

## 2021-1 프로그래밍언어 실습 2

### 2.1 10 진수 숫자의 역순 출력

- 1) 최대 10자리수이내의 10진수를 입력 받고, 그 10진수 숫자를 역순으로 출력하는 프로그램의 pseudo code를 작성하라.
- 2) 위 pseudo code를 기반으로 C 프로그램을 작성하고, 정확한 실행 결과를 확인하라.
- 3) Visual studio의 break point 설정 및 trace 기능을 사용하여, 위 C 프로그램의 중간 실행과정에서의 지역 변수 값을 확인하라.

<프로그램 실행 예>

```
Input positive decimal number (upto 10 digits): 123456789
Digits in reverse order: 9 8 7 6 5 4 3 2 1
```

```
Input positive decimal number (upto 10 digits): 345
Digits in reverse order: 5 4 3
```

### 2.2 16진수 데이터의 10진수, 8진수, 2진수 표현과 Bit-wise 계산

- 1) 정수형 (int) 데이터의 값을 32비트단위로 출력하는 함수 printInt\_Bits(int d)를 작성하라.
- 2) 0x000000 ~ 0xFFFF 범위의 값을 가지는 16진수 2개를 입력 받고, 이들 16진수 데이터 값의 8진수, 10진수 및 2진수 값을 출력하라.
- 3) 입력된 2개의 16진수 값의 덧셈과 뺄셈을 계산하고, 계산 결과값의 16진수 및 2진수 값을 각각 출력하라.
- 4) 16진수 2개 (a, b)를 입력받아 bitwise AND ( a & b), bitwise OR ( a | b), bitwise XOR ( a ^ b), shift\_left ( a << 3), shift\_right ( b >> 3), bitwise NOT (~b)를 각각 계산하고, 계산 결과 값의 16진수 및 2진수 값을 각각 출력하라.

<프로그램 실행 예: 16 진수 입력>

```
Input first hexadecimal number in (0x00000000 ~ 0xFFFFFFFF): 0x1234ABCD
x = 0x1234ABCD = (in decimal) 305441741 = (in octal) 2215125715
= (in bits) 00010010 00110100 10101011 11001101
Input second hexadecimal number in (0x00000000 ~ 0xFFFFFFFF): 0x87654321
y = 0x87654321 = (in decimal) -2023406815 = (in octal) 20731241441
= (in bits) 10000111 01100101 01000011 00100001

Bitwise AND : 0x1234ABCD & 0x87654321 => 0x02240301
00010010 00110100 10101011 11001101
& 10000111 01100101 01000011 00100001
-----
00000010 00100100 00000011 00000001

Bitwise OR : 0x1234ABCD | 0x87654321 => 0x9775EBED
00010010 00110100 10101011 11001101
| 10000111 01100101 01000011 00100001
-----
10010111 01110101 11101011 11101101

Bitwise XOR : 0x1234ABCD ^ 0x87654321 => 0x9551EBEC
00010010 00110100 10101011 11001101
^ 10000111 01100101 01000011 00100001
-----
10010101 01010001 11101000 11101100

Bitwise Shift Left : 0x1234ABCD << 0x3 => 0x91A55E68
00010010 00110100 10101011 11001101
<< 00000000 00000000 00000000 00000011
-----
10010001 10100101 01011110 01101000

Bitwise Shift Right : 0x87654321 >> 0x3 => 0xF0ECAB64
10000111 01100101 01000011 00100001
>> 00000000 00000000 00000000 00000011
-----
11110000 11101100 10101000 01100100

Bitwise NOT : ~0x87654321 => 0x87654321
~ 10000111 01100101 01000011 00100001
-----
01111000 10011010 10111100 11011110
```

## <Oral Test>

Q2.1 정수 (integer) 데이터를 10진수, 8진수, 16진수로 출력하기 위한 printf() 포맷 인 %d, %o, %x에 대하여 설명하라. 8진수는 접두어 0으로 시작하도록 하고, 16진수는 접두어 0x로 시작하도록 하며, 빈 자리에 0 채우기를 하는 방법에 대하여 설명하라.

(Key points: 출력 공간, 소수점 이하 출력 자리 수, prefix (접두어) 출력, 숫자 앞의 빈자리에 0 채우기 등 기능에 대한 설명)

Q2.2 정수 (integer) 데이터를 2진수 문자열 (binary digit string)로 출력하기 위한 알고리즘 (printInt\_Bits(int data))의 pseudo code를 작성하고, 이에 대하여 설명하라.

(Key points: int 자료형에 포함된 32개 비트를 차례로 출력할 수 있는 bitwise 연산 기능 사용)

Q2.3 2진수 데이터의 bit-wise AND, bit-wise OR, bit-wise XOR, shift left, shift right 계산을 예를 들어 설명하라. 음수 (negative number)의 shift right에서 sign bit 부분이 어떻게 처리되는가에 대하여 설명하라. (Key points: printInt\_Bits() 함수를 사용)

Q2.4 컴퓨터 시스템에서 2의 보수 (2's compliment)로 음의 정수 (음수, negative integer)값을 표현하는 방법에 대하여 설명하고, 정수형 변수 (integer variable)에 overflow가 발생하는 상황에 대하여 설명하라. (Key points: 2의 보수 (2's compliment)에 대하여 설명하고, <limits.h>에서 정의하고 있는 INT\_MAX의 비트 표현과 INT\_MAX + 1의 비트 패턴, INT\_MIN과 INT\_MIN - 1의 비트 패턴에 대하여 설명, printInt\_Bits() 함수를 사용하여 출력하여 확인.)