



# Lab 10

2023학년도 2학기 프로그래밍개론

01, 02, 03분반

숙명여자대학교 소프트웨어학부

데이터 지능 연구실

TA 유사라

4ra@sookmyung.ac.kr

---

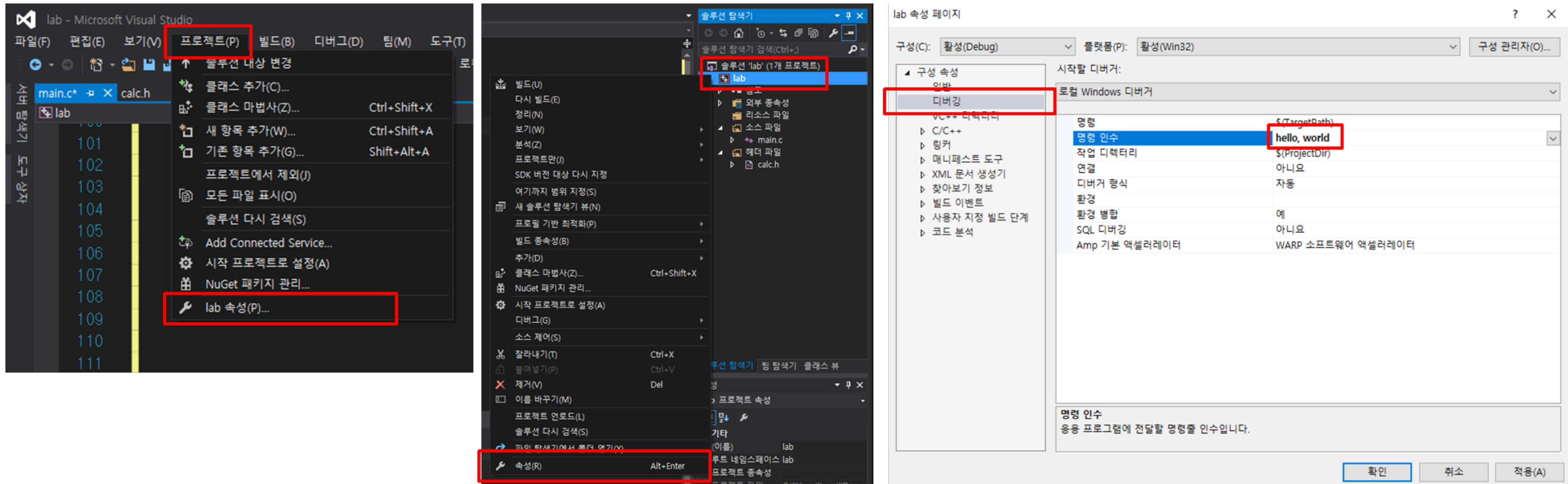
- 과제 1: 명령 행 인수와 함수에 대한 포인터를 사용하여 적분함수 `integ()`를 작성해보자.

- 명령 행 인수(command line arguments list)란?
  - 프로그램에 인수 값을 인가하는 것. = 프로그램 시작 시, 인수를 전달 해주는 것.
  - 명령 행 인수를 처리하기 위해서, 프로그램에서는 main()함수에 매개변수를 정의해야한다.
- main(int argc, char \*argv[])
  - argc (=argument count) : 인수의 개수
  - argv (=argument vector) : 인수값을 문자열로 저장하는 배열
- 명령 행 인수 전달 방법
  - Visual studio안에서 프로그램 및 코드를 그냥 실행하면 명령 행 인수를 제공할 수 없음.
  - 명령 행 인수를 따로 전달 해주어야함 !

