

c언어 세미나

목차

01

조건문

- if~else문
- switch~case

02

반복문

- while문
- do~while문
- for문

03

함수와 변수

- 함수란?
- 함수의 장점
- 함수 원형
- 함수의 유효범위

04

배열

- 배열 선언
- 배열 초기화
- 다차원 배열

05

포인터

- 포인터
- 간접 참조 연산자*
- 포인터의 연산
- 포인터와 배열
- 포인터와 함수

06

과제 구동영상

- 베스킨라빈스31
- 진수변환
- 0~999 정렬
- 다이아몬드 별찍기

01. 조건문

-if~else 문
-switch~case 문

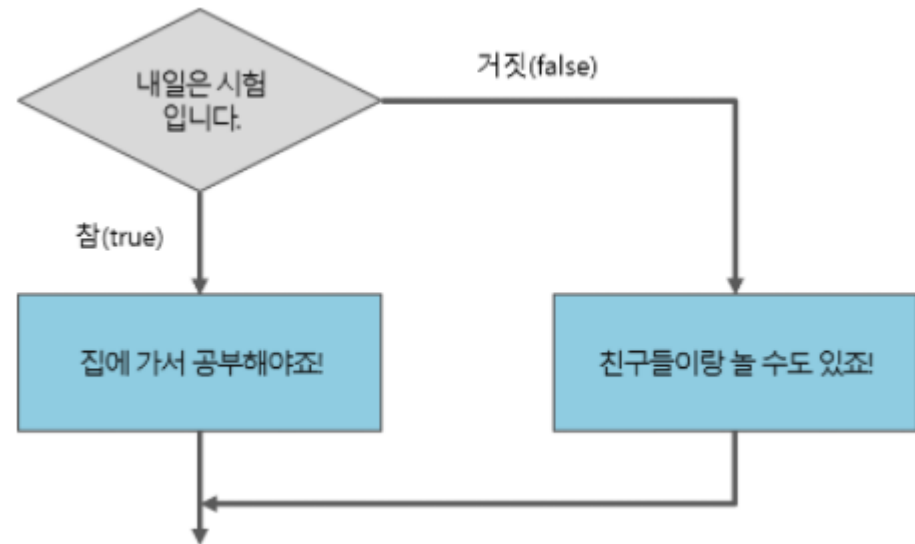
01. 조건문

문법

```
if (조건식)
{
    조건식의 결과가 참일 때 실행하고자 하는 명령문;
}

else
{
    조건식의 결과가 거짓일 때 실행하고자 하는 명령문;
}
```

