

C언어 기초 part. 1

Week 3 – 제어흐름 (조건문과 반복문)

QnA 메일 : edujongkook@gmail.com

Pdf 파일 : [github.com/edujongkook
/pdf_sbs_c_weekend](https://github.com/edujongkook/pdf_sbs_c_weekend)

목차

A table of contents

- 1 if 조건문 - 관계, 논리 연산자
- 2 switch 조건문
- 3 while 반복문 - 복합 대입 연산자
- 3 for 반복문 - 증감 연산자



복습 변수로 학생 데이터 만들기



이 름 : 김철수

나 이 : 16

국어점수 : 80

영어점수 : 100

```
int main(void) {  
    char name[ ] = "김철수";  
    int age = 16;  
  
    int kor = 80;  
    int eng = 100 ;  
}
```

복습 변수로 학생 데이터 만들기

```
int main(void) {  
    char name[] = "김철수";  
    int age = 16, kor = 100, eng = 80;  
  
    printf("==== 학생정보를 출력합니다 =====\n");  
    printf("이름 : %s\n", name);  
    printf("나이 : %d세\n", age);  
    printf("국어점수 : %d점\n", kor);  
    printf("영어점수 : %d점\n", eng);  
}
```

출력결과

==== 학생정보를 출력합니다 =====

이름:김철수

나이:16세

국어점수:80점

영어점수:100점

복습 scanf_s로 학생 데이터 입력



이름 :

나이 :

국어점수 :

영어점수 :

키보드 입력

```
int main(void) {  
    char name[100];  
    int age, kor, eng;  
  
    printf("이름을 입력해 주세요:\n");  
    scanf_s("%s", &name, 100);  
    printf("나이를 입력해 주세요:\n");  
    scanf_s("%d", &age);  
    printf("국어점수를 입력해 주세요:\n");  
    scanf_s("%d", &kor);  
    printf("영어점수를 입력해 주세요:\n");  
    scanf_s("%d", &eng);  
}
```

복습 scanf_s로 학생 데이터 입력

이름 :

나이 :

국어점수 :

영어점수 :

키보드 입력



===== 학생정보를 출력합니다 =====

이름: 김채원

나이: 17세

국어점수: 90점

영어점수: 85점

복습 scanf_s로 학생 데이터 입력

===== 학생정보를 출력합니다 =====

이름 : 김채원

나이 : 17세

국어점수 : 90점

영어점수 : 85점

```
printf( "===== 학생정보를 출력합니다 =====\n" );  
printf( "이름 : %s\n", name );  
printf( "나이 : %d세\n", age );  
printf( "국어점수 : %d점\n", kor );  
printf( "영어점수 : %d점\n", eng );
```

복습 연산자를 이용한 총점, 평균 계산

산술 연산자

연산자	기능	사용 예
+	두 피연산자 값을 더합니다.	5 + 3
-	왼쪽 값 에서 오른쪽 값을 뺍니다.	10 - 4
*	두 값을 곱합니다.	6 * 8
/	왼쪽 값을 오른쪽 값으로 나눕니다.	9 / 3
%	왼쪽 값을 오른쪽 값으로 나누었을 때의 나머지를 구합니다.	9 % 2 (피연산자가 정수만 가능)

복습 연산자를 이용한 총점, 평균 계산

총점 :

국어점수 + 영어점수

===== 학생정보를 출력합니다 =====

이름: 김채원

나이: 17세

국어점수: 90점

영어점수: 85점

총점: 175점

복습 연산자를 이용한 총점, 평균 계산

===== 학생정보를 출력합니다 =====

이름: 김채원

나이: 17세

국어점수: 90점

영어점수: 85점

총점: 175점

```
printf("===== 학생정보를 출력합니다 =====\n");  
printf("이름 : %s\n", name);  
printf("나이 : %d세\n", age);  
printf("국어점수 : %d점\n", kor);  
printf("영어점수 : %d점\n", eng);  
printf("총점 : %d점\n", kor + eng);
```

복습 연산자를 이용한 총점, 평균 계산

평균 :

총점 나누기 2 (과목수)

===== 학생정보를 출력합니다 =====

이름: 김채원

나이: 17세

국어점수: 90점

영어점수: 85점

총점: 175점

평균: 87점

복습 연산자를 이용한 총점, 평균 계산

===== 학생정보를 출력합니다 =====

이름 : 김채원

나이 : 17세

국어점수 : 90점

영어점수 : 85점

총점 : 175점

평균 : 87점

```
printf("===== 학생정보를 출력합니다 =====\n");  
printf("이름 : %s\n", name);  
printf("나이 : %d세\n", age);  
printf("국어점수 : %d점\n", kor);  
printf("영어점수 : %d점\n", eng);  
printf("총점 : %d점\n", kor + eng);  
printf("평균 : %d점\n", (kor + eng) / 2);
```

복습 연산자를 이용한 총점, 평균 계산

===== 학생정보를 출력합니다 =====

이름 : 김채원

나이 : 17세

국어점수 : 90점

영어점수 : 85점

총점 : 175점

평균 : 87.5점

```
printf("===== 학생정보를 출력합니다 =====\n");  
printf("이름 : %s\n", name);  
printf("나이 : %d세\n", age);  
printf("국어점수 : %d점\n", kor);  
printf("영어점수 : %d점\n", eng);  
printf("총점 : %d점\n", kor + eng);  
printf("평균 : %.1f점\n", ((float)kor + eng) / 2);
```