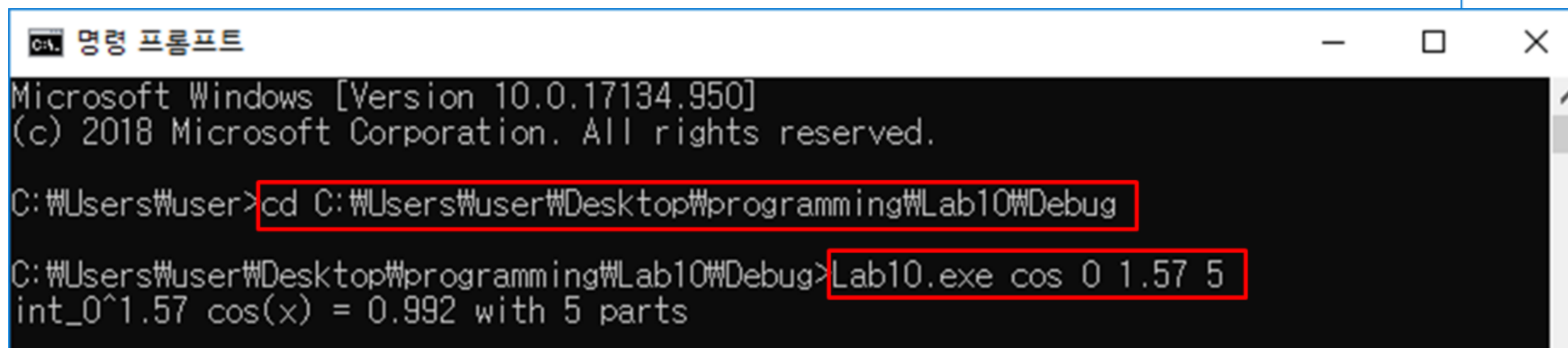
- visual studio 자체에서 명령 행 인수 전달하는 방법

1) 프로젝트 메뉴 > lab 속성 > 구성속성 > 디버깅 > 명령 인수 칸에 직접입력

2) 솔루션 탐색기 창 프로젝트 마우스 우클릭 > 속성 > 구성속성 > 디버깅 > 명령 인수 칸에 직접입력



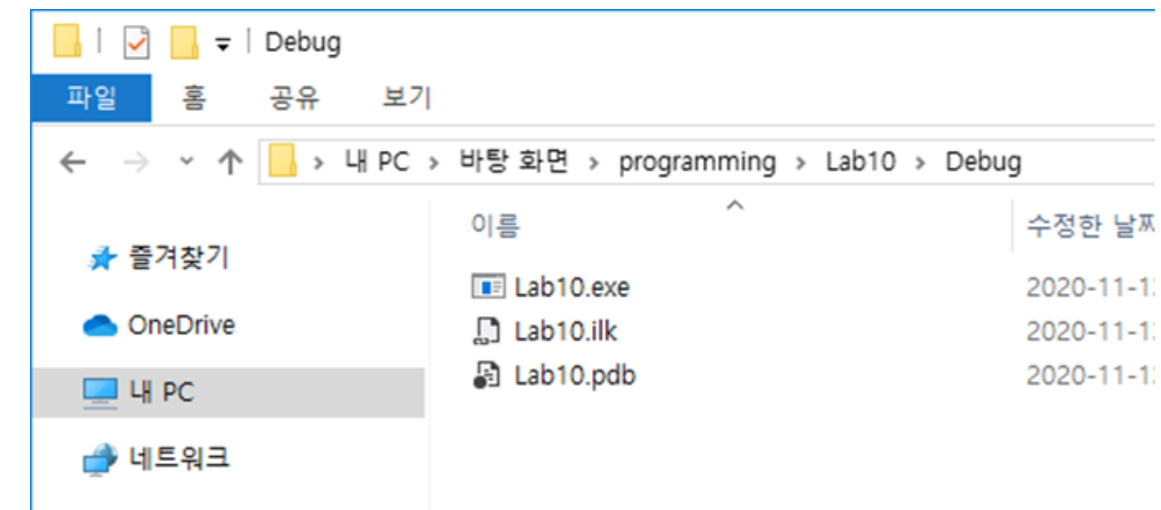
- 명령프롬프트(cmd)로 프로그램을 실행시켜 명령 행 인수 전달하는 방법
  - 시작 메뉴 > windows 시스템 > 명령프롬프트(cmd) > 프로젝트 폴더의 Debug폴더위치로 이동
  - 명령어 cd (=change directory) : 현재위치를 변경해라.
    - >cd 이동하고싶은경로 ex ) cd C:\Users\user\Desktop\programming\Lab10\Debug
  - Debug 폴더 안, 프로젝트 이름의 exe실행파일 존재.
  - 실행
    - >프로젝트이름.exe 인수1 인수2 인수3 인수4



```
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>cd C:\Users\User\Desktop\programming\Lab10\Debug

C:\Users\User\Desktop\programming\Lab10\Debug>Lab10.exe cos 0 1.57 5
int_0^1.57 cos(x) = 0.992 with 5 parts
```



- 구분구적법 (Integration by parts)

