**Lab 02**

|  |
| --- |
| 그림입니다. 원본 그림의 이름: YU_UI_RGB-10.png 원본 그림의 크기: 가로 2256pixel, 세로 3047pixel 프로그램 이름 : Adobe ImageReady |

|  |  |
| --- | --- |
| 과목명 | 프로그래밍 언어 |
| 교수님 | 김영탁 교수님 |
| 이 름 | 김주환 |
| 학 번 | 21812158 |
| 일 자 | 2021.03.11.목 |

1. **10진수 숫자의 역순 출력**
   1. 최대 10자리수 이내의 10진수를 입력받고, 그 10진수 숫자를 역순으로 출력하는 프로그램의 pseudo code를 작성하라.

Procedure decimal\_reverse(void)

1. long num = 0;
2. int buf = 0;
3. input num;
4. while (num > 0) {
5. buf = num % 10;
6. num /= 10;
7. printout buf;
8. }

End of Procedure decimal\_reverse(void)

* 1. 위 pseudo code를 기반으로 C 프로그램을 작성하고, 정확한 실행 결과를 확인하라.

#include <stdio.h>

int main(void) {

long num = 0;

int buf = 0;

printf("Input positive decimal number (upto 10 digits) : ");

scanf("%d", &num);

printf("Digits in revers order : ");

while (num > 0) {

buf = num % 10;

num /= 10;

printf("%d ", buf);

}

return 0;

}



* 1. Visual studio의 break point 설정 및 trace 기능을 사용하여, 위 C 프로그램의 중간 실행과정에서의 지역 변수 값을 확인하라.

1. **16진수 데이터의 10진수, 8진수, 2진수 표현과 Bit-wise 계산**
   1. 정수형 데이터의 값을 32비트단위로 출력하는 함수 printInt\_Bits(int d)를 작성하라.

#define bit\_mask 0x01

#define num\_bits\_int 32

void printint\_bits(int d) {

unsigned long l1, l2, msk;

int n = num\_bits\_int - 1;

while (--n >= 0) {

msk = bit\_mask << n;

l1 = d & msk;

l2 = l1 >> n;

printf("%d", l2);

if ((n % 8) == 0) {

printf(" ");

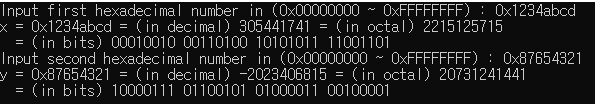
}

}

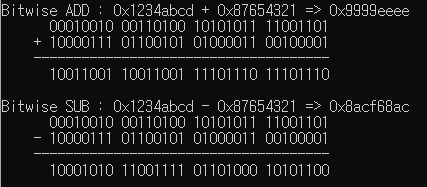
printf("\n");

}

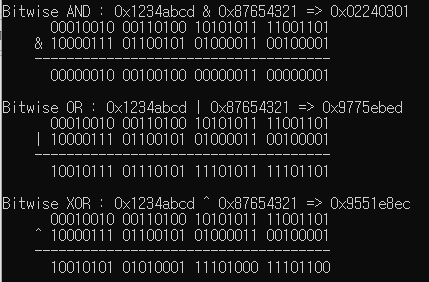
* 1. 0x00000 ~ 0xFFFF범위의 값을 가지는 16진수 2개를 입력 받고, 이들 16진수 데이터 값의 8진수, 10진수, 및 2진수 값을 출력하라.

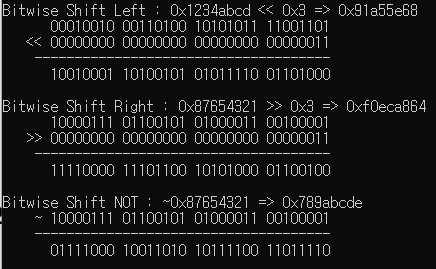


* 1. 입력된 2개의 16진수 값의 덧셈과 뺄셈을 계산하고, 계산 결과값의 16진수 및 2진수 값을 각각 출력하라.



