하는 문자를 출력하는 프로그램을 작성하시오.
// 예를 들어서 65를 입력하면 'A'가 출력된다.
// 작성자 : 소프트웨어학과 32151671 박민혁 이메일 : a1sgur9784@naver.com

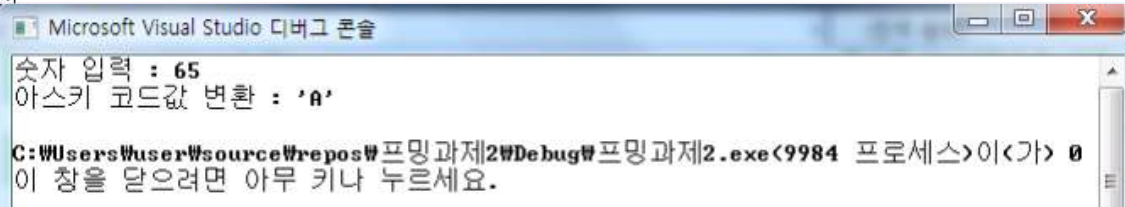
#include<stdio.h> // 표준 입출력 함수의 정보를 포함함

int main(void) // main 함수
{
    int value = 0; // value 변수를 int형으로 선언

    printf("숫자 입력 : "); // "숫자 입력 : "이라는 텍스트 출력
    scanf_s("%d", &value); // 키보드로부터 값을 받아서 value에 저장

    printf("아스키 코드값 변환 : '%c'\n", value); // 아스키 코드값 변환 출력 아스키 코드는 문자형이므로 %c로 출력

    return 0; // 0을 반환하여 main() 종료
}
```



Microsoft Visual Studio 디버그 콘솔

숫자 입력 : 65
아스키 코드값 변환 : 'A'

C:\Users\User\source\repos\프로그래밍과제2\Debug\프로그래밍과제2.exe(9984 프로세스)이<가> 0
이 창을 닫으려면 아무 키나 누르세요.

3. Example – 03

// 3. 조건 연산자를 이용하여 사용자로부터 받은 세 개의 정수 중에서 가장 작은 수를 출력하는 프로그램을 작성하시오.
 // 작성자 : 소프트웨어학과 32151671 박민혁 이메일 : a1sgur9784@naver.com

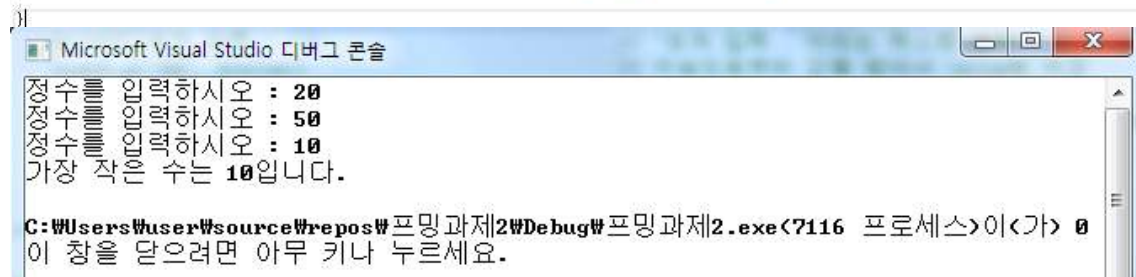
```
#include<stdio.h> // 표준 입출력 함수의 정보를 포함함

int main(void) // main 함수
{
    int num1 = 0;
    int num2 = 0;
    int num3 = 0; // num1 num2 num3 변수를 int형으로 선언

    printf("정수를 입력하시오 : ");
    scanf_s("%d", &num1);
    printf("정수를 입력하시오 : ");
    scanf_s("%d", &num2);
    printf("정수를 입력하시오 : ");
    scanf_s("%d", &num3); // 키보드로부터 값을 받아서 num1 num2 num3에 저장

