

<기초문제>

1. 아래의 프로그램을 작성하시오. (/구현/ 부분을 채울 것, 표의 상단: 소스코드, 하단: 실행결과)
square 함수 : 입력의 제곱을 반환하는 함수

```
/*구현*/ //함수 선언

int main() {
    int x;
    x = 5;
    cout << /*구현*/ << endl; // square 함수 사용
}

//함수 정의
```



2. 아래의 프로그램을 작성하시오. (/구현/ 부분을 채울 것)

```
int get_num();
int myadd(int x, int y);
int mysub(int x, int y);
int mymul(int x, int y);
float mydiv(int x, int y);

int main() {
    int x, y;
    x = /*구현*/ //키보드로부터 값 입력
    y = /*구현*/
    cout << "x = " << x << ", y = " << y << endl;
    cout << "x + y = " << /*구현*/ << endl;
    cout << "x - y = " << /*구현*/ << endl;
    cout << "x * y = " << /*구현*/ << endl;
    cout << "x / y = " << /*구현*/ << endl;

    return 0;
}

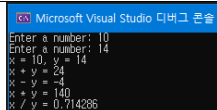
int get_num() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    return num;
}

int myadd(int x, int y) {
    return /*구현*/
}

int mysub(int x, int y) {
    return /*구현*/
}
```

```
int mymul(int x, int y) {
    return /*구현*/
}

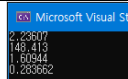
float mydiv(int x, int y) {
    return /*구현*/ //소수점 출력되도록
}
```



3. 아래의 프로그램을 작성하시오. (/구현/ 부분을 채울 것)

```
int main() {
    double value;
    value = 5;
    cout << /*구현*/ << endl; //제곱근
    cout << /*구현*/ << endl; //e의 지수 값 연산
    cout << /*구현*/ << endl; //로그연산(밑:자연상수e)
    cout << /*구현*/ << endl; //cosine

    return 0;
}
```

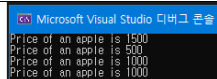


4. 아래의 프로그램을 작성하시오. (/구현/ 부분을 채울 것)

```
void applePrice(int a = 1000) {
    cout << "Price of an apple is " << a << endl;
}

// default augment 중요!
int main() {
    /*구현*/
    /*구현*/
    /*구현*/
    /*구현*/ // default arguments

    return 0;
}
```

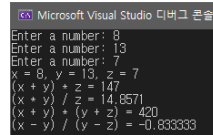


<응용문제>

1. 기초문제 2번 문제에서 구현한 4개의 함수 (myadd, mysub, mymul, mydiv) 만을 사용하여, 복합적인 연산을 수행하는 아래의 함수들을 구현하시오.

- addmul(int x, int y, int z) // return value: (x + y) * z
- muldiv(int x, int y, int z) // return value: (x * y) / z
- addmuladd(int x, int y, int z) // return value: (x + y) * (y + z)
- subdivsub(int x, int y, int z) // return value: (x - y) / (y - z)

1.출력화면:



2. 다음 조건을 만족하는 함수를 구현하시오.

- print_DOB 함수는 정수형 year, month, day 3개의 파라미터를 입력으로 받는다.
- print_DOB 함수는 default arguments를 설정하여 print_DOB 함수의 입력 값이 없을 경우 아래 화면과 같이 기본 출력 값이 나온다.
- 함수의 입력 값이 있으면 아래 화면과 같이 생년월일을 출력한다.

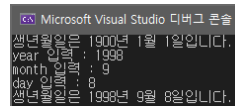
입출력 예시:

```
- 입력 값이 없을 경우
(output) 생년월일은 1900년 1월 1일입니다.

- 입력 값이 있을 경우
(input) 1998
      9
      8
(output) 생년월일은 1998년 9월 8일입니다.
```

※ 입력 값이 없을 경우 출력되는 default arguments를 확인하기 위해 print_DOB 함수 먼저 실행함.

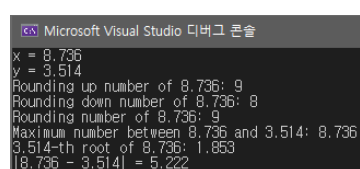
2.출력화면:



3. 두 개의 double 형 실수(x, y)를 키보드로 입력 받아 cmath 라이브러리를 이용하여 아래의 값을 계산 후 화면에 출력하시오. (참고: <http://www.cplusplus.com/reference/cmath/>)

- ceil(x) x를 올림한 값
- floor(x) x를 버림한 값
- round(x) x를 반올림한 값
- max(x,y) x와 y중에서 큰 값
- \sqrt{x} y제곱근 x ($x^{\frac{1}{2}}$ 과 같음)
- |x - y| x와 y의 차이 (절대값)

3.출력화면:



4. Call by value와 Call by reference를 사용하여 swap함수를 구현하시오. Call by value와 Call by reference의 차이를 보기 위해 아래의 출력화면과 같이 두 swap 함수 사용 전, 후의 x, y값을 출력하시오.

- Call by value를 사용한 swap 함수의 이름을 swap_call_by_value로 할 것.
- Call by reference를 사용한 swap 함수의 이름을 swap_call_by_reference로 할 것.
- 키보드로부터 정수 x, y를 입력 받는 get_data(int& x, int& y)를 구현할 것. main함수에서는 get_data함수를 통해 두 정수 x, y를 키보드로부터 입력 받음.

4.출력화면:

```
Microsoft Visual Studio 디버그 콘솔
x 입력 : 3
y 입력 : 5

swap_call_by_value 함수 사용 전
x = 3, y = 5
swap_call_by_value 함수 사용 후
x = 3, y = 5

swap_call_by_referece 함수 사용 전
x = 3, y = 5
swap_call_by_referece 함수 사용 후
x = 5, y = 3
```