순서도

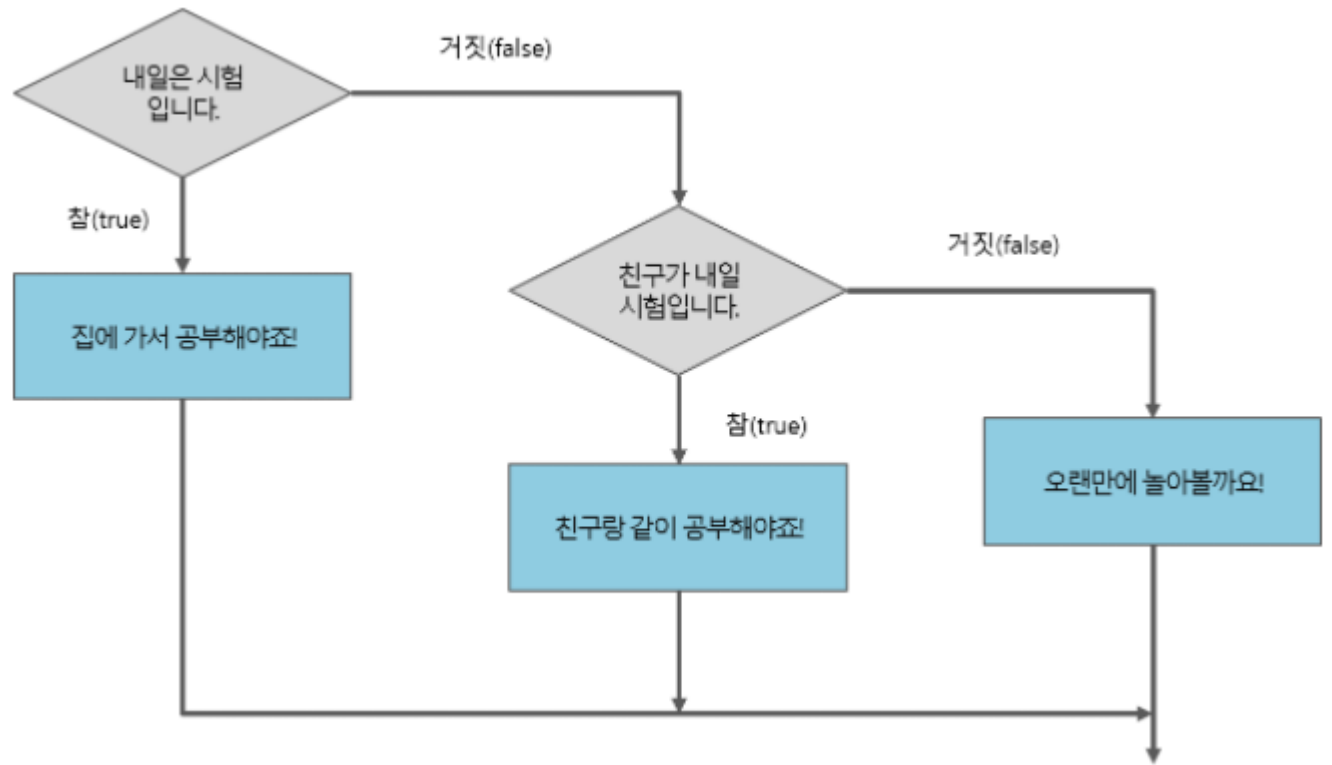


01. 조건문

문법

```
if (조건식1)
{
    조건식1의 결과가 참일 때 실행하고자 하는 명령문;
}
else if (조건식2)
{
    조건식2의 결과가 참일 때 실행하고자 하는 명령문;
}
else
{
    조건식1의 결과도 거짓이고, 조건식2의 결과도 거짓
    일 때 실행하고자 하는 명령문;
}
```

순서도



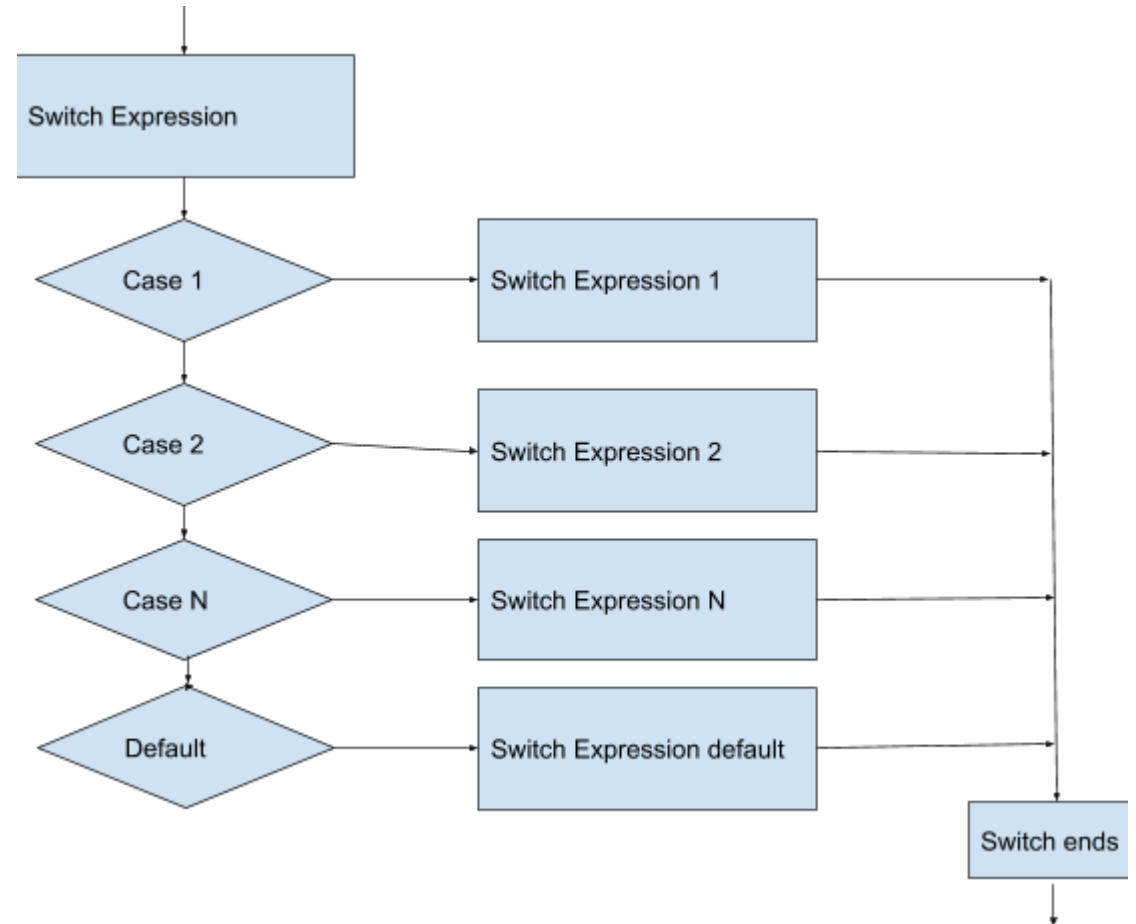
01.조건문

문법

switch (조건 값)

```
{  
    case 값1:  
        조건 값이 값1일 때 실행하고자 하는 명령문;  
        break;  
    case 값2:  
        조건 값이 값2일 때 실행하고자 하는 명령문;  
        break;  
    ...  
    default:  
        조건 값이 어떠한 case 절에도 해당하지 않을  
        때 실행하고자 하는 명령문;  
        break;  
}
```