    if (num1 <= num2 && num1 <= num3) // num1이 num2보다 작거나 같고, num3보다 작거나 같으면
        printf("가장 작은 수는 %d입니다.\n", num1); // num1 값을 최소값으로 출력
    else if (num2 <= num1 && num2 <= num3) // num2가 num1보다 작거나 같고, num3보다 작거나 같으면
        printf("가장 작은 수는 %d입니다.\n", num2); // num2 값을 최소값으로 출력
    else // num3이 num1보다 작거나 같고, num3보다 작거나 같으면
        printf("가장 작은 수는 %d입니다.\n", num3); // num3 값을 최소값으로 출력

    return 0; // 0을 반환하여 main() 종료
}
```



4. Example – 04 : 다음의 문장을 switch문을 사용하여 재 작성하시오.

<pre>if(i==10) n=n+1; else if(i==20) m=m+1; else x=x+1;</pre>	<pre>switch(i) { case 10 : n=n+1; break; case 20 : m=m+1; break; default : x=x+1;</pre>
---	---

5. Example – 05 : 다음의 문자를 연속적인 if-else를 사용하여 재 작성하십시오.

<pre>switch(code) { case 'X' : x++; break; case 'Y' : y++; break; default : printf("X나 Y를 입력하십시오.\n"); break; }</pre>	<pre>if(code=='X') x++; else if(code=='Y') y++; else printf("X나 Y를 입력하십시오.\n");</pre>
---	---

6. Example – 06

```
// 6. 사용자로부터 입력받은 두 수의 합과 차를 구하는 프로그램을 작성하십시오.
// (두 수의 차는 큰수에서 작은 수를 뺀 것을 한다.)
// 만약 첫 번째 정수가 두 번째 정수보다 작을 경우 고려, 그래서 if-else문으로 조건을 처리해야함.
// 작성자 : 소프트웨어학과 32151671 박민혁 이메일 : alsur9784@naver.com

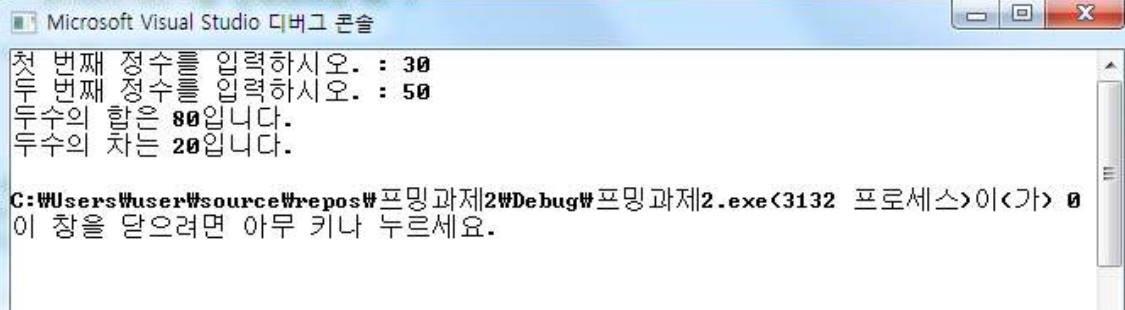
#include<stdio.h> // 표준 입출력 함수의 정보를 포함함

int main(void) // main 함수
{
    int num1 = 0; // num1과 num2 변수를 int형으로 선언
    int num2 = 0;

    printf("첫 번째 정수를 입력하십시오. : ");
    scanf_s("%d", &num1); // 키보드로부터 값을 받아서 num1과 num2에 저장
    printf("두 번째 정수를 입력하십시오. : ");
    scanf_s("%d", &num2);

    printf("두수의 합은 %d입니다.\n", num1 + num2); // num1과 num2의 합을 먼저 출력
    if (num1 >= num2) // num1이 num2보다 크거나 같으면
        printf("두수의 차는 %d입니다.\n", num1 - num2); // num1에서 num2를 뺀 것을 출력
    else // num2가 num1보다 크면
        printf("두수의 차는 %d입니다.\n", num2 - num1); // num2에서 num1을 뺀 것을 출력