: 적분구간을 n등분하고, 내접/외접 직사각형이나 내접 사다리꼴의 면적의 합으로 적분 값의 근사값을 구하는 방법  
(아래 식에서는 사다리꼴을 이용하고 있음)

$$\int_a^b f(x) \approx \sum_{i=1}^n \frac{f(a + (i-1)h) + f(a + ih)}{2} h \quad (\text{단, } h = \frac{b-a}{n})$$

- 먼저  $y = x*x + x$ ,  $y = \sin(x)$  등에 대한 개별 구분구적함수 `integ1()`, `integ2()`를 작성해본다.  
임의의 함수에 대한 구분구적함수 `integ()`를 작성한다.

```
#include <stdio.h>
#include <math.h>
#define PI 3.142592
main() {
    double integ1(double a, double b, int n); /* integrate  $y = x^2 + x$  */
    double integ2(double a, double b, int n); /* integrate  $y = \sin(x)$  */
    printf("int_0^1 x*x + x = %.3g\n", 1.0 / 3 + 1.0 / 2);
    printf("int_0^1 x*x + x = %.3g with 5 parts\n", integ1(0, 1, 5));
    printf("int_0^1 x*x + x = %.3g with 10 parts\n", integ1(0, 1, 10));
    printf("int_1^2 x*x + x = %.3g with 100 parts\n", integ1(0, 1, 100));
    printf("\n");
    printf("int_0^PI/2 sin(x) = %.3g\n", 1.0);
    printf("int_0^PI/2 sin(x) = %.3g with 5 parts\n", integ2(0, PI/2, 5));
    printf("int_0^PI/2 sin(x) = %.3g with 10 parts\n", integ2(0, PI/2, 10));
    printf("int_0^PI/2 sin(x) = %.3g with 100 parts\n", integ2(0, PI/2, 100));
}
```

```
double integ1(double a, double b, int n) {  
    double sum, x, x1, delta, area;  
    int i;  
    delta = (b - a) / n;  
    for (i = 0, sum = 0.0; i < n; i++) {  
        x = a + i * delta;  
        x1 = x + delta;  
        area = delta * ((x * x + x) + (x1 * x1 + x1)) / 2;  
        sum += area;  
    }  
    return sum;  
}
```

```
double integ2(double a, double b, int n) {  
    double sum, x, x1, delta, area;  
    int i;  
    delta = (b - a) / n;  
    for (i = 0, sum = 0.0; i < n; i++) {  
        x = a + i * delta;  
        x1 = x + delta;  
        area = delta * (sin(x) + sin(x1)) / 2;  
        sum += area;  
    }  
    return sum;  
}
```



$\int_0^1 x \cdot x + x = 0.833$

$\int_0^1 x \cdot x + x = 0.84$  with 5 parts

$\int_0^1 x \cdot x + x = 0.835$  with 10 parts

$\int_1^2 x \cdot x + x = 0.833$  with 100 parts

$\int_0^{\pi/2} \sin(x) = 1$

$\int_0^{\pi/2} \sin(x) = 0.992$  with 5 parts

$\int_0^{\pi/2} \sin(x) = 0.998$  with 10 parts

$\int_0^{\pi/2} \sin(x) = 1$  with 100 parts

```
#include <stdio.h>
#include <math.h>
#define PI 3.141592
main() {
    double integ(double a, double b, int n, double (*f)(double)); /* integrate y = f(x) */
    double f1(double x); /* f1(x) = x * x + x; */
    printf("int_0^1 x*x + x = %.3g\n", 1.0 / 3 + 1.0 / 2);
    printf("int_0^1 x*x + x = %.3g with 5 parts\n", integ(0, 1, 5, f1));
    printf("int_0^1 x*x + x = %.3g with 10 parts\n", integ(0, 1, 10, f1));
    printf("int_1^2 x*x + x = %.3g with 100 parts\n", integ(0, 1, 100, f1));
    printf("\n");
    printf("int_0^PI/2 sin(x) = %.3g\n", 1.0);
    printf("int_0^PI/2 sin(x) = %.3g with 5 parts\n", integ(0, PI/2, 5, sin));
    printf("int_0^PI/2 sin(x) = %.3g with 10 parts\n", integ(0, PI/2, 10, sin));
    printf("int_0^PI/2 sin(x) = %.3g with 100 parts\n", integ(0, PI/2, 100, sin));
}
```

```
double integ(double a, double b, int n,  
             double (*f)(double))  
{  
    double sum, x, x1, delta, area;  
    int i;  
    delta = (b - a) / n;  
    for (i = 0, sum = 0.0; i < n; i++) {  
        x = a + i * delta;  
        x1 = x + delta;  
        area = delta * (f(x) + f(x1)) / 2;  
        sum += area;  
    }  
    return sum;  
}  
  
double f1(double x) {  
    return x * x + x;  
}
```