순서도



02. 반복문

- while 문
- do~while 문
- for 문

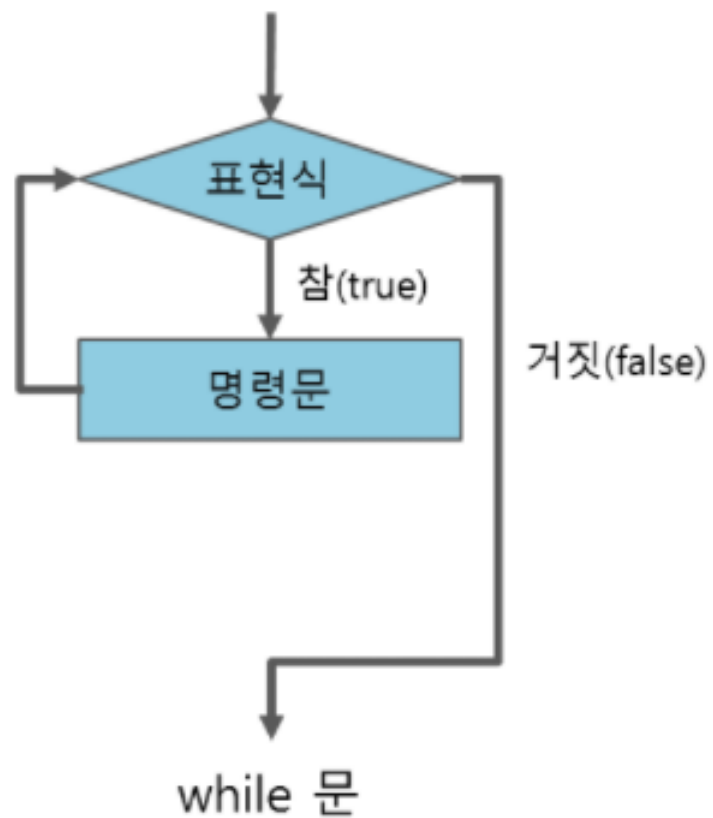
02. 반복문

소스

```
while (조건식)
```

```
{  
    조건식의 결과가 참인 동안 반복적으로 실행하고자  
    하는 명령문;  
}
```

순서도

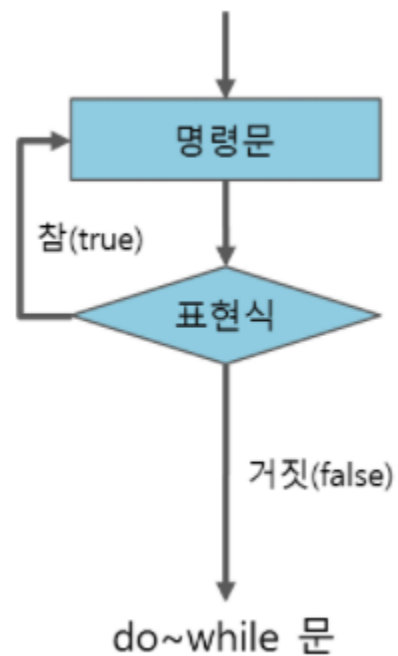


02. 반복문

소스

```
do {  
    표현식의 결과가 참인 동안 반복적으로 실행하고자  
    하는 실행문;  
} while (표현식);
```

순서도



02. 반복문

소스

```
for (초기식; 조건식; 증감식) {
```

 표현식의 결과가 참인 동안 반복적으로 실행하고자
 하는 실행문;

```
}
```

ex)

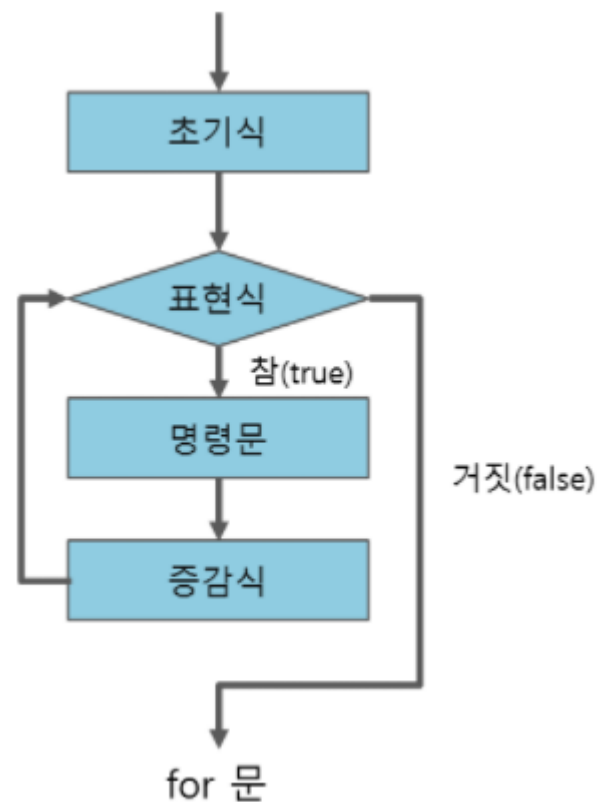
```
for (int x=0 ; x<5; x++) {  
    printf(“%d ”,x);
```

```
}
```

