1-1문제)

텍스트, 스크린샷, 폰트, 화이트이(가) 표시된 사진

자동 생성된 설명

**답 ) char< unsigned char< short < int < long < unsigned(int) < float < double**

float : 4byte

char : 1byte

unsigned (int) : 4byte

unsigned char : 1byte

double : 8byte

int : 4byte

long : 8byte

short : 2byte

1-2문제)

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

**15의 10진수 : 15**

**15의 2의 보수 : 11110001**

스케치, 도표, 화이트, 그림이(가) 표시된 사진

자동 생성된 설명

먼저 2진수(00001111)로 바꿔준 뒤 비트를 반전시켜준다. 그러면 1의 보수가 11110000이 되고 1를 더해줘서 11110001이 2의 보수가 된다.

**15의 16진수 : 0x0F**

16진수에서는 A가 10진수의 10이 되고 B(11), C(12), D(13), E(14), F(15)이므로 F이다.

**-9의 10진수 : -9**

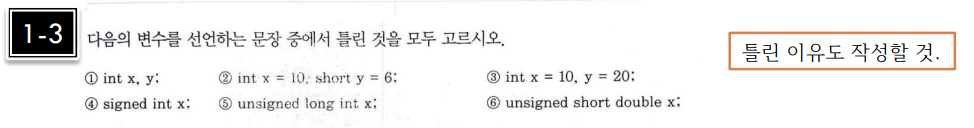
**-9의 2의 보수 : 01110111**

8비트는 8자리의 이진수를 의미하므로 컴퓨터는 음수 표현을 위해 가장 왼쪽에 한 비트를 부호로 사용하기 때문에 -9의 2진수는(10001001)로 바꿔준 뒤 비트를 반전시켜준다. 그러면 1의 보수가 01110110가 되고 1를 더해줘서 01110111이 2의 보수가 된다.

**-9의 16진수 : -0x09**

9이하의 16진수는 10진수와 다를 바가 없다.

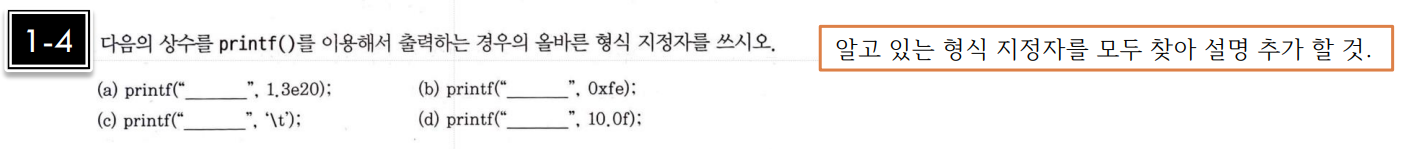
1-3문제)

****

**답 ) 2, 6번**

**이유 )** 2번이 맞으려면 int x = 10; short y = 6;이 되어야 한다. 자료형이 다른데 콤마만으로 저렇게 선언할 수는 없다. 4번에서 unsigned short double이라는 자료형은 C에서 지원하지 않는 자료형이므로 틀렸다.

1-4문제)

****

**답 ) (a) %f or %lf (b) %x (c) %c (d) %f**

%d = 부호 있는 10진수 정수형 출력(int형 변수)

%f = 소수점 고정형 표기 형식으로 출력(float형 변수)

