

Object-Oriented Programming

Lab #04_2019100174안용상

Department: 화학공학과 Student ID: 2019101074 Name : 안용상

1. 아래 코드에서 1) 오류가 있으면, 오류를 설명하고, 2) 오류가 없다면 출력 결과를 설명하라.

(a) `std::cout << sqrt(4.5) << '\n';`

Sqrt는 double 형을 1개 받아서 제곱근을 구한뒤 double형으로 반환하는 함수이다. 따라서 이 경우 인자한개가 파라미터에 적절하게 위치하고있고, 4.5의 제곱근인 2.12132를 반환하고 std::cout함수가 실행되어 출력된다. \n또한 이어서 출력되어 줄바꿈이 실행된다



(b) `std::cout << sqrt(4.5, 3.1) << '\n';`

이 코드는 오류가 난다. 왜냐하면 sqrt라는 함수에는 파라미터가 2개가 아니라 1개만 존재하기 때문에, 하나의 인자만 받을 수 있기 때문이다.

(c) `std::cout << srand(55) << '\n';`

이 코드는 오류가 난다. 왜냐하면 srand라는 함수는 void형 함수이므로 아무 반환값도 내뱉지않기 때문에, cout함수로 출력할 무엇인가가 존재하지 않게된다.

(d) `std::cout << tolower('A') << '\n';`

'A'라는 character를 tolower라는 함수의 인자로써 전달하면 int tolower(int x)라는 함수는 문자를 int형으로 바꿔 이경우 97로 만든 뒤에, int형으로 반환한다.

따라서 이경우 tolower('A')자체는 97이 되고, 이것을 다시 문자로 바꾸지않고 출력했기에 출력또한 97이 된다. 이어서 줄바꿈이 실행된다

(e) `std::cout << exp() << '\n';`

이 코드는 오류가 난다. 왜냐하면 double exp(double x)라는 함수는 파라미터가 1개 이면서 default value가 없기때문에, 아무것도 인자로 넣어주지 않으면 함수에서 처리할 대상이 없어, 오류가 나게된다

(f) `std::cout << sqrt() << '\n';`

이 코드는 오류가 난다. 왜냐하면 `double sqrt(double x)`라는 함수는 파라미터가 1개 이면서 default value가 없기때문에, 아무것도 인자로 넣어주지 않으면 함수에서 처리할 대상이 없어, 오류가 나게된다

(g) `std::cout << static_cast<char>(toupper('E')) << '\n';`

'E'라는 character를 `toupper`라는 함수의 인자로써 전달하면 `int toupper(int x)`라는 함수는 문자를 int형으로 67을 얻은 뒤, 대문자인지 소문자인지 확인한다. 이때 이 경우 대문자 E기 때문에 이미 upper형태여서 upper에서 따로 추가적인 연산이 필요하지 않다. 따라서 `toupper('E')` 자체는 67이되고, 이것을 `static_cast<char>`로 character로 바꿔줬기에 67은 'E'로 다시 변환되어 출력된다. 그리고 줄바꿈이 실행된다

(h) `std::cout << static_cast<char>(toupper('e')) << '\n';`

'e'라는 character를 `toupper`라는 함수의 인자로써 전달하면 `int toupper(int x)`라는 함수는 문자를 int형으로 101을 얻은 뒤, 대문자인지 소문자인지 확인한다. 이때 이 경우 소문자 E기 때문에 알파벳 개수만큼 아스키코드값을 차감시킨다. 따라서 `toupper('e')` 자체는 67이되고, 이것을 `static_cast<char>`로 character로 바꿔줬기에 67은 'E'로 다시 변환되어 출력된다. 그리고 줄바꿈이 실행된다

(i) `std::cout << static_cast<char>(toupper("e")) << '\n';`

이 경우 오류가난다. 왜냐하면 `toupper`가 수행되려면 전달되는 문자가 char형태로 들어와 그 문자의 아스키코드로 형 변환이 되어야하지만, 이때는 리터럴인 "e"가 들어왔기 때문에 아스키코드로 형변환되지 못해 int형으로 받지못하고 이어서 코드가 실행되지 않는것이다

2. 아래 코드의 오류를 설명하라.

```
int proc(int x) {  
    return x + 2;  
}  
int proc(int n) {  
    return 2*n + 1;  
}  
int main() {  
    int x = proc(5);  
}
```

>> 오류설명

이 때 main함수위에 똑 같은 이름의 함수 `proc`이 정의되어 있다

이때 이 이름이 같은 이 두함수는, 함수 내의 로직과 결과값 반환연산이 다르지만, 똑같은 타입, 똑 같은 개수의 파라미터를 가진다.