* 1. 16진수 2개를 입력받아 bitwise AND, bitwise OR, bitwise XOR, shift\_left, shift\_right, bitwise NOT를 각각 계산하고, 계산 결과 값의 16진수 및 2진수 값을 각각 출력하라.





Oral Test

|  |
| --- |
| **(1) 정수 데이터를 10진수, 8진수, 16진수로 출력하기 위한 printf() 포맷인 %d, %o, %x에 대하여 설명하라. 8진수는 접두어 0으로 시작하도록 하고, 16진수는 접두어 0x로 시작하도록 하며, 빈자리에 0채우기를 하는 방법에 대하여 설명하라.**  %d : 부호가 있는 10진수 에 사용되는 포멧 지정자  %o : 부호가 없는 8진수 에 사용되는 포멧지정자  %x : 부호가 없는 16진수에 사용되는 포멧 지정자  % 뒤에 #을 입력하면 8진수는 접두어 0, 16진수는 접두어 0x로 시작한다.  출력공간을 정해주기 위해서 % 뒤에 수를 입력해주면 그 수만큼 공간을 가진 채로 출력한다.  소수점 이하 출력 자리 수를 정해주기 위해서 % 뒤에 .n(수)을 입력해주면 그 수만큼 소수점 아래 자리를 출력한다.  숫자 앞 빈 자리에 0을 채워주기 위해서 % 뒤에 0n(수)을 입력해주면 그 수만큼의 출력 공간을 생성하고 빈 자리에 0을 채워서 출력한다.  출력공간을 생성한 상태는 우측 정렬 되어있는 상태다. 수 앞에 -를 입력하면 왼쪽 정렬된다.  +를 붙이면 생략된 +부호가 함께 출력된다. |

|  |
| --- |
| **(2) 정수 데이터를 2진수 문자열로 출력하기 위한 알고리즘의 pseudo code를 작성하고, 이에 대하여 설명하라.**  Procedure printInt\_Bits(int d)  #define bit\_mask 0x01  #define num\_bits\_int 32   1. unsigned long l1, l2, msk; 2. int n = num\_bits\_int – 1; 3. while (--n >= 0) { 4. msk = bit\_mask << n; 5. l1 = d & msk; 6. l2 = l1 >>n; 7. printout l2; 8. if ((n % 8) == 0) { 9. printout “ “; 10. } 11. }   End of Procedure printInt\_Bits(int d)  마스킹 값을 n의 크기만큼 bit shift와 bit AND를 통해서 원하는 위치의 값이 1인지 0인지 판단한다.  n이 0이 될 때 까지 반복하고 매 반복마다 해당 수를 출력하고 8번 반복마다 공란을 채워준다. |

|  |
| --- |
| **(3) 2진수 데이터의 bit-wise AND, bit-wise OR, bit-wise XOR, shift left, shift right 계산을 예를 들어 설명하라. 음수의 shift right에서 sign bit 부분이 어떻게 처리되는가에 대하여 설명하라.**  bit-wise AND : 두 피연산자의 해당 비트가 모두 1이면 1, 아니면 0  bit-wise OR : 두 피연산자의 해당 비트 중 하나만 1이면 1, 아니면 0  bit-wise XOR : 두 피연산자의 해당 비트의 값이 같으면 0, 아니면 1  shift left : 지정된 개수만큼 모든 비트를 왼쪽으로 이동한다.  shift right : 지정된 개수만큼 모든 비트를 오른쪽으로 이동한다.      음수의 shift right에서 sign bit는 기존의 부호비트를 반복된다. |

|  |
| --- |
| **(4) 컴퓨터 시스템에서 2의 보수로 음의 정수 값을 표현하는 방법에 대하여 설명하고, 정수형 변수에 overflow가 발생하는 상황에 대하여 설명하라.**  해당 정수의 비트 표현에서 0을 1로, 1을 0으로 변환한다.  그 결과에 1을 더한다.    INT\_MAX는 자료형 int의 최대값을 나타낸다. 이 값에 +1을 하면 overflow로 인해 값이 최소값으로 넘어간다. 반대로 자료형 int의 최소값을 나타내는 INT\_MIN에 -1을 하면 최대값으로 넘어간다. |