복습 연산자를 이용한 총점, 평균 계산

===== 학생정보를 출력합니다 =====

이름 : 김채원

나이 : 17세

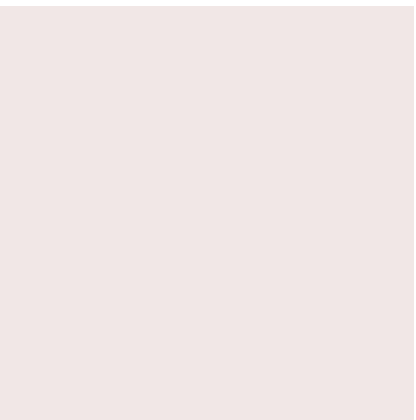
국어점수 : 90점

영어점수 : 85점

총점 : 175점

평균 : 87.5점

```
printf("===== 학생정보를 출력합니다 =====\n");  
printf("이름 : %s\n", name);  
printf("나이 : %d세\n", age);  
printf("국어점수 : %d점\n", kor);  
printf("영어점수 : %d점\n", eng);  
printf("총점 : %d점\n", kor + eng);  
printf("평균 : %.1f점\n", ((float)kor + eng) / 2);
```



1.

if 조건문

- 관계, 논리 연산자



```
if (조건식) {  
    조건이 참인 경우 실행되는 코드  
}
```


비교(관계) 연산자

종류	사용 예	설명	연산 예시	결과
==	$a == b$	a와 b가 같다	$5 == 8$	0 (거짓)
!=	$a != b$	a와 b가 다르다	$5 != 8$	1 (참)
>	$a > b$	a는 b보다 크다	$5 > 8$	0 (거짓)
<	$a < b$	a는 b보다 작다	$5 < 8$	1 (참)
>=	$a >= b$	a는 b보다 크거나 같다	$5 >= 8$	0 (거짓)
<=	$a <= b$	a는 b보다 작거나 같다	$5 <= 8$	1 (참)

조건식을 만드는 비교 연산자

종류	사용 예	설명	연산 예시	결과
==	a == b	a와 b가 같다	5 == 8	0 (거짓)
!=	a != b	a와 b가 다르다	5 != 8	1 (참)
>	a > b	a는 b보다 크다	5 > 8	0 (거짓)
<	a < b	a는 b보다 작다	5 < 8	1 (참)
>=	a >= b	a는 b보다 크거나 같다	5 >= 8	0 (거짓)
<=	a <= b	a는 b보다 작거나 같다	5 <= 8	1 (참)

```
int main(void) {  
    printf("결과 : %d", 5 != 8);  
}
```

조건식을 만드는 비교 연산자

종류	사용 예	설명	연산 예시	결과
==	a == b	a와 b가 같다	5 == 8	0 (거짓)
!=	a != b	a와 b가 다르다	5 != 8	1 (참)
>	a > b	a는 b보다 크다	5 > 8	0 (거짓)
<	a < b	a는 b보다 작다	5 < 8	1 (참)
>=	a >= b	a는 b보다 크거나 같다	5 >= 8	0 (거짓)
<=	a <= b	a는 b보다 작거나 같다	5 <= 8	1 (참)

```
int main(void) {  
    printf("결과 : %d", 'a' == 'A');  
}
```

조건식을 만드는 비교 연산자

종류	사용 예	설명	연산 예시	결과
==	a == b	a와 b가 같다	5 == 8	0 (거짓)
!=	a != b	a와 b가 다르다	5 != 8	1 (참)
>	a > b	a는 b보다 크다	5 > 8	0 (거짓)
<	a < b	a는 b보다 작다	5 < 8	1 (참)
>=	a >= b	a는 b보다 크거나 같다	5 >= 8	0 (거짓)
<=	a <= b	a는 b보다 작거나 같다	5 <= 8	1 (참)

```
int main(void) {  
    printf("결과 : %d", 'A' == 65);  
}
```

1 기본적인 if 문

```
int main(void) {  
    if (5 > 0) {  
        printf("조건식이 참인경우 출력");  
    }  
}
```

```
int main(void) {  
    int n = 10;  
  
    if (n > 0) {  
        puts("n은 0보다 크다.");  
    }  
}
```

```
int main(void) {  
    char blood; // 혈액형  
    scanf_s("%c", &blood, 1);  
    if (blood == 'a') {  
        printf("a형은 조용한 편이라고 합니다.\n");  
    }  
    if (blood == 'b') {  
        printf("b형은 다혈질이라고 해요\n");  
    }  
}
```

```
int main(void) {  
    int speed; // 차량의 속도  
    scanf_s("%d", &speed);  
    if (speed >= 60) {  
        printf("과속입니다.\n");  
    }  
}
```



```
if (조건식) {
```

```
    조건이 참인 경우 실행되는 코드
```

```
}
```

```
else {
```

```
    조건이 거짓인 경우 실행되는 코드
```

```
}
```

```
int main(void) {  
    int speed; // 차량의 속도  
    scanf_s("%d", &speed);  
    if (speed >= 60) {  
        printf("과속입니다.\n");  
    }  
    else {  
        printf("정상 속도입니다.\n");  
    }  
}
```

1 if / else if 문

```
if (조건식 1) {
```

조건식1이 참인 경우 실행되는 코드

```
}
```

```
else if (조건식 2) {
```

조건식1이 거짓 조건식 2가 참인 경우 실행되는 코드

```
}
```

1 if / else 문

```
int main(void) {  
    int speed; // 차량의 속도  
    scanf_s("%d", &speed);  
    if (speed > 100) {  
        printf("과속입니다.\n");  
    }  
    else if(speed < 50){  
        printf("너무 속도가 느립니다.\n");  
    }  
}
```



```
if (조건식 1) {
```