이때 main함수에서 proc함수를 이용하기 위해, 인자로 5를 전달했지만, 이 main함수에서 이용할 수 있는 proc함수가 두개나 존재하기에 오류가 존재한다

3. 아래 코드의 오류를 설명하라.

```
#include <iostream>
int proc(int x) {
    return 2*x*x;
}
int main() {
    std::cout << proc(5, 4) << '\n';
}
```

>> 오류설명

위 코드는 에러가 난다. 왜냐하면 proc함수의 파라미터는 1개밖에 없는데, main함수에서 호출했을 때, proc에 두개의 인자를 전달하기 때문이다. 이때 파라미터 개수보다 많은 인자를 전달했기 때문에 에러가난다.

4. 아래 코드의 오류를 설명하라.

```
#include <iostream>
void proc(int x) {
    x = 2*x*x;
}
int main() {
    std::cout << proc(2) << '\n';
}
```

>> 오류설명

위 코드는 오류가 있다

Proc을 main위에 함수로 정의를 했다. 이때 void타입 함수로써, 계산만을 수행하고 반환값을 리턴하지 않는다.

그래서 main함수의 std::cout<< proc(2) 이부분에서 proc(2)가 아무 값도 반환하지 않으므로 std::cout은 출력할 값을 전달받지 못하게되고 ,오류가 생기게된다.

5. 아래 코드의 오류를 설명하라.

```
#include <iostream>
int proc(int x, y) {
    return x+y;
}
int main() {
    std::cout << proc(5, 4) << '\n';
}
```

>>오류설명

위의 코드는 에러가난다.

왜냐하면 proc함수의 y파라미터에 대한 타입지정을 해주지 않았기 때문이다.

Int proc(int x, y)라는 함수는 둘 다 타입지정이 된 것 처럼 보이지만, 사실은 int x만 int 형이라고 지정이 된것이고, y는 아무 타입도 지정받지 못한채 덩그러니 놓여있는 것이다. 이것은 인간의 문법으로는 x와 y모두 int형으로 변환되어도 이해가능한 문장이지만, 컴퓨터 C++문법에서는 어긋나므로 이는 명확하게 x만 타입지정이 되게 된다

따라서 이때 proc(5,4)라는 함수는 5만을 인자로 문제없이 전달받고, 4를 인자로 전달 받는 과정에서 컴파일러가 y의 타입을 알 수가 없기에 오류가 나게된다.

6. 아래 코드의 동작을 설명하라.


```
int main() {
    int num = 10;
    proc(num);
    std::cout << num << '\n';
}
```

>>동작설명

결론부터 말하면 이 코드는 10을 출력한다

proc함수를 거쳐 10이 제공되면서 2를 곱해졌지만, 그것은 proc함수 내부에서의 지역 변수 x일 뿐이므로 proc함수를 벗어나면서 이 x의 값은 무의미해진다. 다시 proc함수의 영역을 지나 main함수의 영역으로 나왔을 때, x는 proc함수에 전달될 때 값 10이 그대로 유지된 상태이고, 변하지 않았다. 따라서 이 10값은 std::cout << num의 num변수의 실제값으로 사용되며 10이라는 값이 출력이 된다.

이때 추가설명을 하자면 proc함수는 void형 함수로 아무것도 반환하지 않는다. 그리고 아무것도 출력하지도 않고, reference variable 또한 받지 않아서 함수밖의 어떤 변수의 실제 값도 바꿔주지 않는다. 그저 $x = 2 * x * x$ 라는 연산을 한뒤에 사라질 뿐인 함수이다.

 Microsoft Visual Studio 디버그 콘솔

```
10
C:\Users\qhfkd\Project1\Debug\Project1.exe(프로세스 30084개)이(가)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...■
```

7. 아래 코드의 동작을 설명하라.

```
#include <iostream>
int proc(int x) {
    return 2*x*x;
}
int main() {
    int x = 10;
    std::cout << proc(x) << '\n';
}
```

>>코드설명

결론부터 말하자면, 위 코드는 200을 출력한다.