result)

0 1 2 3 4

순서도



03. 함수와 변수

- 함수란?
- 함수의 장점
- 함수 원형
- 변수의 유효범위

03. 함수와 변수

함수란?

함수(function)란 하나의 특별한 목적의 작업을 수행하기 위해 독립적으로 설계된 프로그램 코드의 집합으로 정의할 수 있습니다.

C 프로그램은 이러한 함수들로 구성되며, 포함된 함수들을 사용하여 프로그램의 목적을 달성하게 됩니다.

C언어에서 함수는 크게 라이브러리 함수와 사용자 정의 함수로 구분할 수 있습니다.

라이브러리 함수

-컴파일러에서 지원되는 함수
ex) printf(),scanf()

사용자 정의 함수

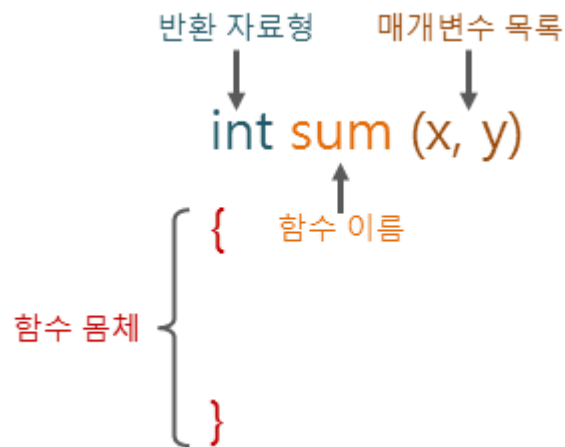
-프로그래머 자신이 만들어서 사용하는 함수
ex) momo(),mumu()

03. 함수와 변수

함수의 장점

1. 함수를 사용하면 코드가 중복되는 것을 막을 수 있다.
2. 한번 사용한 함수는 재사용이 가능하다.
3. 함수를 사용하면 전체 프로그램을 모듈로 나눌 수 있어서 개발 과정이 쉬워지고 체계적이게 되면서 유지보수도 쉬워진다.

함수의 정의



1. 반환 타입
 - 함수가 작업을 마치고 반환하는 타입
2. 함수 이름
 - 함수를 호출하기 위한 이름
3. 매개변수 목록
 - 함수 호출시 전달되는 인수의 값을 저장할 변수
4. 함수 몸체
 - 함수의 고유 기능을 수행하는 명령문들의 집합

03. 함수와 변수

함수의 원형선언

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

```
int add(int i, int n){  
    return (i+n);  
}
```

```
int main(void){  
    int i = 1, y = 2;  
    printf("%d", add(i,n));  
    return 0;  
} //컴파일 오류
```



위치 변경

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

```
int add(int i, int n); //함수의 원형선언
```

```
int main(void){  
    int i = 1, y = 2;  
    printf("%d", add(i,n));  
    return 0;  
}
```

```
int add(int i, int n){  
    return (i+n);  
}
```

03. 함수와 변수

변수의 유효범위

지역변수 - 블록내에서 선언된 변수

```
ex) int main(){  
    int i =0;}
```

전역변수 - 함수의 외부에서 선언된 변수

```
ex) #include<stdio.h>  
int momo()  
int v=1; //전역 변수 선언  
...  
  
int main()  
int momo()
```

정적변수(static) - 지역 변수와 전역 변수의 특징을 모두 가진 함수

```
void sub(void)  
int main(void){  
    int i;  
    for(i=0;i<3;i++)  
        sub();  
    return 0;  
}  
  
void sub(void){  
    int auto_count =0; //함수종료시 소멸  
    static int static_count =0; //함수종료되어도 자동소멸 x  
    auto_count++; //이전값 유지x  
    static_count++; //이전값 유지되면서 증가  
    printf("auto_count=%d\n", auto_count);  
    printf("static_count=%d\n", static_count);  
}
```

실행결과

```
auto_count=1  
static_count=1  
auto_count=1  
static_count=2  
auto_counts=1  
static_count=3
```

