

## 프는 3<sup>rd</sup> 수업에서 다루어지는 것들 - 변수와 자료형

- \* 알고리즘
- \* 순서도(flow chart)
- \* sizeof 연산자
- \* 리터럴(literal) 상수
- \* 기호(symbolic)상수
- \* 이스케이프 시퀀스(escape sequence)
- \* 십진수, 8진수, 16진수
- \* 데이터 자료형(data type)

### //scanf 복습 & more

```
scanf("%d", &변수);
scanf("%d %d", &변수1, &변수2); // 두 정수를 한 줄에서 입력받을 때
```

### // 변수 more

```
//변수 선언
```

```
// 값 배정
```

```
// 값 입력
scanf("%d", &iVar);
scanf("%c", &cVar);
scanf("%f", &fVar);
scanf("%lf", &dVar);
```

```
// 값 출력
printf("값은 %d\n", iVar);
printf("값은 %c\n", cVar);
printf("값은 %f\n", fVar);
printf("값은 %lf\n", dVar);
```

```
// detail
```

### 상수

- literal 상수; 168 1;

- symbolic 상수:

방법1:(변수 선언처럼)  
const int DAYS = 7

방법2(함수 외부에 위치):  
#define DAYS 7

I love errors!

Errors are your best teacher.

### 짝(Pair Programming)?

짝(Pair Programming)은 한 컴퓨터에서 두 명의 프로그래머가 설계, 알고리즘, 코드, 테스트 등을 협력하여 완성하는 프로그래밍 방식이다. 두 명중 한명이 드라이버(Driver)는 컴퓨터 앞에서 타이핑을 하거나 알고리즘을 적는다. 나머지 한명인 파트너(Partner)는 바로 옆에서 그것을 관찰하여 오류를 발견하거나 대안을 생각한다.

둘은 대화를 나누며 합의된 내용들을 구현한다. 중간에 드라이버와 파트너는 자주 그역할을 교체할 수 있다.

짝프로그래밍으로 프로그래밍을 배워보자~

짝프로그래밍을 사용하여 어떤 방식의 교육이 이루어질 수있는가?

가장 중요한 점은 교육의 참여자가 단순히 기존의 선생과 학생에 국한되지 않고 동등한 학생끼리 서로 같은 과제를 수행하면서 서로의 지식을 배울 수도 있게 된다는 것이다. 예를 들어 특정 강좌의 수강생들끼리 주어진 과제를 함께 짝 프로그래밍하면서 이제까지 배웠던 지식들을 공고히 할 수 있다.

지도 교육

어떤 분야에 대해 학생보다 나은 지식을 가지고 있는 사람이 짝 프로그래밍 방식을 통해 그를 가르치는 형태이다. 선생이 간단한 과제를 제시하고 학생이 그것을 달성 할 수 있도록 보조자 역할을 한다. 이때 선생도 상대방을 가르치면서 미처 몰랐던 부분이나 확실하게 알지 못했던 부분을 깨닫는 경우가 많다. 이때 중요한 것은 상대방에게 입량적으로 지식을 전달하는 것이 아니라, 상대방이 스스로 문제를 풀어 나가도록 옆에서 힌트와 조언을 통해 스스로 깨닫게 유도하는 것이다.

협력 교육

어떤 한명이 월등한 권위를 갖고 상대방을 가르치는 것이 아니라 서로 대등한 관계에서 협력하여 동일한 프로그램을 완성해 나가면서 서로의 지식을 배운다. 짝 프로그래밍의 특별한 장점으로, 상대방에게 자신의 의도를 설명하고 또한 상대방의 의도를 이해하면서 서로 자신의 유용한 지식을 상대방에게 암시적으로 전파하게 한다.

```
#include <stdio.h>

int main(void)
{
    double x = 12345.23456789;
    int num;

    printf("그냥 출력 %f\n", x);
    printf("1 나누기 3은 %f\n", 1.0/3.0);
    printf("2 나누기 3은 %f\n\n", 2.0/3.0);

    printf("소수 3째자리까지 출력 %.3f\n", x);
    printf("소수 1째자리까지 출력 %.1f\n\n", x);

    printf("%e를 사용한 지수형 출력 %e\n", x);
    printf("%E를 사용한 지수형 출력 %E\n", x);

    printf("I said % Hi, Sue!%\n");
    printf("I am 100%% sure.\n");
    printf("%%\n");

    num = 11;
    printf("%0tt%dtt%x\n", num, num, num);
    num = 011; // 십진수로 ??
    printf("%0tt%dtt%x\n", num, num, num);
    num = 0x11; // 십진수로 ??
    printf("%0tt%dtt%x\n", num, num, num);
}
```