    return 0; // 0을 반환하여 main() 종료
}
```



7. Example – 07

```

// 7. 컴퓨터와 가위바위보 게임을 하는 프로그램을 작성하시오.
// 컴퓨터는 사용자에게 알리지 않고 가위,바위,보 중에서 임의로 하나 선택한다.
// 사용자는 프로그램의 안내 메시지에 따라서 3개중의 하나를 선택하게 된다.
// 사용자의 선택이 끝나면 컴퓨터는 누가 무엇을 선택하였고, 누가 이겼는지를 출력한다.
// 기본 점수로 승패의 경우를 미리 정해보자.
// 사용자가 가위(0)를 선택하면, 비겼다. 사용자가 바위(1)를 선택하면 컴퓨터가 졌다. 사용자가 보(2)를 선택하면 컴퓨터가 이겼다.
// -> 추가점수는 컴퓨터가 랜덤값 처리를 해서 실제 가위바위보 게임을 만들 경우
// 작성자 : 소프트웨어학과 32151671 박민혁 이메일 : a1sgur9784@naver.com

#include<stdio.h> // 표준 입출력 함수의 정보를 포함함
#include<stdlib.h> // 랜덤함수 사용을 위한 헤더 파일
#include<time.h> // 현재 시간을 기준으로 랜덤함수를 돌리기 위해 필요한 헤더 파일

int main(void) // main 함수
{
    srand((int)time(NULL)); // 현재 시간을 이용해서 씨드 설정
    int user = 0; // user 변수를 int형으로 선언
    int computer = rand() % 3; // computer 변수를 int형으로 선언 하고, 0,1,2 세 숫자로만 랜덤함수를 돌리기 위해 3으로 나눈 나머지 값 이용

    printf("가위<0>, 바위<1>, 보<2>를 입력하세요 : "); // "가위<0>, 바위<1>, 보<2>를 입력하세요 : " 텍스트 출력
    scanf_s("%d", &user); // 키보드로부터 입력 받아서 user에 저장

    if (user < 0 || user > 2) // 키보드로부터 입력 받은 user의 값이 0보다 작거나 2보다 크면
    {
        printf("error\n"); // "error" 텍스트 출력
        return 1; // 1을 반환하여 main() 오류 처리
    }

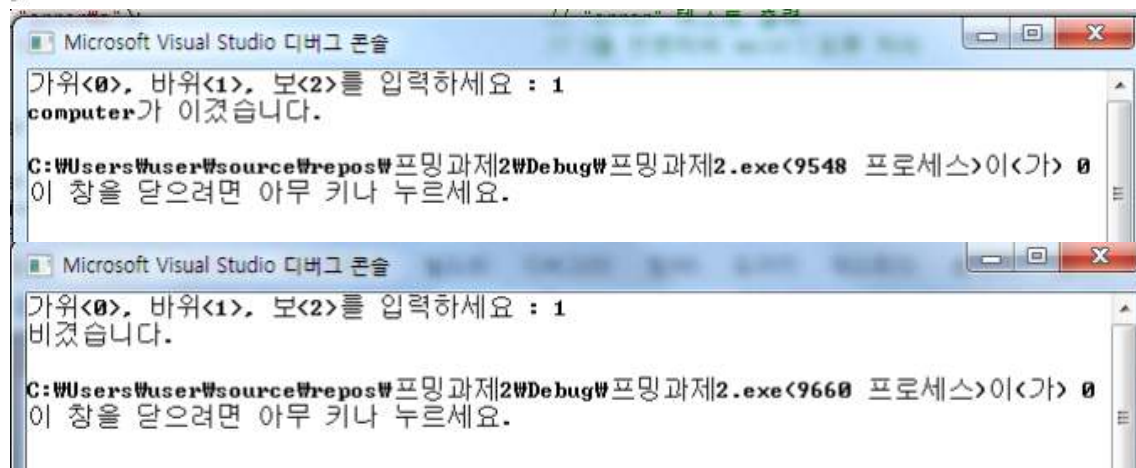
    if (computer == 0) // computer 랜덤 변수가 가위(0)이면
    {
        if (user == 0) // user 변수가 가위(0)이면
            printf("비겼습니다.\n"); // "비겼습니다" 텍스트 출력
        else if (user == 1) // user 변수가 바위(1)이면
            printf("computer가 졌습니다.\n"); // "computer가 졌습니다" 텍스트 출력
        else if (user == 2) // user 변수가 보(2)이면
            printf("computer가 이겼습니다.\n"); // "computer가 이겼습니다" 텍스트 출력
    }