04. 배열

- 배열 선언
- 배열 초기화
- 다차원 배열

-배열선언

- ✓자료형 배열이름[배열크기];
- ✓Ex) int grade [5];
- ✓배열크기는 양의 정수만 가능
- ✓배열원소
- ✓인덱스
- ✓배열원소의 접근

grade[0]	grade[1]	grade[2]	grade[3]	grade[4]
----------	----------	----------	----------	----------

- 각각의 데이터는 정수로 된 번호에 의해서 접근
- 대량의 데이터 효율적 처리

-배열 초기화

- ✓ 자료형 배열이름[배열크기] = { , , }
- ✓ `int grade[5]={1,2,3};`
- ✓ `int grade[]={1,2};`
- ✓ `int grade[5]={0};`

grade[0]	grade[1]	grade[2]	grade[3]	grade[4]
----------	----------	----------	----------	----------

-다차원 배열

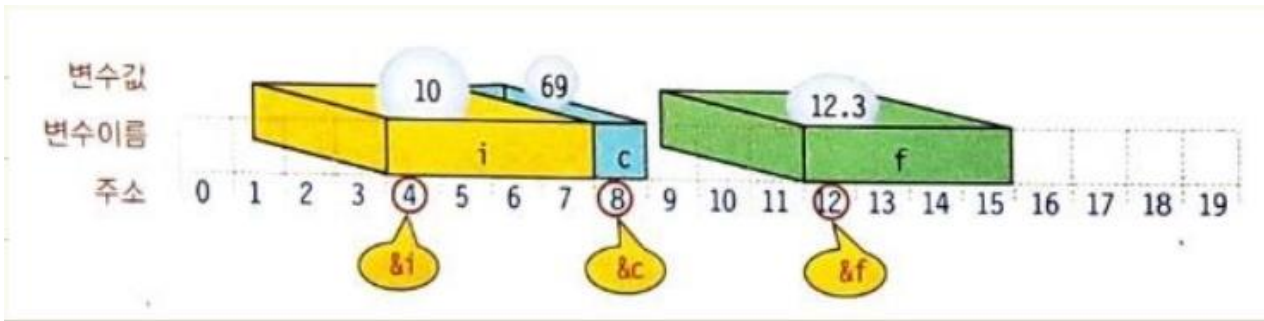
- ✓ 2차원 배열: 2개의 인덱스(행번호, 열번호)
- ✓ Ex) `int arr[3][3]={`
- ✓ `{1,2,3},`
- ✓ `{4,5,6},`
- ✓ `{7,8,9}};`

arr[0][0] 1	arr[0][1] 2	arr[0][2] 3
arr[1][0] 4	arr[1][1] 5	arr[1][2] 6
arr[2][0] 7	arr[2][1] 8	arr[2][2] 9