조건식1이 참인 경우 실행되는 코드

```
}
```

```
else if (조건식 2) {
```

조건식1이 거짓 조건식 2가 참인 경우 실행되는 코드

```
}
```

```
else {
```

모든 조건식이 거짓인 경우 실행되는 코드

```
}
```

1 if / else if / else 문

```
int main(void) {  
    int speed; // 차량의 속도  
    scanf_s("%d", &speed);  
    if (speed > 100) {  
        printf("과속입니다.\n");  
    }  
    else if(speed < 50){  
        printf("너무 속도가 느립니다.\n");  
    }  
    else {  
        printf("정상속도 입니다.\n");  
    }  
}
```



1 if / else if / else 문

```
int main(void) {  
    int n;  
    puts("정수를 입력하세요.");  
    scanf_s("%d", &n);  
  
    if (n > 0) {  
        puts("n은 양수입니다.");  
    }  
    else if (n < 0) {  
        puts("n은 음수입니다.");  
    }  
    else {  
        puts("n은 0 입니다.");  
    }  
}
```

총점 평균에 따라 A, B, C, D, F 등의 등급을 출력하고자 합니다.

1. 가장 먼저 프로그램에 필요한 변수와 상수를 선언합니다. (값은 키보드로 입력)

```
#include <stdio.h>
#define MAX_NAME 100

int main(void) {
    char name[MAX_NAME];
    int kor, eng, math;
    int mean; // 평균을 저장할 변수
```


2. 이어서 학생의 이름과 성적을 입력받는 scanf_s 코드를 작성합니다.

```
printf("학생의 이름을 입력하세요 : \n");  
scanf_s("%s", &name, MAX_NAME);  
printf("국어 영어 수학 성적을 입력하세요\n");  
scanf_s("%d %d %d", &kor, &eng, &math);
```

3. 입력받은 내용으로 평균점수를 구하고 출력하는 부분입니다.

```
mean = (kor + eng + math) / 3;  
printf("총점의 평균은 %d점 입니다.\n", mean);
```

4. if else 를 사용하여 평균점수에 따라 등급을 출력합니다.

등급	점수 구간	조건
A학점	90점 이상 100점 이하	입력한 점수가 90 ~ 100점이면 A 학점
B학점	80점 이상 90점 미만	입력한 점수가 80 ~ 89점이면 B 학점
C학점	70점 이상 80점 미만	입력한 점수가 70 ~ 79점이면 C 학점
D학점	60점 이상 70점 미만	입력한 점수가 60 ~ 69점이면 D 학점
F학점	0점 이상 60점 미만	입력한 점수가 0 ~ 60점이면 F 학점

4. if else 를 사용하여 평균점수에 따라 등급을 출력합니다.

```
if (mean >= 90) {  
    printf("A 등급입니다.");  
}  
else if (mean >= 80) {  
    printf("B 등급입니다.");  
} // 이하 생략
```

```
int main(void) {  
    if (22 > 50) {  
        printf("가장 큰 수 : %d\n", 22);  
    }  
    else {  
        printf("가장 큰 수 : %d\n", 50);  
    }  
}
```

1 숫자 변수 비교

```
int main(void) {  
    int num1 = 22;  
    int num2 = 55;  
  
    if (num1 > num2) {  
        printf("가장 큰 수 : %d\n", num1);  
    }  
    else {  
        printf("가장 큰 수 : %d\n", num2);  
    }  
}
```

두개의 숫자 중에 큰 수 찾기

아래와 같이 두개의 정수를 입력 받아서 큰 수를 출력하는 코드를 만들어보겠습니다.

두개의 수를 입력하세요: 40 22

가장 큰 수: 40

두개의 숫자 중에 큰 수 찾기

```
int main(void) {  
    int num1, num2;  
    printf("두개의 수를 입력하세요: ");  
    scanf_s("%d %d", &num1, &num2);  
    if (num1 > num2) {  
        printf("가장 큰 수 : %d\n", num1);  
    }  
    else {  
        printf("가장 큰 수 : %d\n", num2);  
    }  
}
```


1

세개의 수를 입력 받아 가장 큰 수를 찾기

아래와 같이 세개의 정수를 입력 받아서 가장 큰 수를 출력하는 코드를 만들어보겠습니다.

세개의 수를 입력하세요: 40 100 22

가장 큰 수: 100

세개의 수를 입력받아 가장 큰 수를 찾기

```
int num1, num2, num3;
int max; // 최대값을 저장
printf("세개의 수를 입력하세요: ");
scanf_s("%d %d %d", &num1, &num2, &num3);
if (num1 > num2) {
    max = num1;
}
else {
    max = num2;
}
if (max < num3) {
    max = num3;
}
printf("가장 큰 수: %d\n", max);
```

1

세개의 수를 입력 받아 가장 큰 수를 찾기

이번엔 세개의 수 중에서 가장 작은 수를 출력하려면 어떻게 해야 할까요?