Int형 x를 받아 $x = 2 * x * x$ 를 수행해서 x에 2를 곱하고 자기자신을 곱하는 연산을 해주고 다시 자기자신에 그 연산값을 할당하는 연산을 수행한 뒤에 그것을 int형 값으로 return 한다. 이때 proc함수를 return값을 내보내면서 빠져나오면, 이 return값은 std::cout 함수에 의해 출력되게된다. 그래서 200이 출력되고 이어서 줄바꿈이 실행된다.

200

C:\Users\qhfkd\Project1\Debug\Project1.exe(프로세스 18008개)이(가) 종료되었습니다. 디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [하도록 설정합니다]. 이 창을 닫으려면 아무 키나 누르세요....

8. 실수를 반복문으로 입력 받아서 (입력 종료는 EOF), 산술 평균(average), 기하 평균(geometric mean)과 조화평균(harmonic mean)을 계산해서 출력하는 프로그램을 작성하라.

```
#include <iostream>
double input_double(int& check, float& count) {
    // for 반복문으로 실행할, 실수를 받아주는 함수 작성
    //여기서 reference 들을 받았는데, input_double이라는 함수의 스코프 밖의 변수를
    // input_double자체에서 조작할 수 있게만들기 위함이다
    // check라는 인수는, main함수에서 input_double함수를 호출하기 전에
    // 선언과 초기화가 되어있고
    // 사용자의 EOF신호를 받기전까지 1로 유지된다.
    //만약 사용자의 EOF신호를 받는다면 check가 0으로 바뀌어 더이상 for문을
    //지속하지못하게 해서 input_double함수를 호출하지 못하게 한다.
    // 또한 count라는 reference인자는 실제로 main함수에서 선언과 초기화를 해준
    // count 변수가 들어갈것인데,
    // 이 count라는 main함수의 변수는 총 몇번 실수를 받았는지를 세어주는 역할을 한다.
    // cin이 정상적으로 받아졌을때, input_double함수를 빠져나오기 직전 +1이 되어 저장된다.
    double number; //실수를 담을 number라는 더블형 변수를 선언한다.
    std::cout << "실수를 입력하세요"; // 사용자에게 지시하는 출력문을 작성한다
    std::cin >> number; // 사용자에게 입력을 받아서 number변수에 넣는다
    if (std::cin.eof() == 1) {
        // 만약 cin이 eof일때, check값을 0 으로 바꾸며 0을 반환함과 동시에 함수를 빠져나간다
        check = 0;
        return 0;
    }
    else if (std::cin.fail() == 1) { // 만약 실수가 아니라 다른 입력을 받았을 시
        //아래의 코드가 실행된다
        while (std::cin.fail() == 1) { // 실수가 아니라 다른입력을 받았다면,

            //아래와 같은 코드를 무한 반복한다. 언제까지? 실수를 받을때까지.
            std::cin.clear(); // 입력 오류플래그를 초기화한다.
            std::cin.ignore(1000, '\n'); // 입력버퍼에 남아있는 문자들을 \n이 나올때까지
            //최대 1000개까지 지운다.
            std::cout << "반드시 실수를 입력하셔야합니다! : 다시입력 \n ";
            // 다시 입력하라는 지시문을 출력
            std::cin >> number; // number에 입력을 받아 할당한다.
            if (std::cin.eof() == 1) {
```

```

// 만약 cin이 eof일때, check값을 0 으로 바꾸며 0을 반환함과 동시에 함수를 빠져나간다
    check = 0;
    return 0;
}
}
}
count += 1; // eof가 아닌경우에 함수를 빠져나갈때 count reference값을 +1해준다.
return number; // 받은 실수를 반환하며 빠져나간다.
}
int main() {
    int check = 1;
    // input_double 안에 들어가서 eof를 판별하게 될 변수 check를 선언하고 초기화한다.
    double sum = 0;
    // 산술 평균을 계산할 변수를 0으로 초기화한다
    float count = 0;
    // input_double 안에 들어가서 몇번 실수를 입력받았는지 계산할 변수
    // count를 선언하고 초기화한다.
    double number = 0;
    // input_double함수를 나오며 반환받은 값을 저장할 변수 number를
    //0으로 초기화한다.
    double dot = 1;
    // 기하평균을 연산하기 위해 필요한 dot변수를 선언 및 1로 초기화한다.
    double inverse_sum = 0;
    // 조화평균을 연산하기 위해 필요한 inverse_sum변수를 선언 및 1로 초기화한다.

    for (; check == 1;) {
        // check가 1이 아닌 경우가 나올때까지 반복한다.
        number = input_double(check, count);
        // input_double함수를 호출해서, reference파라미터에 check와 count를 인자로 넣어준다
        //그리고 그 반환값을 number에 담는다
        if (check == 1) {
            // 만약 input_double함수를 빠져나왔는데도 여전히 check가 1인 상황이라면
            //아래의 코드를 수행한다.
            sum += number;
            // 산술평균을 위한 변수 sum에 사용자에게 입력받아 반환받은 실수 number를 더한다
            dot *= number;
            // 기하평균을 위한 변수 dot에 사용자에게 입력받아 반환받은 실수 number를 곱한다
            inverse_sum += 1 / number;
            // 조화평균을 위한 변수 inverse_sum에 사용자에게 입력받아 반환받은 실수 number를
            // 역수를 취해 더해준다
        }
    }
    // check 가 1이 아닌채로 input_double함수를 나왔다면 for 문은 멈추고
    //아래와 같은 코드를 실행한다
    std::cout << "산술평균 : " << sum / count << std::endl;
    // 계산해준 산술평균을 출력한다
    std::cout << "기하평균 : " << std::pow(dot, 1.0 / count) << std::endl;
    // 계산해준 dot변수에 받은 실수의 개수만큼 1/n제공해준다. 즉 기하평균을 구해준다.
    // 그리고 출력한다
    std::cout << "조화평균 : " << count / inverse_sum << std::endl;
    // 계산해준 inverse_sum을 출력한다. 그리고 최종적으로 조화평균을 구하기위해
    //inverse_sum의 역수에 받은실수의 개수만큼 곱해준다.
    // 즉 조화평균을 구한다. 그리고 출력한다.
}