    else if (computer == 1) // computer 랜덤 변수가 바위(1)이면
    {
        if (user == 0) // user 변수가 가위(0)이면
            printf("computer가 이겼습니다.\n"); // "computer가 이겼습니다" 텍스트 출력
        else if (user == 1) // user 변수가 바위(1)이면
            printf("비겼습니다.\n"); // "비겼습니다" 텍스트 출력
        else // user 변수가 보(2)이면
            printf("computer가 졌습니다.\n"); // "computer가 졌습니다" 텍스트 출력
    }

    else // computer 랜덤 변수가 보(2)이면
    {
        if (user == 0) // user 변수가 가위(0)이면
            printf("computer가 졌습니다.\n"); // "computer가 졌습니다" 텍스트 출력
        else if (user == 1) // user 변수가 바위(1)이면
            printf("computer가 이겼습니다.\n"); // "computer가 이겼습니다" 텍스트 출력
        else // user 변수가 보(2)이면
            printf("비겼습니다.\n"); // "비겼습니다" 텍스트 출력
    }

    return 0; // 0을 반환하여 main() 종료
}

```



8. Example – 08 : 다음의 코드에서 잘못된 점이 있으면 지적하고 올바르게 수정하여라. (논리적인 오류도 포함)

<수정전> <pre>int i=0; while(i<10) { printf("%d\n",i); }</pre>	<수정전> <pre>int i=0; for(i=0; i<10; i++); { printf("i=%d\n",i); }</pre>
<수정 후> <pre>int i=0; while(i<10) { printf("%d\n",i); i++; }</pre>	<수정 후> <pre>int i=0; for(i=0; i<10; i++) { printf("i=%d\n",i); }</pre>

첫 번째 while문은 무한 반복이다. 무한 반복을 없애기 위해서는 증감식이 while문 안에 들어야 한다.

두 번째 for문은 반복문이 실행되기는 하지만 아무것도 출력이 되지 않는다. 그 이유는, for 문 다음 바로 세미콜론(;)이 찍혀 있기 때문이다.

9. Example – 09 : 다음의 while루프는 for루프로, for루프는 while 루프로 변경하여라.

<수정전> <pre>int i=0; while(i<10) { printf("%d\n",i); }</pre>	<수정전> <pre>int i; for(i=1; i<=20; i++) { printf("%d\n",i); }</pre>
<수정 후> <pre>int i; for(i=0; i<10;) { printf("%d\n",i); }</pre>	<수정 후> <pre>int i=1; while(i<=20) { printf("%d\n",i); i++; }</pre>

10. Example – 10

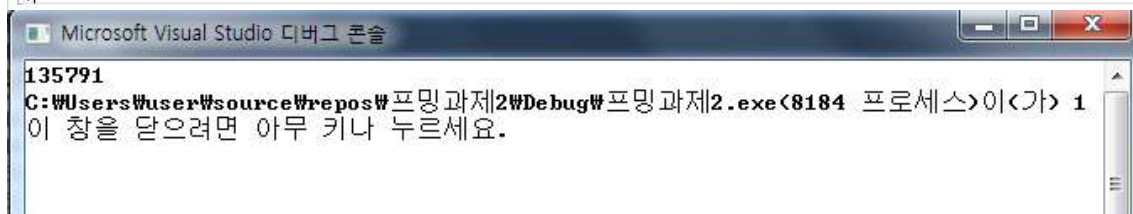
// 10. 다음 두 개의 프로그램을 보고 출력결과를 표시하고, 그 이유를 적으시오.
 // 작성자 : 소프트웨어학과 32151671 박민혁 이메일 : aalsgur9784@naver.com