## LAB 3

(공지사항)HW3 부터 프로그램의 \*제일\* 앞부분에 아래의 내용을 주석문으로 달아주기 바랍니다.

```
/******HW3 부터 프로그램의 *제일* 앞부분에 아래의 내용을 주석문으로 달아주기 바랍니다.
HW3_1 박수희 19950000 컴퓨터학과 17 학년
*****/
```

■ LAB3\_0(연산자 연습) 다음의 출력결과를 예측하라. 실제 실행시켜 자신의 예측과 비교해보라.

```
#include <stdio.h>
int main(void)
{
    int x = 10, y = 20;
    int result1, result2, result3;

    printf("두 수의 덧셈      : %d \n", x + y);

    x += 3;
    y *= 4;
    printf("result : %d %d\n", x, y);

    //단항연산자
    x = -x;
    printf("x=-x 연산 이후 x은 %d \n", x);

    x = 10;
    y = 10;
    printf("선 연산 후 증가 : %d \n", x++);
    printf("다시 한번 출력      : %d \n\n", x);
    printf("선 증가 후 연산 : %d \n", ++y);
    printf("다시 한번 출력      : %d \n", y);

    x = 10;
    y = (x--) + 2;
    printf("x : %d \n", x);
    printf("y : %d \n", y);

    //관계연산자
    x = 10, y = 10; //콤마연산자
    result1 = (x == y);
    result2 = (x <= y);
    result3 = (x > y);
    printf("result1 : %d \n", result1);
    printf("result2 : %d \n", result2);
    printf("result3 : %d \n", result3);

    //논리연산자
    x = 10;
    y = 20;
    result1 = (x == 10 && y == 10);
    result2 = (x == 10 || y == 10);
    result3 = !(x == 10);
    printf("result1 : %d \n", result1);
    printf("result2 : %d \n", result2);
    printf("result3 : %d \n", result3);

    return 0;
}
```

### ■ LAB3\_1(scanf 문, 정수형 변수)

사용자로부터 두 개의 정수를 입력받아서, 덧셈, 뺄셈, 곱셈, 나눗셈의 결과를 출력하는 프로그램을 작성하려한다.

실행결과 1: (밑줄친 부분은 입력)

```
Enter the first number: 50
Enter the second number: 10
50 + 10 = 60
50 - 10 = 40
50 * 10 = 500
50 / 10 = 5
```

실행결과 2: (밑줄친 부분은 입력)

```
Enter the first number: 7
Enter the second number: 11
7 + 11 = 18
7 - 11 = -4
7 * 11 = 77
7 / 11 = 0
```

프로그램을 작성하기 위한 준비작업으로 아래에 답하라.

☐ 필요한 변수들은 무엇인가 생각해보라.

☐ 최소한으로 필요한 변수는 2개이다. 이 두 변수들의 이름을 정하여 **변수를 선언**하라.

☐ **입력**을 처리하는 부분을 위한 코드를 완성하라.

```
printf("Enter the first number: ");
scanf(                );
printf("Enter the second number: ");
scanf(                );
```

☐ **계산과 출력**을 한꺼번에 처리하는 부분을 위한 코드를 완성하라.

```
printf(                );
printf(                );
printf(                );
printf(                );
```

☐ 위의 작업을 이용하여 프로그램을 작성하고 위의 실행결과처럼 실행해보라. 주어진 실행결과 외에 다른 입력으로도 실행해보라.

### ■ LAB3\_2(scanf 문, 정수형 변수)

사용자로부터 분을 입력받아 이를 시와 분으로 출력하는 프로그램을 작성하려한다.