매개변수로 double 값을 취하고 double 값을 리턴하는  
함수에 대한 포인터 f

$\int_0^1 x \cdot x + x = 0.833$

$\int_0^1 x \cdot x + x = 0.84$  with 5 parts

$\int_0^1 x \cdot x + x = 0.835$  with 10 parts

$\int_1^2 x \cdot x + x = 0.833$  with 100 parts

$\int_0^{\pi/2} \sin(x) = 1$

$\int_0^{\pi/2} \sin(x) = 0.992$  with 5 parts

$\int_0^{\pi/2} \sin(x) = 0.998$  with 10 parts

$\int_0^{\pi/2} \sin(x) = 1$  with 100 parts

# 과제 1: 명령 행 인수와 함수에 대한 포인터 (integ.c)

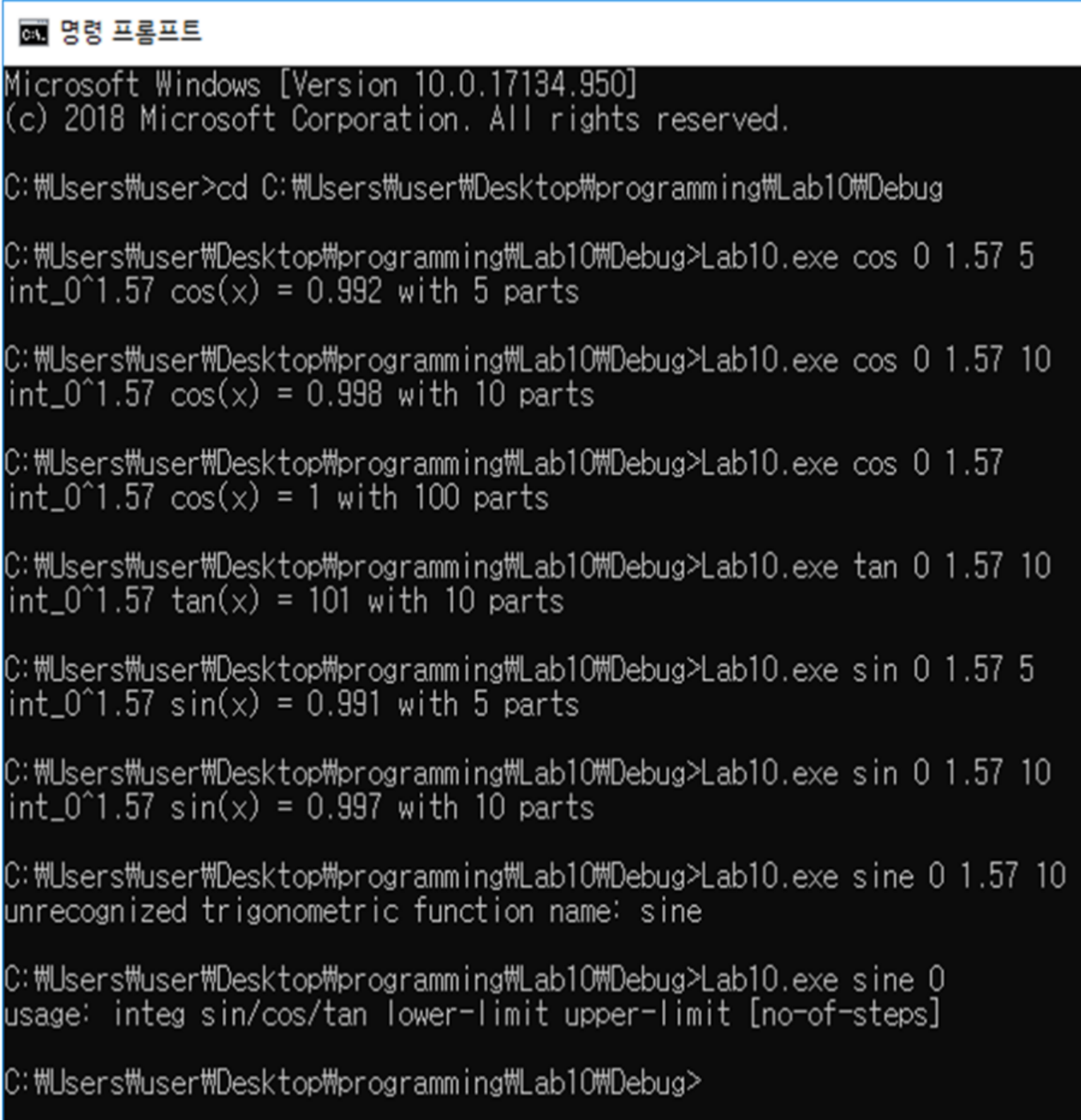
- 아래 지시 사항에 따라 프로그램 integ를 작성하라.
  - 적분 대상 삼각함수의 이름(sin, cos, tan 중의 하나), 적분 구간을 나타내는 2개의 실수, 분할 구간의 수 등을 명령 행 인수로 취한다.
  - 분할 구간의 수가 주어지지 않을 경우에는 100으로 가정한다.
  - 삼각함수의 이름이 sin, cos, tan 중의 하나가 아니거나 인수의 개수가 부족하면 적절한 오류 메시지를 출력한다.
  - 인수들이 적절하게 주어질 경우 강의자료에 주어져 있는 integ() 함수를 사용하여 적분 값을 구하여 적절히 출력한다.  
(integ() 함수를 변경하여서는 안 된다.)
  - 모든 출력은 다음 페이지 실행 예제의 출력 내용을 따라야 한다.