```
#include<stdio.h> // 표준 입출력 함수의 정보를 포함함

int main(void) // main 함수
{
    for (int i = 1; i <= 10; i++) // i의 초기값 1 / i가 10보다 작거나 같으면 반복 / 한 번 실행 후 i값 1증가
    {
        if (i % 2 == 0) // i가 짝수면
            continue; // 계속
        printf("%d", i); // 홀수 출력
    }

    for (int i = 1; i <= 10; i++) // i의 초기값 1 / i가 10보다 작거나 같으면 반복 / 한 번 실행 후 i값 1증가
    {
        if (i % 2 == 0) // i가 짝수면
            return 1; // 1을 반환하여 main() 오류 처리
        printf("%d", i); // 홀수 출력
    }

    return 0; // 0을 반환하여 main() 종료
}
```



위에 주석과 같이 첫 번째 for문에서는 짝수일 경우에는 for의 시작 부분으로 다시 이동하여 13579라는 텍스트가 출력 된다. 하지만 두 번째 for문에서는 짝수일 경우에 return 1을 사용하여 오류 처리를 했다. 그래서 1만 출력이 되고 2가 출력 될 차례 때 프로그램 수행이 종료된다.

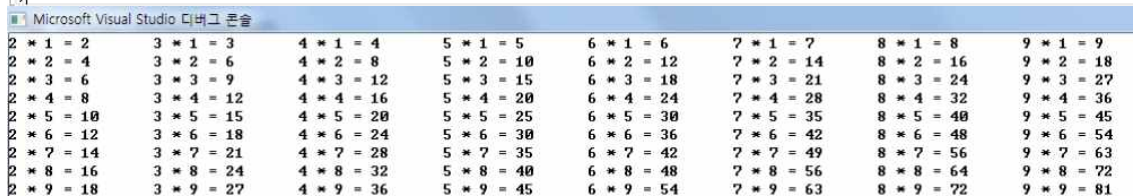
11. Example – 11

// 11. 구구단(2단~9단)을 가로로 출력하는 프로그램을 작성하시오. (for 중첩문)
 // 작성자 : 소프트웨어학과 32151671 박민혁 이메일 : aalsgur9784@naver.com

```
#include<stdio.h> // 표준 입출력 함수의 정보를 포함함

int main(void) // main 함수
{
    for (int i = 1; i <= 9; i++) // i의 초기값 1 / i가 9보다 작거나 같으면 반복 / 한 번 실행 후 i값 1증가
    {
        for (int j = 2; j <= 9; j++) // j의 초기값 2 / j가 9보다 작거나 같으면 반복 / 한 번 실행 후 j값 1증가
        {
            printf("%d * %d = %d\t", j, i, j*i); // 가로부터 출력
            printf("\n"); // 한 번 수행이 끝나면 줄바꿈
        }