실행결과 1: (밑줄친 부분은 입력)

```
Enter the total minutes: 130
130 minutes
2h 10m
```

실행결과 2: (밑줄친 부분은 입력)

```
Enter the total minutes: 345
345 minutes
5h 45m
```

- 필요한 변수들은 무엇인가 생각해보라.
- 3 개의 변수를 사용하려한다. 이름을 정하여 **변수 선언**하라.

- **입력**하는 부분을 완성하라.

```
printf("Enter the total minutes: ");
scanf(                );
```

- 시와 분을 **계산**하는 부분을 작성하라.

- 위의 실행결과처럼 **출력**하는 부분을 작성하라.

- 위의 작업을 이용하여 프로그램을 작성하고 위의 실행결과처럼 실행해보라. 변수 3 개를 사용하라. 주어진 실행결과 외에 다른 입력으로도 실행해보라.

### Tip

일반적으로 프로그래밍은

1. 변수선언
2. 입력
3. 계산
4. 출력

으로 구성된다. (물론 LAB3\_1 의 경우처럼 계산과 출력이 하나로 합쳐진 경우도 있다.)  
대체로(항상은 아니지만) 위의 각 분류간에 **빈 줄**을 넣는 것이  
코드를 논리적으로 분류하여 프로그램을 이해하기 쉽도록 하여준다.

LAB 3\_1 의 경우

```
변수선언
(빈줄)
입력
(빈줄)
계산과 출력
```

LAB3\_2 의 경우

```
변수선언
(빈줄)
입력
(빈줄)
계산
(빈줄)
출력
```

### ■ LAB3\_3(scanf 문, 정수형 변수)

사용자로부터 시와 분을 입력받아서 총 분을 출력하는 프로그램을 만들려한다.

실행결과 1: (밑줄친 부분은 입력)

```
Enter hour: 2
Enter minute: 10
Total 130 minutes
```

실행결과 2: (밑줄친 부분은 입력)

```
Enter hour: 5
Enter minute: 45
Total 345 minutes
```

위의 LAB3\_2 처럼  
변수선언,  
입력,  
계산,  
그리고 출력  
의 순서로 코드를 구상하면서 프로그램을 작성하여 위의 실행결과처럼 실행해보라. 변수 3 개를 사용하라. 주어진 실행결과 외에 다른 입력으로도 실행해보라.

힌트: LAB3\_2 에서 사용된 변수와 동일하게 할 수 있다.

### ■ LAB3\_4 삼각형의 넓이 계산

아래와 같은 실행결과를 내는 프로그램을 작성하라. 즉, 밑변과 높이를 입력받아 넓이를 계산하여 출력하는 프로그램이다. 변수 3 개를 사용하여 프로그램을 작성하라.  
하나의 scanf 를 사용하라.

```
C:\ *C:\DOCUMENTS AND SETTINGS\WUSER\바탕 화면\W++참고\
```

```
밑변과 높이를 입력하세요: 10 20
밑변과 높이가 각각 10와 20인 삼각형의 넓이는 100이다
Press any key to continue
```

### Tip: 변수의 이름(naming)

1. 변수의 이름은 의미있는 영문으로 만드는 것이 보통이며, 한 단어인 경우 모두 소문자를 사용한다.  
예: base(밑변), height(높이), area(넓이), width(가로), circumference(둘레)
2. 변수가 길 때는 약자로 이름을 만들기도 한다. 물론 긴 이름을 그대로 사용하기도 한다.  
예: number -> num, standard -> std
3. 두 단어를 붙여서 하나의 변수를 만들 때는 두 번째 이상의 단어에서의 첫 자는 대문자를 사용한다.  
예: totalMinutes, studentNum, rightAnswer

### HW 3

■ **HW3\_1**(scanf 문, 정수형 변수)

사용자로부터 두 개의 점수를 입력받아서, 이 두수의 평균을 출력하는 프로그램을 작성하라.

- 세 개의 변수를 사용하라. 변수의 이름은 score1, score2, average 로 한다.

실행결과 1: (밑줄친 부분은 입력)

Enter two scores: 80 90  
The average is 85.