세개의 수를 입력하세요: 40 100 22

가장 작은 수: 22

세개의 수를 입력 받아 가장 작은 수 찾기

```
int main(void) {
    int num1, num2, num3;
    int min;
    printf("세개의 수를 입력하세요: ");
    scanf_s("%d %d %d", &num1, &num2, &num3);
    if (num1 > num2) {
        min = num2;
    }
    else {
        min = num1;
    }
    if (num3 < min) {
        min = num3;
    }
    printf("가장 작은 수 : %d\n", min);
}
```

논리 연산자

종류	이름	사용 예	설명	결과
!	NOT	!(5 > 8)	조건식이 참이면 거짓, 거짓이면 참	1 (참)
&&	AND	(5 > 8) && (3 > 1)	조건식이 모두 참인 경우 참, 하나라도 거짓이면 거짓	0 (거짓)
	OR	(5 > 8) (3 > 1)	조건식 중 하나라도 참이면 참, 모두 거짓이면 거짓	1 (참)

조건식들을 검사하는 논리 연산자

종류	이름	사용 예	설명	결과
!	NOT	!(5 > 8)	조건식이 참이면 거짓, 거짓이면 참	1 (참)
&&	AND	(5 > 8) && (3 > 1)	조건식이 모두 참인 경우 참, 하나라도 거짓이면 거짓	0 (거짓)
	OR	(5 > 8) (3 > 1)	조건식 중 하나라도 참이면 참, 모두 거짓이면 거짓	1 (참)

```
int main(void) {  
    printf("결과 : %d", !(10 > 5));  
}
```

논리연산자는 두가지 이상의 조건을 동시에 검사할 때 사용됩니다.

예를 들어 알람이

일요일 아침 10:30 으로 설정되어 있는 경우

두가지 조건이 동시에 참이 되어야 합니다. (&& - AND 기호 사용)

일요일 체크

&&

10:30 시간 체크

논리연산자는 두가지 이상의 조건을 동시에 검사할 때 사용됩니다.

둘 중에 하나만 참이 되면 되는 경우도 있습니다.

예를 들어 토요일이거나 일요일이면 실행되는 코드가 있다면

두 조건 중 하나만 참이면 참이 되는 (||-OR 기호를 사용합니다)

토요일 체크

||

일요일 체크

논리연산자를 사용하여 아래의 조건을 검사해 보겠습니다.

등급	점수 구간	조건
A학점	90점 이상 100점 이하	입력한 점수가 90 ~ 100점이면 A 학점

```
if ((mean <= 100) && (mean >= 90)) {  
    printf("A 등급입니다.");  
}
```

if문의 중첩

if 문 안에 if문을 넣어서 같은 조건을 만들어 줄 수도 있습니다.

(하지만 if문이 여러 번 중첩되면 코드를 읽기 점점 힘들어 집니다

```
if ((mean <= 100) && (mean >= 90)) {  
    printf("A 등급입니다.");  
}
```

```
if (mean <= 100){  
    if (mean >= 90) {  
        printf("A 등급입니다.");  
    }  
}
```

어떤 문자를 입력받아 알파벳인지 확인하는 코드를 만들어 보겠습니다.

키보드입력

문자를 입력하세요:

a

입력된 문자:a

알파벳 문자입니다.

문자를 입력하세요:

\$

입력된 문자:\$

알파벳 문자가 아닙니다.

어떤 문자를 입력받아 알파벳인지 확인하는 코드를 만들어 보겠습니다.

```
int main(void) {  
    char c;  
    printf("문자를 입력하세요: ");  
    scanf_s("%c", &c, 1);  
    printf("입력된 문자: %c\n", c);  
}
```

if문 - 논리연산자 응용

아스키 코드를 이용하여 입력된 문자가 알파벳인지를 체크합니다.

소문자 a부터 z까지 아스키 코드 : 97 ~ 122

대문자 A부터 Z까지 아스키 코드 : 65 ~ 90

```
if (97 <= c && c <= 122) {  
    // 소문자  
}  
if (65 <= c && c <= 90) {  
    // 대문자  
}
```

if문 - 논리연산자 응용

소문자를 체크한 조건문과 대문자를 체크한 조건문 두가지를
동시에 체크할 때 논리 연산자를 사용할 수 있습니다.

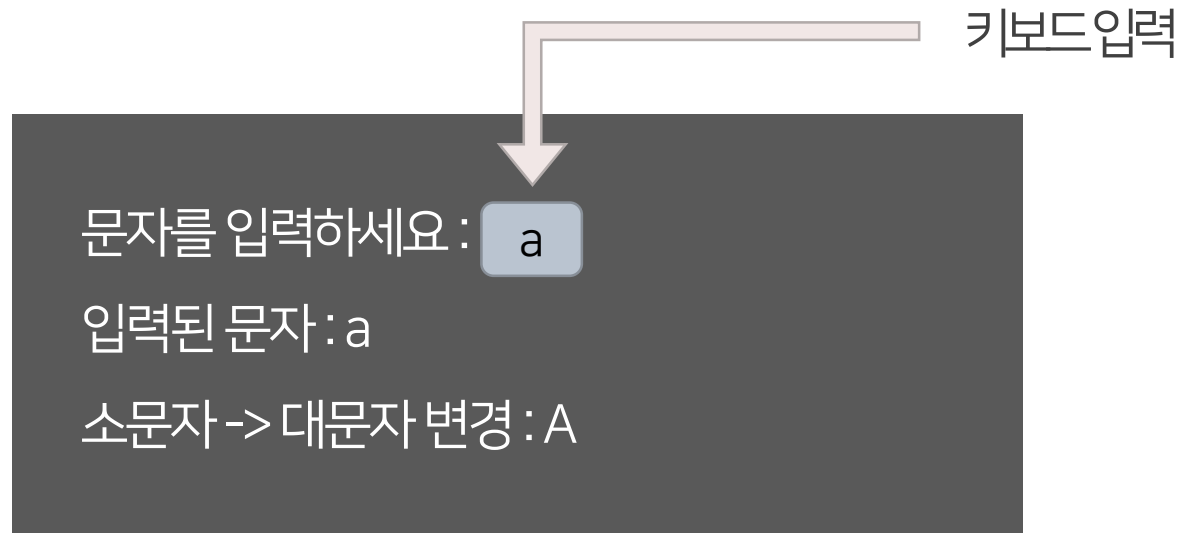
```
(97 <= c && c <= 122) || (65 <= c && c <= 90)
```

```
if (97 <= c && c <= 122) {  
    // 소문자  
}  
if (65 <= c && c <= 90) {  
    // 대문자  
}
```