# 과제 1: 명령 행 인수와 함수에 대한 포인터 (integ.c)

프로그래밍개론

- 실행 화면 및 보고서 캡처화면 예시

```
C:\CProjects\integ\Debug>integ cos 0 1.57 5
int_0^1.57 cos(x) = 0.992 with 5 parts
C:\CProjects\integ\Debug>integ cos 0 1.57 10
int_0^1.57 cos(x) = 0.998 with 10 parts
C:\CProjects\integ\Debug>integ cos 0 1.57
int_0^1.57 cos(x) = 1 with 100 parts
C:\CProjects\integ\Debug>integ tan 0 1.57 10
int_0^1.57 tan(x) = 101 with 10 parts
C:\CProjects\integ\Debug>integ sin 0 1.57 5
int_0^1.57 sin(x) = 0.991 with 5 parts
C:\CProjects\integ\Debug>integ sin 0 1.57 10
int_0^1.57 sin(x) = 0.997 with 10 parts
C:\CProjects\integ\Debug>integ sine 0 1.57 10
unrecognized trigonometric function name: sine
C:\CProjects\integ\Debug>integ sine 0
usage: integ sin/cos/tan lower-limit upper-limit [no-of-steps]
```



```
Microsoft Windows [Version 10.0.17134.950]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>cd C:\Users\User\Desktop\programming\Lab10\Debug

C:\Users\User\Desktop\programming\Lab10\Debug>Lab10.exe cos 0 1.57 5
int_0^1.57 cos(x) = 0.992 with 5 parts

C:\Users\User\Desktop\programming\Lab10\Debug>Lab10.exe cos 0 1.57 10
int_0^1.57 cos(x) = 0.998 with 10 parts

C:\Users\User\Desktop\programming\Lab10\Debug>Lab10.exe cos 0 1.57
int_0^1.57 cos(x) = 1 with 100 parts

C:\Users\User\Desktop\programming\Lab10\Debug>Lab10.exe tan 0 1.57 10
int_0^1.57 tan(x) = 101 with 10 parts

C:\Users\User\Desktop\programming\Lab10\Debug>Lab10.exe sin 0 1.57 5
int_0^1.57 sin(x) = 0.991 with 5 parts

C:\Users\User\Desktop\programming\Lab10\Debug>Lab10.exe sin 0 1.57 10
int_0^1.57 sin(x) = 0.997 with 10 parts

C:\Users\User\Desktop\programming\Lab10\Debug>Lab10.exe sine 0 1.57 10
unrecognized trigonometric function name: sine

C:\Users\User\Desktop\programming\Lab10\Debug>Lab10.exe sine 0
usage: integ sin/cos/tan lower-limit upper-limit [no-of-steps]

C:\Users\User\Desktop\programming\Lab10\Debug>
```

# 과제 1: 명령 행 인수와 함수에 대한 포인터 (integ.c)

프로그래밍개론

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
int main(int argc, char *argv[])
{
    double integ(double a, double b, int n, double (*f)(double));
    double *f(double);
    double lower, upper, val;
    int nsteps;
    if(/* 명령행인수개수 확인하여 integ함수의 사용법에 맞는지 체크 */){
        ...
    }
    nsteps = /* 분할 구간의 수가 주어지지 않을 경우에는 100으로 가정 */
    lower = /* 명령행인수2 */
    upper = /* 명령행인수3 */
```