실행결과 2: (밑줄친 부분은 입력)

```
Enter two scores: 100 95
The average is 97.
```

■ HW3\_2(scanf 문, 정수형 변수)

사용자로부터 초를 입력받아 이를 시와 분과 초로 출력하는 프로그램을 작성하라.

- 4 개의 변수 (총초, 시, 분, 초)를 사용한다.
- 계산식은 다음과 같다.
  - 시= 총초 / 3600
  - 분= 총초 % 3600 / 60
  - 초= 총초 % 3600 % 60 (혹은 초 = 총초 % 60 도 같은 결과..)

실행결과 1: (밑줄친 부분은 입력)

```
Enter the total seconds: 3800
---Calculation Result---
3800 seconds
1h 3m 20s
```

실행결과 2: (밑줄친 부분은 입력)

```
Enter the total seconds: 15000
---Calculation Result---
15000 seconds
4h 10m 0s
```

### ■ HW3\_3 (scanf 문, 정수형 변수)

사용자로부터 시, 분, 초를 입력받아서 총 초를 출력하는 프로그램을 작성하라. 4개의 변수를 사용하라.

실행결과 1: (밑줄친 부분은 입력)

```
Enter h m s: 1 3 20
---Calculation Result---
Total 3800 seconds
```

### 실행결과 2: (밑줄친 부분은 입력)

```
Enter h m s: 4 10 0
---Calculation Result---
Total 15000 seconds
```

### 실행결과 3: (밑줄친 부분은 입력)

```
Enter h m s: 0 2 10
---Calculation Result---
Total 130 seconds
```

■ **HW3\_4** 삼각형의 넓이 계산, 직사각형의 둘레와 넓이 계산

요구사항: 위의 LAB3\_4에서 사용한 변수에 가로, 둘레에 대한 두 개의 변수만 새로이 추가한다.

아래와 같은 실행결과를 내는 프로그램을 작성하라.

ex) C:\DOCUMENTS AND SETTINGS\USER\바탕 화면\W+참고\WHW3\_4\WDe  
 밑변과 높이를 입력하세요: 10 20  
 밑변과 높이가 각각 10와 20인 삼각형의 넓이는 100이다  
 직사각형의 가로와 높이를 입력하세요: 10 20  
 가로와 높이가 각각 10와 20인 직사각형의 둘레와 넓이는 60와 200이다  
 Press any key to continue

## 프로그래밍 참고사항:

### Indentation: white character의 적절한 사용

**white character** - 화면 상에 나오지 않는 문자로 다음의 세 가지가 여기에 해당됩니다.

[Space bar] : 공백문자, 빈칸  
[Tab] : 탭  
[Enter] : 줄바꿈, 빈줄

프로그램을 작성할 때는 이 white character단위로 프로그램 식별자의 구분이 이루어집니다.

아래의 네 가지 경우를 다 같은 것으로 인식합니다.

```
int[Space bar]height;  
int[Tab]height;  
int[Space bar][Space bar]height;  
int[Enter]height;
```

이러한 white character를 사용하여 프로그램을 할 때 일정한 규칙(관습?)이 있으며 좋은 프로그래머라면 이러한 규칙하에 프로그램을 작성합니다.

(규칙은 사람에 따라 다를 수 있으나 적어도 자신의 프로그램에서는 일관된 규칙을 사용하여야 합니다!)

많이 쓰이는 규칙(제가 좋다고 생각하는)을 소개합니다. 앞으로 이 수업에서는 이 규칙을 따르기 바랍니다.

(참고로, 여러분이 다른 프로그래밍 수업을 들을 때 다른 교수님은 조금 상이한 규칙을 요구할 수 있습니다. 또한 여러분이 회사에 일할 때 여러분의 상사가 조금 다른 규칙을 요구할 수도 있습니다...)

#### ■ 띄어쓰기 관련 1: white character의 적절한 사용

// [Space bar]는 로 표시합니다.