```

```
실수를 입력하세요d
반드시 실수를 입력하셔야합니다! : 다시입력
?
반드시 실수를 입력하셔야합니다! : 다시입력
1
실수를 입력하세요2
실수를 입력하세요3
실수를 입력하세요40
실수를 입력하세요531
실수를 입력하세요234
실수를 입력하세요^Z
산술평균 : 135.167
기하평균 : 17.6098
조화평균 : 3.21804

C:\Users\Wadmin\source\repos\cpp\cpp\Debug\cpp.exe( 프로세스 3280개)이(가) 종료되었습니다(코
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...]
```

9. 아래와 같은 순서를 가지는 피보나치 수열(Fibonacci series)을 출력하는 프로그램을 사용자 정의 함수(int Fibonacci(int n))를 이용하여 작성하라. (Recursive function는 사용 금지)



0, 1, 1, 2, 3, 5, ..., 34

<코드>

```
#include <iostream>
void prompt(int number, int last);
// 출력을 담당하는 사용자 함수 프로토타입 상단에 기재
int Fibonacci(int n);
// 피보나치 연산을 담당하는 사용자 함수 프로토타입 상단에 기재

int main() {
    int n;
    // 사용자로부터 받은 피보나치 수열 길이를 저장하는 변수 n
    std::cout << "몇 번째 피보나치 수열까지 보시겠습니까? ";
    // 사용자에게 입력을 요청하는 출력문
    std::cin >> n; // 사용자로부터 입력 받기

    if (std::cin.fail() == 1 || (n < 0)) {
        // 만약 cin이 실패하면 (즉, 0보다 작은정수거나
        //정수가 아닌 값이 입력되면)
        while ((std::cin.fail() == 1) || (n < 0)) {
            // 올바른 정수가 입력될 때까지 반복
            std::cout << "다시 입력하세요: ";
            // 사용자에게 다시 입력하라는 메시지 출력
```



```

        std::cin.clear(); // 입력 버퍼 초기화
        std::cin.ignore(1000, '\n'); // 입력 버퍼 비우기
        std::cin >> n; // 새로운 입력 받기
    }
}

int last = 0;
// 마지막 순회인지를 판별하기 위한 변수 초기화
for (int i = 1; i <= n; ++i) {
    // 1부터 n까지 반복문 실행
    if (i == n) // 마지막 순회인 경우
        last = 1; // last 변수에 1 할당
    prompt(Fibonacci(i), last);
    // 피보나치 수열의 i번째 값을 prompt 함수에 넣어 출력
}
}

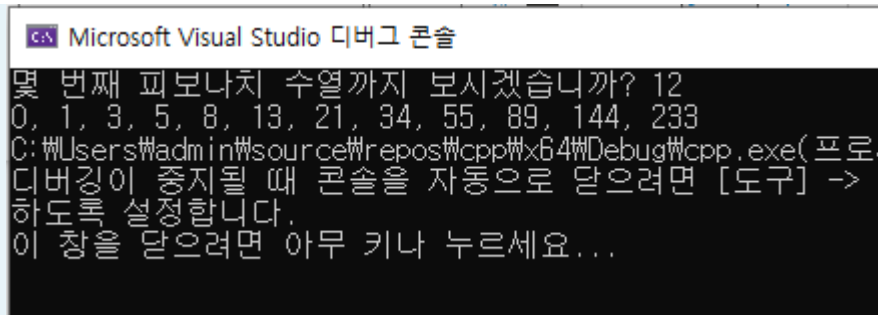
void prompt(int number, int last) {
    std::cout << number; // 숫자 출력
    if (last != 1) { // 마지막 순회가 아닌 경우
        std::cout << ", "; // 쉼표와 공백 출력
    }
}

int Fibonacci(int n) {
    int number_1 = 0; // 첫 번째 수 초기화
    int number_2 = 1; // 두 번째 수 초기화
    int garbage; // 임시 변수 초기화
    for (int i = 1; i <= n; ++i) { // 1부터 n까지 반복문 실행
        if (n == 1) { // n이 1인 경우
            return number_1; // 첫 번째 수 반환
        }