대문자, 소문자가 되는 두 조건을 OR 연산자를 이용하여 알파벳인지 체크합니다.

```
if ((97 <= c && c <= 122) || (65 <= c && c <= 90)) {  
    printf("알파벳 문자입니다.\n");  
}  
else {  
    printf("알파벳 문자가 아닙니다.\n");  
}
```

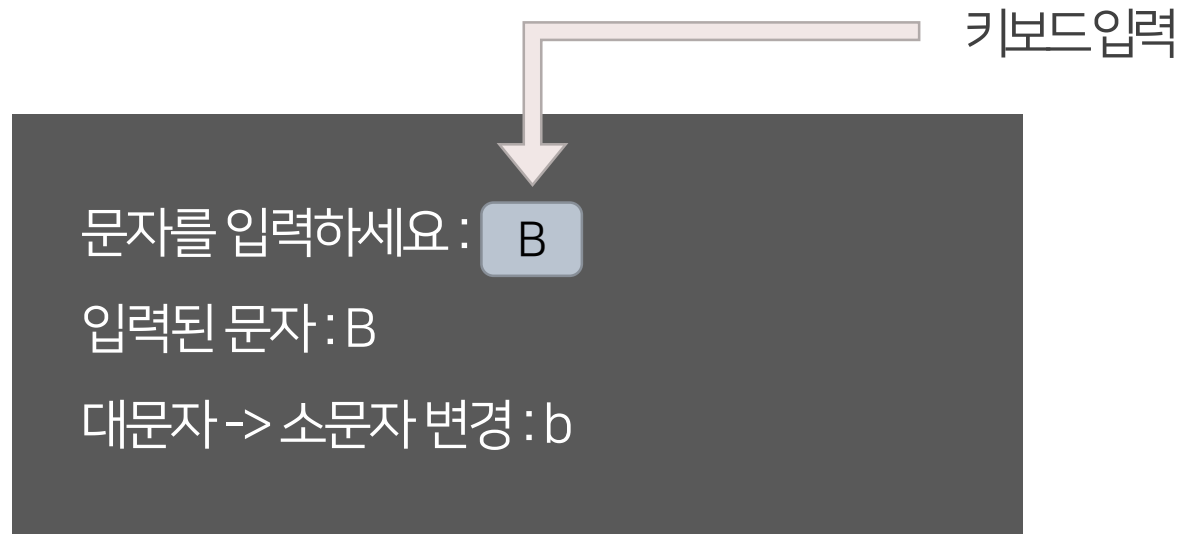
소문자가 입력되면 대문자로 변경해주는 코드를 작성해 보겠습니다..



소문자가 입력 됐을 때 대문자로 변경

```
int main(void) {  
    char c;  
    printf("문자를 입력하세요: ");  
    scanf_s("%c", &c, 1);  
    printf("입력된 문자: %c\n", c);  
    if (97 <= c && c <= 122) {  
        printf("소문자 -> 대문자 변경: %c\n", c - 32);  
    }  
    else if (65 <= c && c <= 90) {  
        // 대문자 입력된 경우  
    }  
    else {  
        printf("알파벳 문자가 아닙니다.\n");  
    }  
}
```

대문자가 입력된 경우에 소문자로 바뀌지도록 하려면 어떻게 해야할까요?



간단한 계산식을 입력받아 계산결과를 출력하는 프로그램을 만들어 보겠습니다.

수식을 입력하세요 :

14 + 8

계산결과 : 22

간단한 계산식을 입력받아 계산결과를 출력하는 프로그램을 만들어 보겠습니다.

```
int main(void) {  
    int num1, num2; // 계산할 두 수  
    char opt; // 산술연산자  
    printf("수식을 입력하세요: ");  
    scanf_s("%d %c %d", &num1, &opt, 1, &num2);  
}
```

```
int main(void) {  
    int num1, num2; // 계산할 두 수  
    char opt; // 산술연산자  
    printf("수식을 입력하세요: ");  
    scanf_s("%d %c %d", &num1, &opt, 1, &num2);  
    if (opt == '+') {  
        printf("계산결과: %d\n", num1 + num2);  
    }  
    if (opt == '-') {  
        printf("계산결과: %d\n", num1 - num2);  
    } // 곱셈, 나누기 생략  
}
```

중복되는 코드들을 줄이는 방향으로 코드를 만드는 것이 좋습니다.

```
if (opt == '+') {  
    printf("계산결과: %d\n", num1 + num2);  
}  
if (opt == '-') {  
    printf("계산결과: %d\n", num1 - num2);  
} // 곱셈, 나누기 생략
```

```
int main(void) {  
    int num1, num2; // 계산할 두 수  
    int result; // 계산결과를 저장  
    char opt; // 산술연산자  
    printf("수식을 입력하세요: ");  
    scanf_s("%d %c %d", &num1, &opt, 1, &num2);  
    if (opt == '+') {  
        result = num1 + num2;  
    }  
    if (opt == '-') {  
        result = num1 - num2;  
    }  
    printf("계산결과: %d\n", result);  
}
```