        return 0; // 0을 반환하여 main() 종료
    }
}
```



C:\Users\User\source\repos\프밍과제2\Debug\프밍과제2.exe (9632 프로세스)이<가> 0 코드로 인해 종료되었습니다.
 이 창을 닫으려면 아무 키나 누르세요.

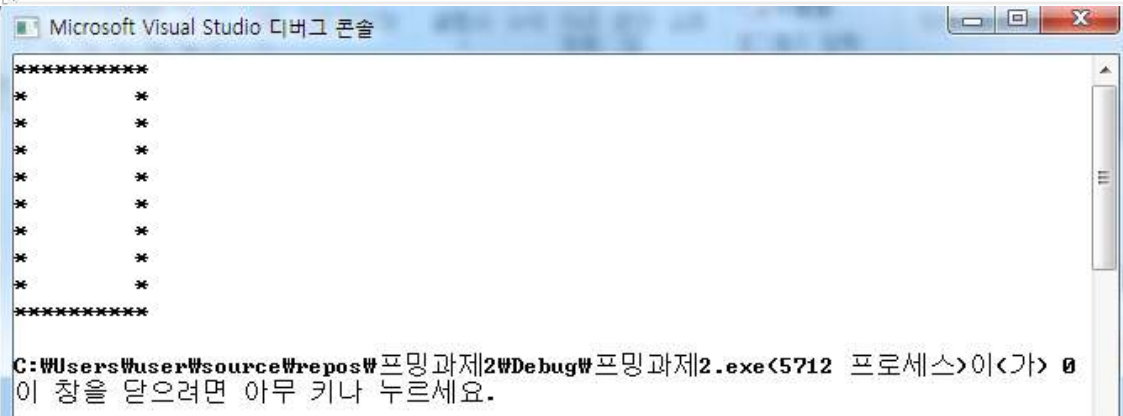
12. Example – 12

```
// 12. 반복문을 이용해 다음과 같은 패턴을 출력하는 프로그램을 작성하시오.
// for 중첩으로 반복하면서 2~7줄에는 1,9번째에만 *을 찍기 위해 if문과 OR연산자를 이용.
// 작성자 : 소프트웨어학과 32151671 박민혁 이메일 : alsur9784@naver.com

#include<stdio.h>                                // 표준 입출력 함수의 정보를 포함함

int main(void)                                   // main 함수
{
    for (int i = 1; i <= 10; i++)                // i의 초기값 1 / i가 10보다 작거나 같으면 반복 / 한 번 실행 후 i값 1증가
    {
        for (int j = 1; j <= 10; j++)            // j의 초기값 1 / j가 10보다 작거나 같으면 반복 / 한 번 실행 후 j값 1증가
        {
            if (i == 1 || i == 10)                // i의 값이 1 또는 10이면
                printf("*");                      // "*" 10개 텍스트 출력
            else                                    // i의 값이 1 또는 10이 아니면
            {
                if (j == 1 || j == 9)              // j의 값이 1 또는 9이면
                    printf("*");                  // 첫 번째 칸과 아홉 번째 칸에 "*" 텍스트 출력
                printf(" ");                      // 띄어쓰기
            }
        }
        printf("\n");                             // 한 번 수행이 끝나면 줄바꿈
    }

    return 0;                                     // 0을 반환하여 main() 종료
}
```



Microsoft Visual Studio 디버그 콘솔

```
*****
*      *
*      *
*      *
*      *
*      *
*      *
*      *
*      *
*****

C:\Users\User\source\repos\프밍과제2\Debug\프밍과제2.exe(5712 프로세스)이<가> 0
이 창을 닫으려면 아무 키나 누르세요.
```

III. 과제 고찰

첫 번째 과제를 한 지 얼마 안 돼서 두 번째 과제가 나왔다. 사실 이번 과제가 전 과제보다 쉬웠다. 그 이유는 코딩을 하는데 있어서 어려웠지 않았기 때문이다. 하지만 별 찍기나 구구단 같은 기본적인 코딩을 한 번 더 복습 하는 것이 좋았다. 딱히 어려운 문제는 아니었지만 기초를 다시 다지는 기분이라 괜찮았다. 그리고 이번 과제는 전 과제보다 시간이 덜 소요 되어서 좋았다. 앞으로 과제가 더 있을지는 모르겠지만 시험공부 하는데 도움이 되었다.