// [Space bar]나 [Enter]는 연이어 쓰지 않습니다. (즉 한번에 하나만 사용)

```
#include <stdio.h>  
int main(void)  
{  
    [Tab] int minutes; // 식별자 간에는 하나의  를 넣고 ;앞에는  를 넣지 않습니다.  
    [Tab] int hour, minute; // ,앞에는  를 넣지 않고 뒤에는  를 넣습니다.  
    [Tab] minutes = 130  
        // 이항 연산자(=, +, -, *, /, %, &&, ||, <, >, <= 등등)과 연산항 사이에는  를 넣음  
        // 단항 연산자(예: ++, --, ! 등등)와 연산항 사이에  를 넣지 않음  
    [Tab] hour = minutes / 60;  
    [Tab] minute = minutes % 60;  
    [Tab] printf("%d seconds:\n", minutes); // \t과 \n의 앞뒤에는  를 넣지 않는다  
    [Tab] printf("%dh\t%dm\n", minute, second);  
    [Tab] return 0;  
}
```

문장과 }사이에는 빈줄을 넣지 않는다

#### ■ 띄어쓰기 관련2: 아래의 두 프로그램 중 어느 것이 더 읽기가 쉬운가요?

```
#include <stdio.h>  
int main(void)  
{  
    int minutes = 130;  
    int hour, minute;  
    hour = minutes / 60;  
    minute = minutes % 60;  
    printf("%d분:\n", minutes);  
    printf("%dh %dm\n", hour, minute);  
    return 0;  
}
```

```
#include <stdio.h>  
int main(void)  
{  
    int minutes = 130;  
    int hour, minute;  
    hour = minutes / 60;  
    minute = minutes % 60;  
    printf("%d분:\n", minutes);  
    printf("%dh %dm\n", hour, minute);  
}
```

선언부

처리부

출력

⇒ 오른쪽의 프로그램처럼 적절하게 빈 줄을 넣는 것이 좋은 프로그램입니다.

⇒ 여기에 정답은 없습니다. 여러분이 어떤 글을 쓸 때 적절히 문단을 나누듯이, 프로그램에서도 적절하게 빈 줄을 넣어줄 수 있습니다.

⇒ 보통(반드시 그런 것은 아니지만) 선언부분, 입력부분, 처리부분, 출력부분으로 나눌 수 있으며 이에 따라 혹은 의미, 복잡도에 따라서 빈 줄을 적절히 넣어줍니다.

좋지 않은 예	괜찮은 예
<pre>##1 #include &lt;stdio.h&gt; int main(void) {     int hour, minute;     int minutes;      printf("Enter 시간:");     scanf("%d", &amp;hour);      printf("Enter 분:");     scanf("%d", &amp;minute);     minutes = hour * 60 + minute;      printf("%d시간 %d분\n", hour, minute);     printf("총 %d분\n", minutes); }</pre>	<pre>##2 #include &lt;stdio.h&gt; int main(void) {     int hour, minute;     int minutes;      printf("Enter 시간:");     scanf("%d", &amp;hour);     printf("Enter 분:");     scanf("%d", &amp;minute);      minutes = hour * 60 + minute;     printf("%d시간 %d분\n", hour, minute);     printf("총 %d분\n", minutes); }</pre>
<pre>##3 #include &lt;stdio.h&gt; int main(void) {     int hour, minute;     int minutes;     printf("Enter 시간:");     scanf("%d", &amp;hour);      printf("Enter 분:");     scanf("%d", &amp;minute);      minutes = hour * 60 + minute;      printf("%d시간 %d분\n", hour, minute);     printf("총 %d분\n", minutes); }</pre>	<pre>##4 #include &lt;stdio.h&gt; int main(void) {     int hour, minute;     int minutes;      printf("Enter 시간:");     scanf("%d", &amp;hour);     printf("Enter 분:");     scanf("%d", &amp;minute);      minutes = hour * 60 + minute;      printf("%d시간 %d분\n", hour, minute);     printf("총 %d분\n", minutes); }</pre>