1 if와 else if 차이

```
int main(void) {  
    int num1, num2; // 계산할 두 수  
    int result; // 계산결과를 저장  
    char opt; // 산술연산자  
    printf("수식을 입력하세요: ");  
    scanf_s("%d %c %d", &num1, &opt, 1, &num2);  
    if (opt == '+') {  
        result = num1 + num2;  
    }  
    else if (opt == '-') {  
        result = num1 - num2;  
    }  
}
```

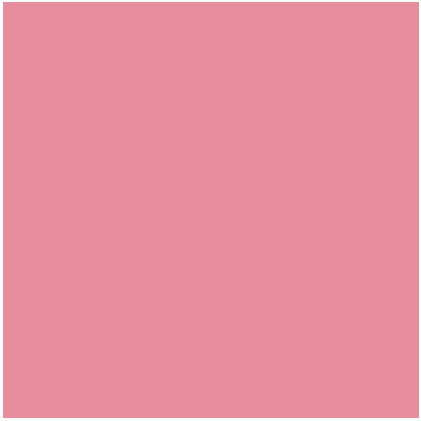

1 if와 else if 차이

if문은 언제나 조건을 체크합니다, 즉 opt가 + 일때도 - (마이너스) 인지 체크합니다.

```
if (opt == '+') {  
    result = num1 + num2;  
}  
if (opt == '-') {  
    result = num1 - num2;  
}
```

else if문은 이전의 if 문이 거짓일 때만 체크합니다. (opt가 + 인 경우에는 else if 문의 나머지 조건을 검사하지 않습니다)

```
if (opt == '+') {  
    result = num1 + num2;  
}  
else if (opt == '-') {  
    result = num1 - num2;  
}
```



2.

switch 조건문



1 switch문의 구조

```
switch (비교값) {
```

```
case 값1 :
```

(비교값 == 값1) 인 경우 실행되는 코드

```
break;
```

```
default:
```

비교값이 어떤 case에도 맞지 않는 경우 실행되는 코드

```
}
```

1 switch문의 구조

```
int main(void) {  
    int n = 10;  
    switch (n) {  
        case 10:  
            printf("10\n");  
            break;  
        case 100:  
            printf("100\n");  
            break;  
        default:  
            printf("default\n");  
    }  
}
```

```
int main(void) {  
    int n = 10;  
    switch (n) {  
        case 10:  
            printf("10");  
        case 100:  
            printf("100");  
            break;  
        default:  
            printf("default");  
    }  
}
```

case에 break가 없다면 아래 case로 이동 하여 실행합니다.

따라서 왼쪽의 코드는 10도 출력되고 100도 출력됩니다.

```
int main(void) {  
    int n = 0;  
    switch (n) {  
        case 10:  
            printf("10");  
        case 100:  
            printf("100");  
            break;  
        default:  
            printf("default");  
    }  
}
```

어떠한 case 에도 해당되지 않을 때
default문이 실행 됩니다.

왼쪽 코드는 default 가 출력됩니다.

이번에는 switch를 이용하여 계산결과를 출력하는 프로그램을 만들어 보겠습니다.

수식을 입력하세요 :

계산결과 : 22

```
int main(void) {
    int num1, num2;
    int result;
    char opt;
    printf("수식을 입력하세요: ");
    scanf_s("%d %c %d", &num1, &opt, 1, &num2);
    switch (opt) {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        default:
            printf("연산자가 잘못되었습니다.\n");
    }
    printf("결과 : %d", result);
}
```


1. if문은 switch문 보다 더 다양한 조건식을 검사할 수 있습니다.
2. switch의 실행 속도가 더 빠르고 코드를 읽기 쉽습니다.
3. case가 많지 않은 경우는 if 문으로 작성하는게 간단합니다.

```
if (opt == '+') {  
    result = num1 + num2;  
}  
else if (opt == '-') {  
    result = num1 - num2;  
}
```

```
switch (opt) {  
case '+':  
    result = num1 + num2;  
    break;  
case '-':  
    result = num1 - num2;  
    break; // 나누기, 곱 생략  
}
```

2 switch 문으로 변환

이전에 if else 로 만들었던 아래와 같은 조건을
switch 문으로 구현하려면 어떻게 해야 할까요?

등급	점수 구간	조건
A학점	90점 이상 100점 이하	입력한 점수가 90 ~ 100점이면 A 학점
B학점	80점 이상 90점 미만	입력한 점수가 80 ~ 89점이면 B 학점
C학점	70점 이상 80점 미만	입력한 점수가 70 ~ 79점이면 C 학점
D학점	60점 이상 70점 미만	입력한 점수가 60 ~ 69점이면 D 학점
F학점	0점 이상 60점 미만	입력한 점수가 0 ~ 60점이면 F 학점

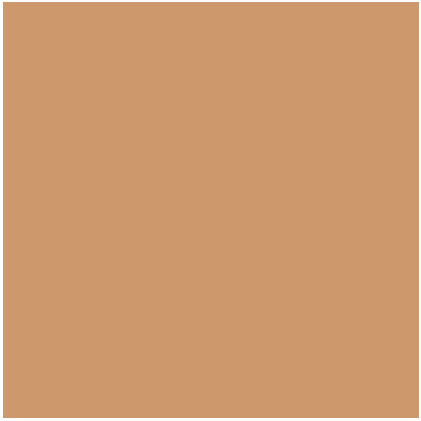
```
int main(void) {  
    int total = 95;  
    switch (total / 10) {  
        case 9:  
            printf("A학점");  
            break;  
        case 8:  
            printf("B학점");  
            break;  
        default :  
            printf("F학점");  
    }  
}
```

if 문을 switch 를 이용하여 변환할 수 있는 경우도 있고 불가능 할 때도 있습니다. (복잡한 조건식의 경우)