%lf = double형 실수를 소수점 고정형 표기로 출력

%c = 하나의 문자 출력

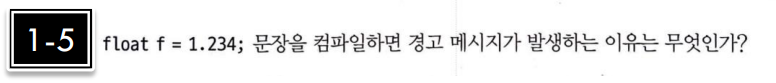
%s = 문자열 출력

%o = 부호가 없는 8진수 출력

%x = 부호가 없는 16진수 출력

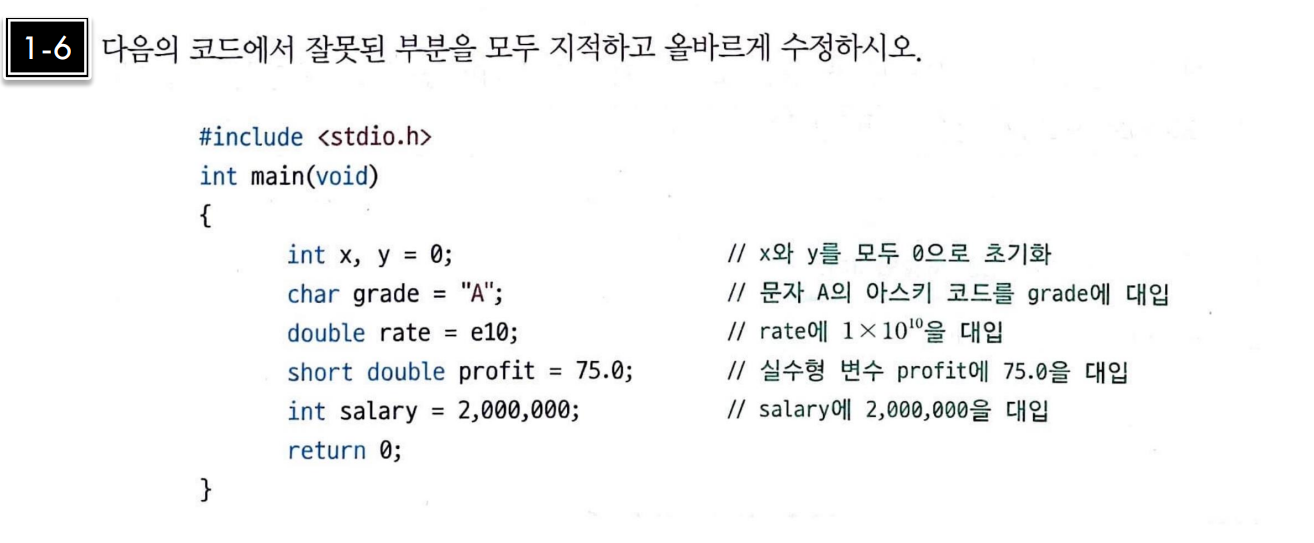
%e = 지수 표기 형식으로 출력

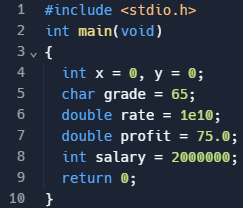
1-5문제)



1.234가 기본적으로 double 데이터 형식으로 인식되기 때문이다. 그럴 경우 형 변환을 통해 float f = (float)1.234; 선언하거나 형식 접미사 ‘f’를 추가하면 해결된다. float f = 1.234f; 사실 이 정도 경고 메시지는 안 띄우는 경우도 있다.

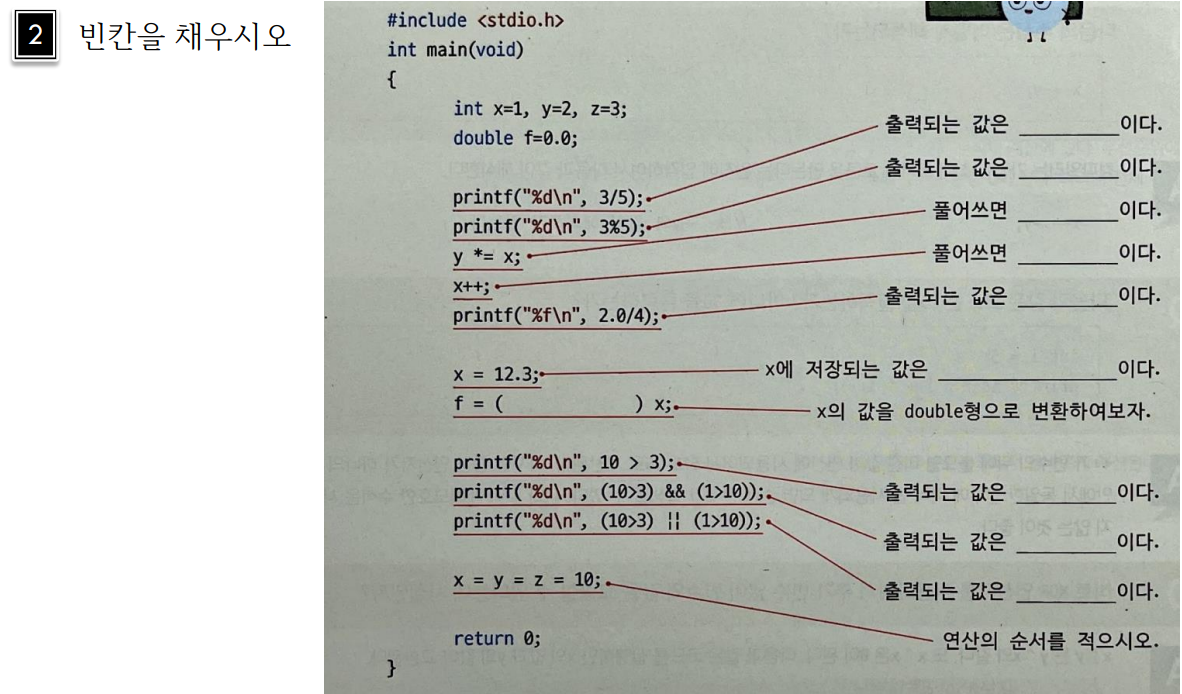
1-6문제)