        else if (n == 2) { // n이 2인 경우
            return number_2; // 두 번째 수 반환
        }
        else if (i <= n) { // i가 n보다 작거나 같은 경우
            garbage = number_1;
            // garbage 변수에 number_1 값 저장
            number_1 = number_2;
            // number_1에 number_2 값 저장
            number_2 += garbage;
            // number_2에 자기 자신과 임시변수 garbage 더하기
        }
    }

    return number_2;
    // n번째 피보나치 수열 반환
}

// Fibonacci 함수는 매개변수 n이 양의 정수인 경우에만 사용 가능.
// prompt 함수는 출력을 담당하는 함수로, 인자로 출력할 숫자와
// //마지막 순회인지 아닌지를 받음
// 사용자가 입력한 값이 올바른지 확인하기 위해 cin.fail() 함수를 사용
// 만약 cin.fail() 값이 1이라면, 즉 사용자가 정수가 아닌 값을 입력한 경우이므로
// "다시 입력!" 메시지를 출력하고, 입력 버퍼를 초기화한 뒤, cin을 다시 받음.

```



10. 두 개의 정수를 매개변수(parameter, n, r)로 받아서 순열(permutation, nPr)을 계산하여 반환하는 함수를 작성하라.

<코드>

```
#include <iostream>
//순열의 계산은
// n 과 r을 받았을 때
// n부터 거꾸로 n-r+1까지 곱해주는거랑 같다.
// ex) n =10 r =3이라면
//10부터 (10-3+1)인 8까지
// 10 * 9 * 8을 해주는것과 같다
//그 원리로 아래 순열을 구현해보겠다

// n과 r을 매개변수로 받아서 nPr 값을 계산하고 반환하는 함수
double get_permutation(int n, int r) {
    int iters = r; // iters 변수에 r 값을 저장
    int dot = 1; // dot 변수를 1로 초기화
    for (int i = 1; i <= iters; i++) { // i가 1부터 iters(즉, r)까지 반복문 실행
        dot *= (n - i + 1); // dot에 (n-i+1)값을 곱해줌
    }
    std::cout << "순열의 값은 " << dot << "입니다"; // 최종 순열 값을 출력
    return dot; // dot 값을 반환
}

int main() {
    int n; int r; // n과 r을 담을 변수 선언
    std::cout << "순열을 계산하겠습니다. 매개변수 n과 r을 입력해주세요!";
    // 사용자에게 입력을 요구하는 출력
    std::cin >> n >> r;
    // 사용자로부터 n과 r을 입력받음
    get_permutation(n, r);
    // get_permutation 함수를 호출
}
```

- get_permutation 함수: 매개변수로 받은 n과 r 값으로 nPr 값을 계산하고 반환하는 함수
- iters 변수: 반복문에서 사용할 변수로, r 값을 저장
- dot 변수: nPr 값을 저장하기 위한 변수로, 1로 초기화
- for 반복문: r번 반복하며 dot 값을 업데이트
- std::cout << dot << std::endl;: 현재 dot 값을 출력
- "순열의 값은 " << dot << "입니다": 최종 dot 값을 출력하는 문자열
- main 함수: 사용자로부터 n과 r 값을 입력받고 get_permutation 함수를 호출

11. 이차 방정식(quadratic equation $ax^2+bx+c=0$)의 실근을 계산하여 출력하는 프로그램을 아래와 같이 처리되도록 작성하라.