왼쪽코드의 경우는 9로 시작되는 점수 (95점, 90 점 등) 는 A학점을 줄 수 있고 8로 시작되는 (83점, 88점 등) 은 B를 줄 수 있기때문에 나누기로 몫을 구해 switch로 변경할 수 있었습니다.

```
int main(void) {  
    int total = 100;  
    switch (total / 10) {  
        case 9:  
            printf("A학점");  
            break;  
        case 8:  
            printf("B학점");  
            break;  
        default :  
            printf("F학점");  
    }  
}
```

왼쪽 코드처럼 점수가 100 점일 때 F 학점이 나오지 않도록 코드를 수정해 보세요
Hint : break를 사용하지 않으면
밑의 case가 이어져서 실행되는 것을 이용



3.

while 반복문



```
while (조건식) {  
    조건식이 참인경우 반복되는 코드  
}
```

```
do {  
    무조건 한번 실행 후 조건식이 참인경우 반복되는 코드  
} while (조건식);
```

3 while문의 구조

항상 참이되는 1 을 조건식에 넣으면 무한히 반복되는
무한반복문이 됩니다.

프로그램 강제종료 : Ctrl + c

```
int main(void){  
    while (1) {  
        printf("hello\n");  
    }  
}
```

특정 횟수만큼 반복

```
int main(void){  
    int n = 1;  
    while (n <= 10) {  
        printf( "%d번째 출력\n", n);  
        n = n + 1;  
    }  
}
```


3 while문의 사용

특정 값을 이용하여 반복을 중단시키는 경우

```
int main(void) {  
    int n = 1;  
    int cmd = 1; // 계속 할지 명령을 저장  
    while (cmd == 1) {  
        printf("%d번째 출력\n", n);  
        n = n + 1;  
        printf("계속은 1, 종료는 0 : ");  
        scanf_s("%d", &cmd);  
    }  
}
```

3 복합 대입 연산자

연산을 더 간단하게 표현하는 복합 대입 연산자

```
int main(void) {  
    int n = 1;  
    int cmd = 1; // 계속 할지 명령을 저장  
    while (cmd == 1) {  
        printf("%d번째 출력\n", n);  
        n += 1; // n = n + 1 과 동일  
        printf("계속은 1, 종료는 0 : ");  
        scanf_s("%d", &cmd);  
    }  
}
```

복합 대입 연산자

종류	사용 예	기본형	의미
<code>+=</code>	<code>n += 1</code>	<code>n = n + 1</code>	n에 1을 더한 값을 n에 대입
<code>-=</code>	<code>n -= 5</code>	<code>n = n - 5</code>	n에서 5를 뺀 값을 n에 대입
<code>*=</code>	<code>n *= 4</code>	<code>n = n * 4</code>	n에 4를 곱한 값을 n에 대입
<code>/=</code>	<code>n /= 10</code>	<code>n = n / 10</code>	n을 10으로 나눈 값을 n에 대입
<code>%=</code>	<code>n %= 2</code>	<code>n = n % 2</code>	n을 2로 나누고 남은 나머지 값을 n에 대입

```
int main(void) {  
    int target = 70; // 정답 숫자  
    int num;  
    printf("숫자를 맞춰보세요 : ");  
    scanf_s("%d", &num);  
    if (target > num) {  
        puts("UP");  
    }  
    else if (target < num) {  
        puts("DOWN");  
    }  
    else if (target == num) {  
        puts("숫자를 맞췄습니다.");  
    }  
}
```

숫자 맞추기 while 반복

```
int main(void) {  
    int target = 70; // 정답 숫자  
    int num;  
    printf("숫자를 맞춰보세요 : ");  
    scanf_s("%d", &num);  
    if (target > num) {  
        puts("UP");  
    }  
    else if (target < num) {  
        puts("DOWN");  
    }  
    else if (target == num) {  
        puts("숫자를 맞췄습니다.");  
    }  
}
```


while (조건식) {

}

```
int target = 70; // 정답 숫자
int num;
while (target != num) {
    printf("숫자를 맞춰보세요 : ");
    scanf_s("%d", &num);
    if (target > num) {
        puts("UP");
    }
    else if (target < num) {
        puts("DOWN");
    }
    else if (target == num) {
        puts("숫자를 맞췄습니다!!!");
    }
}
```

숫자 맞추기 while 반복

```
int target = 70; // 정답 숫자
int num;
while (target != num) {
    printf("숫자를 맞춰보세요 : ");
    scanf_s("%d", &num);
    if (target > num) {
        puts("UP");
    }
    else if (target < num) {
        puts("DOWN");
    }
    else if (target == num) {
        puts("숫자를 맞췄습니다!!!");
    }
}
```



num 값이 scanf_s로
입력받기 전이라
비교할 수가 없음

숫자 맞추기 do while반복

```
int target = 70; // 정답 숫자
int num;
do {
    printf("숫자를 맞춰보세요 : ");
    scanf_s("%d", &num);
    if (target > num) {
        puts("UP");
    }
    else if (target < num) {
        puts("DOWN");
    }
    else if (target == num) {
        puts("숫자를 맞췄습니다!!!");
    }
} while (target != num);
```