**답)** 

**풀이)** 문제와 같이 int x, y = 0; 이라고 하면 x에는 쓰레기값이 들어가고 y만 0이 된다. 문자 A의 아스키 코드를 대입하라고 했으므로 A의 아스키 코드인 65를 사용해 선언했다. (그런데 문제의 의도가 이게 맞는지 모르겠다). 10^10은 C언어에서 1e10과 같이 선언한다. Short double이라는 자료형은 존재하지 않기 때문에 double profit = 75.0; 으로 선언한다. 정수를 선언할 때는 숫자에 ‘,’을 붙이지 않는다.

2문제)



**printf("%d\n", 3/5); -> 출력되는 값은 0이다.**

3을 5로 나누면 몫이 0 이기 때문이다.

**printf("%d\n", 3%5); -> 출력되는 값은 3이다.**

3을 5로 나누면 나머지가 3이기 때문이다.

**y \*= x -> 풀어쓰면 y = y \* x이다.**

**x++ -> 풀어쓰면 x = x + 1이다.**

**printf(“%f\n”, 2.0/4); -> 출력되는 값은 0.50000이다.**

소수점이 어디까지 출력될지 범위를 정해주지 않았기 때문이다.

**x = 12. 3; -> 저장되는 값은 12이다.**

int로 저장했기 때문에 소수점 이하는 버려진다.

**f = (double) x;**

**printf(“%d\n”, 10 > 3); -> 출력되는 값은 1이다.**

참이기 때문에 1이 출력된다.

**printf(“%d\n”, (10>3) && (1>10)); -> 출력되는 값이 0이다.**

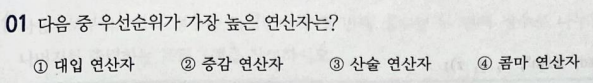
and 연산자이기 때문에 모두 참이어야 참인데 1>10은 참이 아니므로 거짓인 0 이 출력된다.

**printf(“%d\n”, (10>3) || (1>10)); -> 출력되는 값은 1이다.**

or 연산자이기 때문에 하나만 참이어도 참인데 10>3은 참이므로 1이 출력된다.