```
    if (/* sin함수일경우 */)
        val = integ(lower, upper, nsteps, sin);
    else if (/* cos함수일경우 */)
        val = integ(lower, upper, nsteps, cos);
    else if (/* tan함수일경우 */)
        val = integ(lower, upper, nsteps, tan);
    else { /*integ함수의 사용법에 맞지 않을 경우 오류메세지 출력*/
        printf("unrecognized trigonometric function name: %s\n",
               argv[1]);
        return 1;
    }
    printf("int_%g^%g %s(x) = %.3g with %d parts\n", lower, upper,
           argv[1], val, nsteps);
    return 0;
}
```

# 과제 1: 명령 행 인수와 함수에 대한 포인터 (integ.c)

프로그래밍개론

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
int main(int argc, char *argv[])
{
    double integ(double a, double b, int n, double (*f)(double));
    double *f(double);
    double lower, upper, val;
    int nsteps;
    if(/* 명령행인수개수 확인하여 integ함수의 사용법에 맞는지 체크 */){
        ...
    }
    nsteps = /* 분할 구간의 수가 주어지지 않을 경우에는 100으로 가정 */
    lower = /* 명령행인수2 */
    upper = /* 명령행인수3 */
```

```
    if (/* sin함수일경우 */)
        val = integ(lower, upper, nsteps, sin);
    else if (/* cos함수일경우 */)
        val = integ(lower, upper, nsteps, cos);
    else if (/* tan함수일경우 */)
        val = integ(lower, upper, nsteps, tan);
    else { /*integ함수의 사용법에 맞지 않을 경우 오류메세지 출력*/
        printf("unrecognized trigonometric function name: %s\n",
               argv[1]);
        return 1;
    }
    printf("int_%g^%g %s(x) = %.3g with %d parts\n", lower, upper,
           argv[1], val, nsteps);
    return 0;
}
```



## 과제 1: 명령 행 인수와 함수에 대한 포인터 (integ.c)

프로그래밍개론

```
double integ(double a, double b, int n, double (*f)(double))
{
    double sum, x, x1, delta, area;
    int i;

    delta = (b - a) / n;
    for (i = 0, sum = 0.0; i < n; i++) {
        x = a + i * delta;
        x1 = x + delta;
        area = delta * (f(x) + f(x1)) / 2;
        sum += area;
    }
    return sum;
}
```

- 과제 제출 기한
  - 01, 02분반 : 11월 28일 (화) PM 11:59 까지
  - 03분반 : 11월 29일 (수) PM 11:59 까지
- 제출 장소
  - 스노우보드 과제 제출 페이지에 업로드
- 제출 기한 이후 24시간 이내 지각 제출 2점 감점
- 추가 제출 받지 않음

- 소스파일(.c)과 과제보고서(.docx)가 담긴 압축파일(.zip) 제출
    - 압축 파일 이름: Lab10\_학번\_이름.zip
    - 'Lab10\_학번\_이름'으로 된 c파일 + 'Lab10\_학번\_이름'으로 된 .docx파일
  - 소스 파일 이름
    - PPT에 제시
  - 과제보고서 양식
    - 스노우보드에서 다운로드
1. 실행 결과 화면 캡처한 이미지 첨부
  2. 소스 코드
  3. 소스 코드에 대한 설명 (간략하게 3-4줄)

- 조교 메일로 질문 보내기
  - 4ra@sookmyung.ac.kr

- 질문시 주의사항

"• 메일에 반드시 과목, 분반, 전공, 이름, 학번 명시

제목 : 프로그래밍개론 001분반] 2331297 유사라 Lab01 질문 드립니다.

- 몇 번 과제에서 어떤 부분이 막혔는지, 어떤 부분이 문제인지 코드와 함께 설명 첨부  
(그냥 코드만 보내면 어디가 문제인지 알 수 없어요)
- 답장이 늦을수도 있으니 이 점 고려하여 미리 질문 (특히 과제 제출 마지막날 유의!)
- 질문 내용을 구체적으로 명확하게 적어 주시기 바랍니다.
- 오류 메시지를 첨부하고 싶을 경우, 오류 캡처 화면 + 전체 코드 c 파일을 첨부하여 보내주세요. (코드 캡처 사진 X)
- 그 외 출석 등 다른 질문들도 메일로 "