05. 포인터

- 포인터
- 간접 참조 연산자*
- 포인터의 연산
- 포인터와 배열
- 포인터와 함수

-포인터

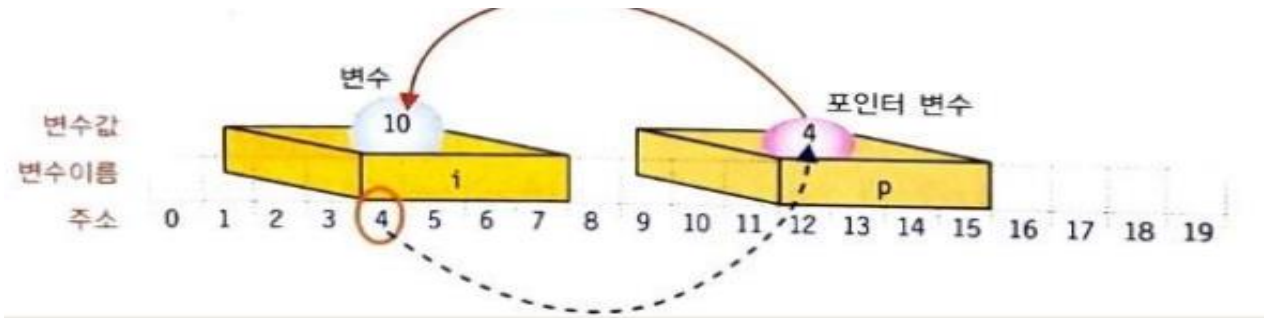


주소연산자 : &

Ex)

```
int i;  
printf("%u",&i);
```

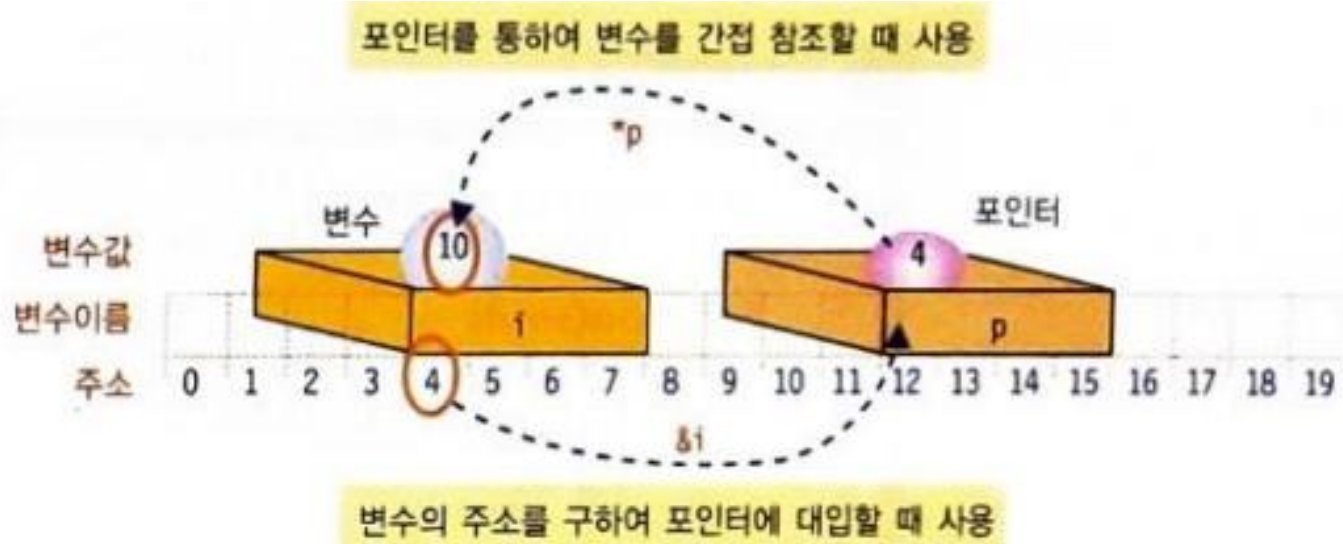
포인터(pointer): 변수의 주소를 가지고 있는 변수



Ex)

```
int i=0;  
int *p=&i;
```

-간접 참조 연산자 *



Ex)

```
Int x,i=10;
```

```
*p=&i;
```

```
x=i;
```

```
x=*p;
```

```
printf("%d",i);
```

```
*p=20;
```

```
printf("%d",i);
```

-포인터의 연산

- 간접 참조 연산자와 증감연산자

수식	의미
<code>v = *p++</code>	p가 가리키는 값을 v에 대입한 후에 p를 증가한다.
<code>v = (*p)++</code>	p가 가리키는 값을 v에 대입한 후에 p가 가리키는 값을 증가한다.
<code>v = *++p</code>	p를 증가시킨 후에 p가 가리키는 값을 v에 대입한다.
<code>v = ++*p</code>	p가 가리키는 값을 가져온 후에 그 값을 증가하여 v에 대입한다.

- 포인터의 형변환

Ex)