무조건 한번 do 블록이
실행 된 이후에
조건문을 검사하여 반복

숫자 맞추기 do while반복

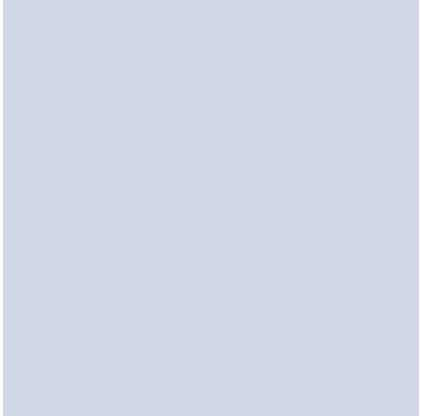
```
int target = 70; // 정답 숫자
int num;
do {
    printf("숫자를 맞춰보세요 : ");
    scanf_s("%d", &num);
    if (target > num) {
        puts("UP");
    }
    else if (target < num) {
        puts("DOWN");
    }
    else if (target == num) {
        puts("숫자를 맞췄습니다!!!");
    }
} while (target != num);
```

무조건 한번 do 블록이
실행 된 이후에
조건문을 검사하여 반복

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

int main(void) {
    srand(time(NULL));
    int target = rand() % 100;
    // 랜덤한 수 (0 ~ 99)
    int num;
    do {
        printf("숫자를 맞춰보세요 : ");
```

stdlib.h 의 rand() 함수는 임의의 숫자부터 시작하여 0~32767 사이의 숫자를 생성합니다. 하지만 처음 시작되는 숫자(seed) 가 항상 같아서 항상 같은 숫자가 생성됩니다. 따라서 srand()를 이용하여 시작되는 숫자를 현재의 시간으로 제공하면 완전히 무작위의 숫자를 생성할 수 있습니다. 현재시간을 초단위로 제공하는 time.h의 time(NULL) 함수 사용



4. for 반복문



```
for ( 초기값 ; 조건식 ; 증감식 )  
{  
    조건식이 참일 때 반복할 코드  
}
```

```
int main(void){
    int i;
    for (i = 0; i < 10; i++) {
        printf( "%d\n", i);
    }
}
```

사용 예	의미
i++	변수 i 를 사용한 뒤 1 증가시킴
++i	변수 i 를 1 증가시킨 뒤 사용
i--	변수 i 를 사용한 뒤 1 감소시킴
--i	변수 i 를 1 감소시킨 뒤 사용

```
int main(void) {  
    int i = 5;  
    printf("i : %d\n", i--);  
    printf("i : %d", i);  
}
```

특정 횟수만큼 반복할 때

```
int main(void) {  
    for (int i = 0; i < 10; i++) {  
        printf("%d번째 실행\n", i);  
    }  
}
```

나중에 다룰 배열과 같은 자료구조를 순회하기 위해 주로 사용

```
int main(void){
    int i;
    int a[5] = { 1, 2, 3, 4, 5 };
    for (i = 0; i < 5; i++) {
        printf("%d\n", a[i]);
    }
}
```


2 x 1 부터 2 x 9 까지 for 문을 이용하여 출력하는 코드입니다.

```
int main(void) {  
    int dan; // 구구단의 단  
    int mul; // 단에 곱해질 수 (1 ~ 9)  
    dan = 2;  
    for (mul = 1; mul < 10; mul++){  
        printf("%d x %d = %d\n", dan, mul, dan * mul);  
    }  
}
```

반복문에서 **break** 를 사용하면 반복문을 탈출합니다.

```
int main(void) {  
    int dan; // 구구단의 단  
    int mul; // 단에 곱해질 수 (1 ~ 9)  
    dan = 2;  
    for (mul = 1; mul < 10; mul++){  
        if (mul == 7) {  
            break;  
        }  
        printf("%d x %d = %d\n", dan, mul, dan * mul);  
    }  
}
```

반복문에서 `continue`를 사용하면 반복문의 처음으로 돌아갑니다.

```
int main(void) {  
    int dan; // 구구단의 단  
    int mul; // 단에 곱해질 수 (1 ~ 9)  
    dan = 2;  
    for (mul = 1; mul < 10; mul++){  
        if (mul == 5) {  
            continue;  
        }  
        printf("%d x %d = %d\n", dan, mul, dan * mul);  
    }  
}
```

블록 { ... } 을 생성하는 모든 문법들은 중첩될 수 있습니다.
이를 이용하여 2x1 부터 9x9 까지 쉽게 출력할 수 있습니다.

```
int main(void) {  
    int dan; // 구구단의 단  
    int mul; // 단에 곱해질 수 (1 ~ 9)  
    for (dan = 2; dan < 10; dan++) {  
        for (mul = 1; mul < 10; mul++) {  
            printf("%d x %d = %d\n", dan, mul, dan * mul);  
        }  
    }  
}
```

중첩에서 break / continue의 범위 - 가장 가까운 반복문에만 영향을 미칩니다.

```
int main(void) {  
    int dan; // 구구단의 단  
    int mul; // 단에 곱해질 수 (1 ~ 9)  
    for (dan = 2; dan < 10; dan++) {  
        for (mul = 1; mul < 10; mul++) {  
            if (mul == 5) {  
                break;  
            }  
            printf("%d x %d = %d\n", dan, mul, dan * mul);  
        }  
    }  
}
```