**x = y = z = 10; -> 연산 순서는 z = 10, y = 10, x =10 순**이다.

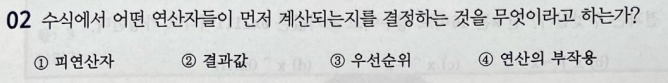
**3-1문제)**



**답 ) 2번**

**풀이)** 빠른 순으로 나열하면 증감 > 산술 > 대입 > 콤마가 된다.

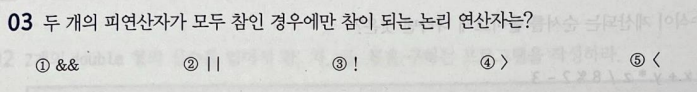
**3-2문제)**



**답 ) 3번**

**풀이)** 1번과 같은 우선순위에 따라 뭐가 먼저 계산되는지 결정된다.

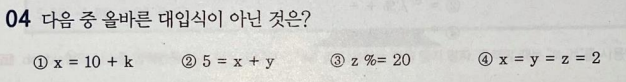
**3-3문제)**



**답 ) 1번**

**풀이)** AND 연산자의 경우 모두 참이어야 참이 된다.

**3-4문제)**



**답 ) 2번**

**풀이) 변수에 값을 넣어야 되는데 상수 값에 변수를 넣을 수는 없다.**

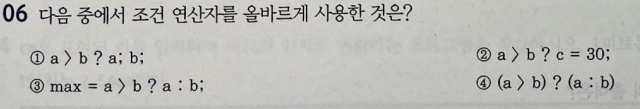
**3-5문제)**



**답 ) (a) 1.5 (b) 1.0 (c) 1.5 (d) 1.5**

**풀이 ) 나누기가 먼저 계산되므로 이렇게 된다 하지만 (b)에서는 1/2의 몫이 소수점을 버리면 0이 되기 때문에 값이 1.0이 된다.**

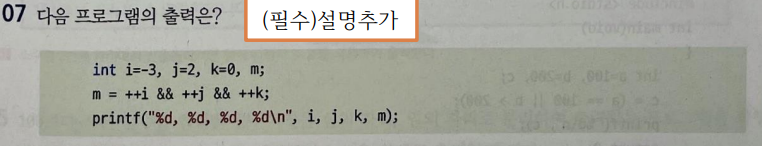
**3-6문제)**



**답 ) 3번**

**풀이 ) 3번은 최대값을 계산하는 문장인데 올바르게 사용했다.**

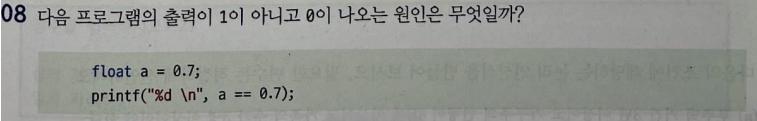
**3-7문제)**

****

**답 ) -2, 3, 1, 1**

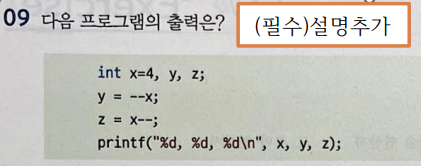
**풀이)** 변수 i, j, k는 먼저 더해주고 그 값에서 비교를 하게 된다. 그래서 0이 아닌 값들이기 때문에 참이 되어 m은 1이 된다. 만약 ++k가 아닌 k++였다면 0이 있기 때문에 거짓이 되어 m은 0이 되었을 것이다.

**3-8문제)**

****

**답 )** 현재 프로그래밍 부동 소수점 방식은 모두 IEEE 754 표준을 따르고 있는데 정확한 실수 값을 표현하지 못하는 경우가 있다. 그래서 보통 프로그래밍을 할 때는 소수점을 잘 이용하지 않으려고 하는 경우가 많다.

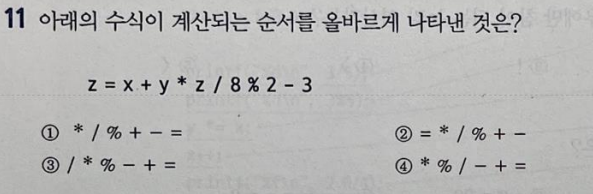
**3-9문제)**



**답 ) 2, 3, 3**

**풀이 )** 일단 x의 경우 두 번 증감연산자로 인해 감소되었으므로 2가 된다. y의 경우 x가 4일 때 먼저 감소가 한번 된 후에 y에 대입되기 때문에 3이 된다. 그렇게 3의 값이 된 x에서 먼저 z의 값에 들어간 이후 x가 한번 더 감소하기 때문에 z는 3이 된다.