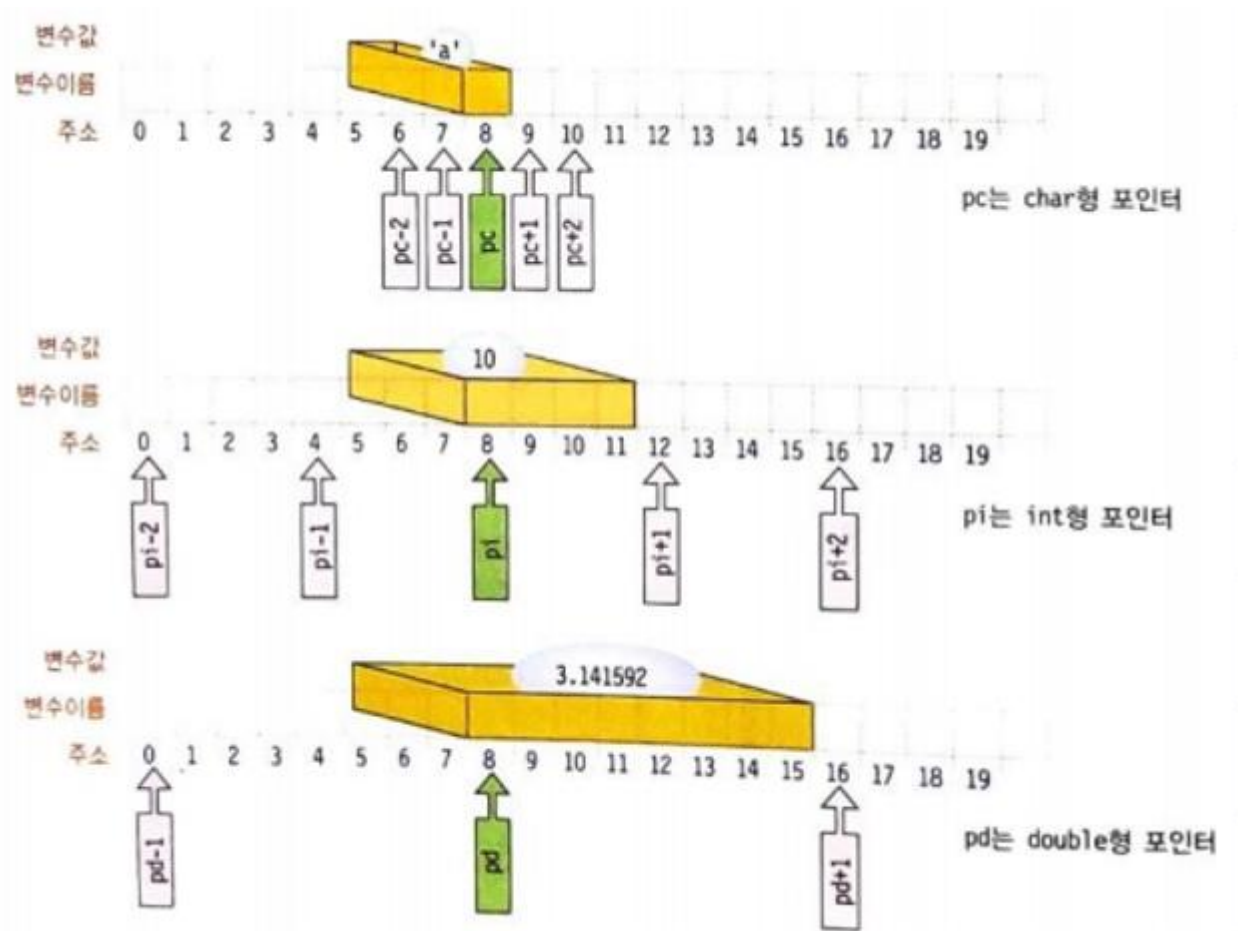
```
double *pd=&f;
```

```
int *pi;
```

```
pi=(int*)pd; // double 형의 포인터를 int형으로 명시적 변환
```

-포인터의 연산

- 덧셈, 뺄셈 : 자료형크기(S)일때,
정수N만큼 증감시 포인터 값은
 $S*N$ 만큼 증감



Ex)

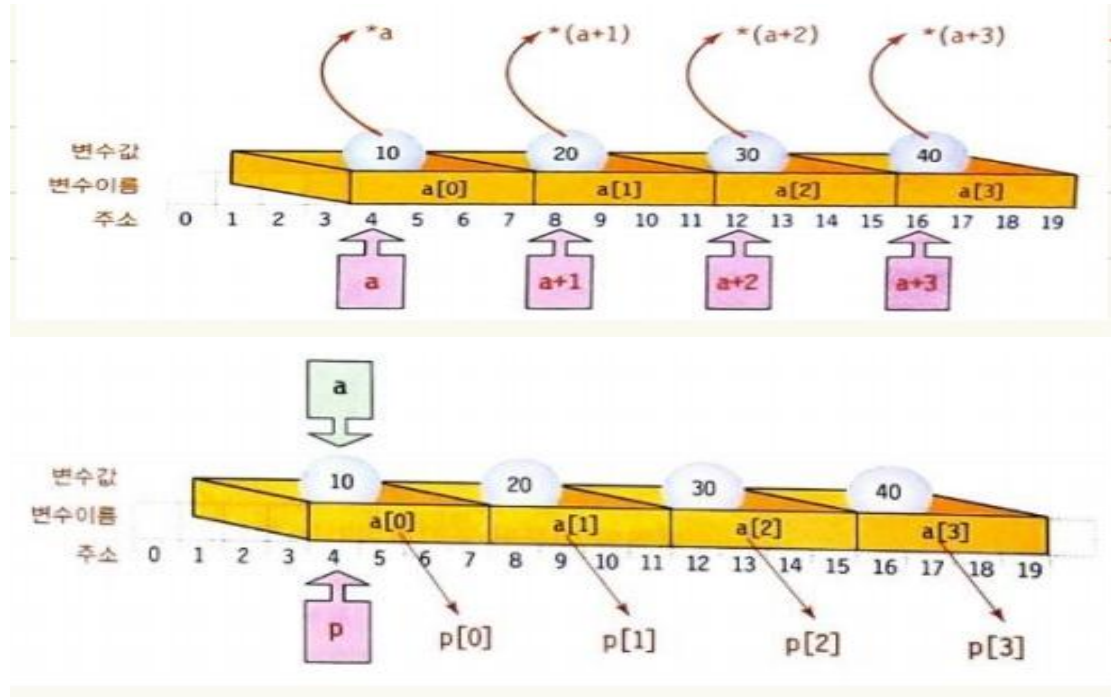
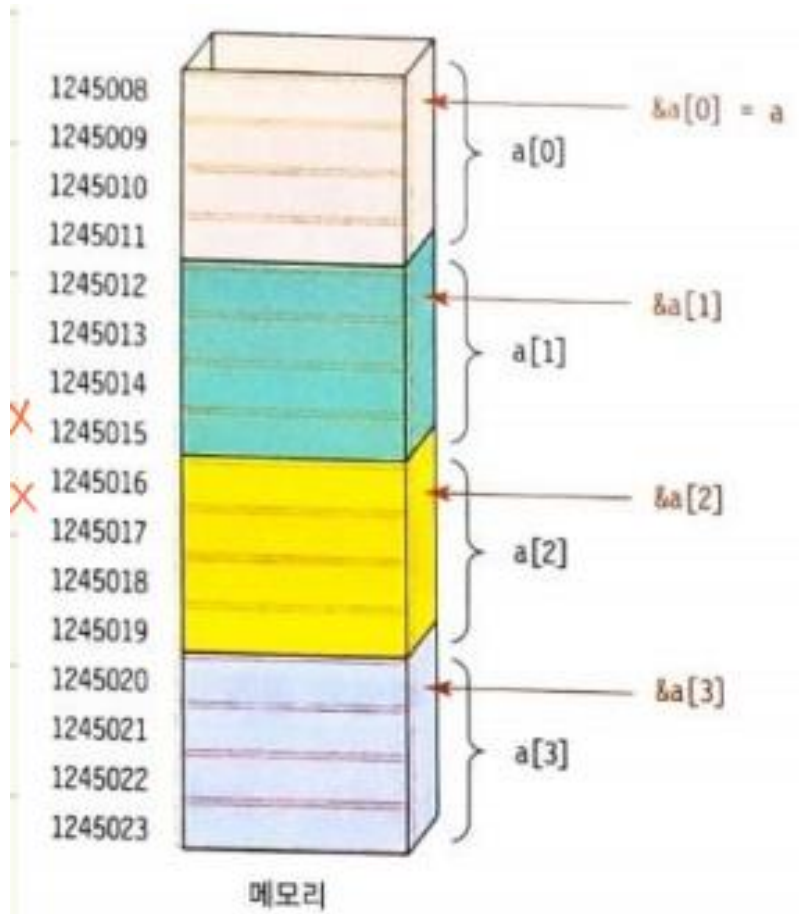
```
char *pc=8;
pc++;    pc--;    pc+2;
```