(a) a,b,c (type double)를 입력 받고

(b) 다음과 같은 조건으로 계산

- $a=0$ 이고 $b=0$ 이면, 입력 오류(not a valid equation)
- $a=0$ 이고 $b \neq 0$ 이면, 1차로 근은 $x=-c/b$
- $b^2 - 4ac < 0$ 이면, 실근 없음(not a real solution)

• $b^2 - 4ac \geq 0$ 이면, 실
근

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

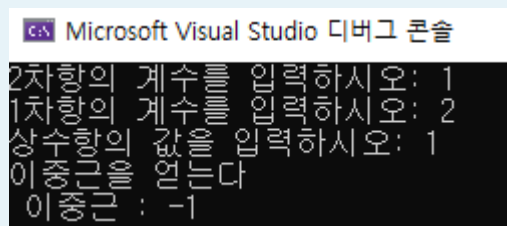
```

#include <iostream>
#include <cmath>

// 이차방정식 해를 구하는 함수
void print_solve(double a, double b, double c) {
    // a와 b가 모두 0이면 입력 오류로 판단하여 "입력오류" 출력
    if (a == 0 && b == 0) {
        std::cout << "입력오류";
    }
    // a가 0이면 일차방정식으로 판단하여 -c/b를 출력
    else if (a == 0 && b != 0) {
        std::cout << -c / b;
    }
    // 판별식이 음수이면 실근이 없으므로 "실근없음" 출력
    else if ((b * b - 4 * a * c) < 0) {
        std::cout << "실근없음";
    }
    // 판별식이 0 이상이면 두 개의 해를 가지므로 근의 공식을 이용하여 두 해를 출력
    else if ((b * b - 4 * a * c) >= 0) {
        double pre = -b / (2 * a); // 근의공식 해에서 -b/2a를 pre라는 변수에
        double post = pow((b * b - 4 * a * c) / (2 * a), 1.0 / 2.0);
        //(루트 b제곱 마이너스 4ac)/2a 를 post변수에 담는다.
        if (post == 0) { //post가 0인경우 이중근이므로 이중근이라고 표시한다.
            std::cout << "이중근을 얻는다 \n 이중근 : " << pre;
        }else
            // post가 0이아닌 값일때 서로다른 두 근을 가진다.
            std::cout << "두 근은 : " << pre + post << " or " << pre - post;
        //그리고 플러스마이너스를 구현해서 근 두개를 얻고 출력한다.
    }
}

int main() {
    double a, b, c;
    std::cout << "2차항의 계수를 입력하십시오: ";
    //2차항의 계수받기
    std::cin >> a;
    std::cout << "1차항의 계수를 입력하십시오: ";
    //1차항의 계수받기
    std::cin >> b;
    std::cout << "상수항의 값을 입력하십시오: ";
    //상수항의 계수받기
    std::cin >> c;
    //함수 호출
    print_solve(a, b, c);
}

```



Microsoft Visual Studio 디버그 콘솔

```
2차항의 계수를 입력하시오: 1
1차항의 계수를 입력하시오: 3
상수항의 값을 입력하시오: 1
두 근은 : 0.0811388 or -3.08114
C:\Users\admin\source\repos\cpp\64\Debug\cpp.exe(프로세스 2048)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션]
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...■
```

Microsoft Visual Studio 디버그 콘솔

```
2차항의 계수를 입력하시오: 0
1차항의 계수를 입력하시오: 1
상수항의 값을 입력하시오: -13
13
C:\Users\admin\source\repos\cpp\64\Debug\cpp.exe(프로세스 2048)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션]
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...■
```

Microsoft Visual Studio 디버그 콘솔

```
2차항의 계수를 입력하시오: 0
1차항의 계수를 입력하시오: 0
상수항의 값을 입력하시오: 31
입력오류
C:\Users\admin\source\repos\cpp\64\Debug\cpp.exe(프로세스 2048)
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션]
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...■
```