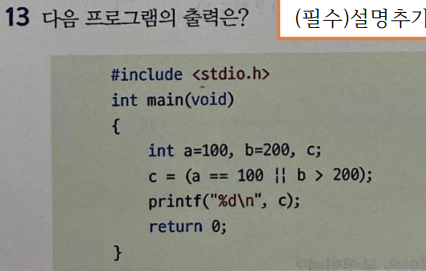
**3-11문제)**



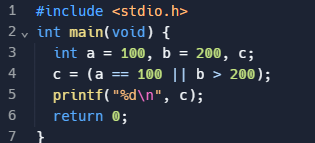
**답 ) 1번**

**풀이 ) 수학과 같이 곱하기, 나누기, 모듈라 연산은 +-보다 먼저한다.**

**3-13문제)**

****

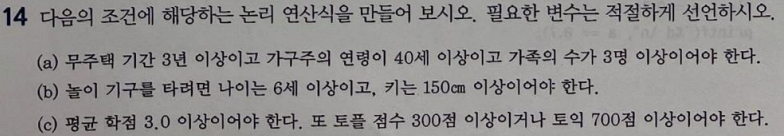
**소스코드)**

****

**실행화면)**

****

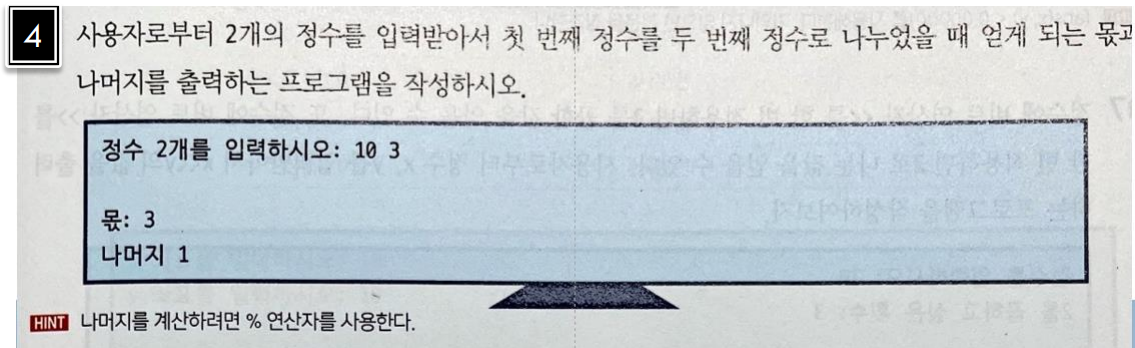
**설명)** OR 연산자는 둘 중에 하나라도 참이면 참이 된다 a가 100인 것은 참이고 b가 200보다 큰 것은 거짓이기 때문에 참이 된다. 참은 1, 거짓은 0의 값이 되기 때문에 c의 값은 1이 된다.

****

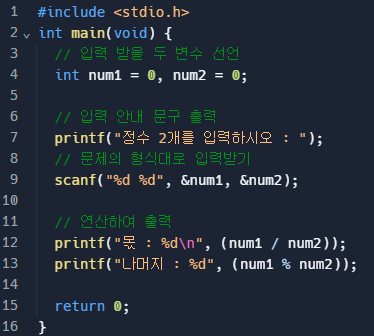
**답 )**

1. **(no\_house\_years >= 3) && (age >= 40) && (family\_members >= 3)**
2. **(age >= 6) && (height >= 150)**
3. **(gpa >= 3.0) && ((tofel >= 300) || (toeic >= 700))**

**4문제)**

****

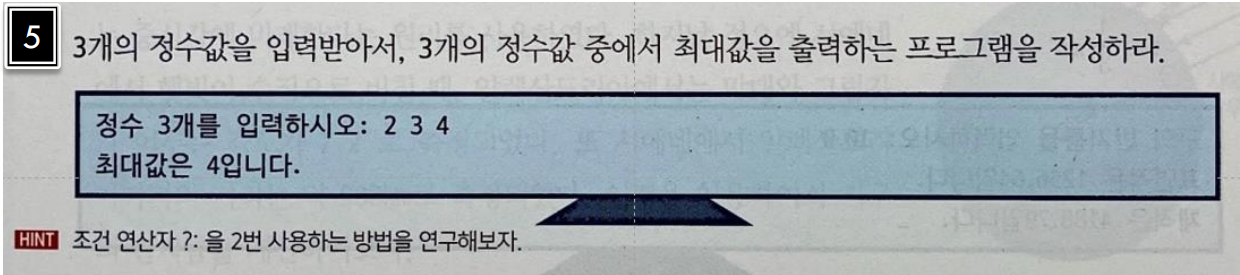
**소스코드)**

****

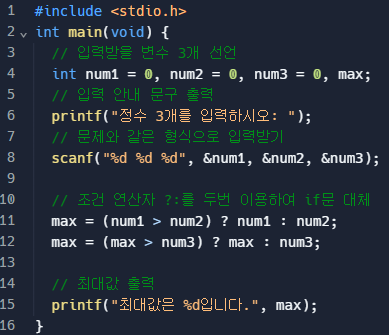
**실행화면)**

****

**5문제)**

****

**소스코드)**

****

**실행화면)**

****