Ex)

```
int *pi=8;
pi++;    pi--;    pi+2;
```

-포인터와 배열

- 배열이름 = 포인터 상수



Ex)

```
int a[]={10,20,30,40};
```

```
int *p;
```

```
p=a;
```


-포인터와 함수

- 함수 호출시 인수 전달 방식
값에 의한 호출 : 복사본 전달, C에서 지원

Ex)

```
#include<stdio.h>
void swap(int x, int y);
int main(){
    int a=100,b=200;
    printf("a=%d b=%d\n",a,b);
    swap(a,b);
    printf("a=%d b=%d\n",a,b);
    return 0;
}
```

```
void swap(int x, int y){
    int tmp;
    printf("x=%d y=%d\n",x,y);
    tmp =x;
    x=y;
    y=tmp;
    printf("x=%d y=%d\n",x,y);
}
```

-포인터와 함수

포인터에 의한 호출 : 원본이 전달, 참조에 의한 호출을 간접구현

Ex)

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

```
void swap(int *px, int *py);
```

```
int main()
```

```
{
```

```
    int a=100,b=200;
```

```
    printf("a=%d b=%d\n",a,b);
```

```
    swap(&a,&b);
```

```
    printf("a=%d b=%d\n",a,b);
```

```
    return 0;
```

```
}
```

```
void swap(int *px, int *py)
```

```
{
```

```
    int tmp;
```

```
    tmp =*px;
```

```
    *px=*py;
```

```
    *py=tmp;
```

```
}
```

06. 과제

- 베스킨라빈스 31
- 진수변환
- 0~999 정렬
- 다이아몬드 별 찍기

06. 과제

3개의 문제는 각각 따로가 아닌 **하나의 소스 내에서 구동**되어야 한다.

구동 방식은 1, 2, 3을 입력 하였을 시 그 문항으로 이동해 진행한다.

문제 1	베스킨 31 게양, 31을 부른 player or computer는 패배
문제 2	2진수 입력할 시 8진수 10진수 16진수 변환 출력
문제 3	0~999 범위에 원하는 만큼의 수를 입력하고, 내림차순 오름차순 정렬
문제 4	<p>다이아몬드를 제작하라.</p> <p>다이아몬드의 크기를 입력 받아 예시 사진과 같이 출력되도록 한다.</p> <p>최소 값 5 이상 입력하도록 한다.</p> <p>(정수가 아닌 다른 수를 입력할 시 다시 입력하도록 한다.)</p> 