## LAB 5

■ **LAB5\_0(형 변환)** 아래의 코드를 잘 살펴보고 결과를 예측한후 실제 수행결과와 비교해보라.

```
#include <stdio.h>
int main(void)
{
    char ch;
    int i, j, k;
    double x, y, z;

    // 자동 형 변환
    ch = 'A' + 1; // +연산자는 문자형과 정수형을 연산항으로 가지므로 문자형'A' ' 정수형 65로 변환
    printf("ch의 아스키값은 %d이고 문자로 표현하면 %c\n", ch, ch);
    ch = 'A' + 32;
    printf("ch의 아스키값은 %d이고 문자로 표현하면 %c\n\n", ch, ch);

    i = 9.99;
    // 소수점이하값이 잘려나감으로 Compile시 우리에게 warning을 준다. 그냥 진행해도 좋다.
    printf("자동 형 변환 결과 : i = %d\n", i);

    x = 10; // 여기서 어떤 변환이 일어나는가? 정수형 상수 10이 double형 상수 10.0이 된다
    printf("자동 형 변환 결과 : x = %f\n", x);

    x = 10 + 9.99;
    printf("자동 형 변환 결과 : x = %f\n\n", x);
    // 정수형 상수 10이 double형 10.0이 된다.
    // 이는 double형 9.99로 인하여 실수형 덧셈이 되었기 때문이다.

    j = 10; k = 4;
    y = 10.0; z = 4.0;

    printf("강제 형 변환(Casting에 의한)\n");
    x = (double) j / k;
    printf("1: x = %f\n", x);
    x = (double) (j / k);
    printf("2: x = %f\n\n", x);

    x = (int) y / z;
    printf("3: x = %f\n", x);
    x = (int) (y / z);
    printf("4: x = %f\n\n", x);

    x = j / k * k;
    printf("5: x = %f\n", x);
    x = (double) j / k * k;
    printf("6: x = %f\n", x);
    x = (double) (j / k) * k;
    printf("7: x = %f\n", x);
    x = j / (double) k * k;
    printf("8: x = %f\n", x);
    x = j / k * (double) k;
    printf("9: x = %f\n", x);
}
```

## 유의사항

- **변수**의 이름붙이기
  - 소문자와 대문자의 적절한 사용, 시작은 반드시 소문자로
  - 의미있는 이름으로 붙인다.
    - 예 score, score1, score2, average
  - 대부분은 소문자로, 둘 이상의 단어를 붙일때는 이음 단어의 첫 문자를 대문자로
    - 예: studentNumber, totalMinutes, maxScore
- **심볼릭 상수**의 이름은 모두 대문자로 한다. 두개 이상의 단어를 합칠때는 주로 \_(언더바)를 사용
  - 예: PI, MAX\_NUMBER

## ■ LAB5\_1(문자형, 정수형, 실수형 변수)

아래와 같은 실행결과를 갖는 간단한 프로그램을 작성하려한다.

```
C:\WINDOWS\SYSTEM32\cmd.exe
C:\WINDOWS\SYSTEM32\cmd.exe
Enter your gender: M
Enter your age: 35
Enter your height: 173.5
성별      나이      키
M          35      173.500000
Press any key to continue.
```

- 필요한 변수들의 이름과 그 각각의 자료형을 생각해보라. 이를 선언하라. 실수형을 위해서는 double을 사용하라.

- 이를 입력하는 부분을 작성하라. 이때 서식문자에 유의하라. (아래의 표 참조)

	입력시(scanf문)	출력시 printf문)
정수형 int	%d	%d
문자형 char	%c	%c
실수형 float	%f	%f
실수형 double	%lf(주의할 것!)	%f

```
printf("Enter your gender: ");
scanf(
);
printf("Enter your age: ");
scanf(
);
printf("Enter your height: ");
scanf(
);
```

- 이를 출력하는 부분을 작성하라.

- 위의 작업을 이용하여 위와 같은 실행결과가 나오도록 프로그램을 작성하라.

■ **LAB5\_2**(서식문자 %d 와 %c 의 사용, 교재 p610 의 부록 Appendix-A <아스키 코드 표>참조)

□ 아래의 printf 문을 살펴보자. 출력결과는 무엇일까 예측해보라.

```
printf("%d %c\n", 'A', 'A');
printf("%d %c\n", 65, 65);
```

위를 이해한 후, 아래와 같은 일을 하는 프로그램을 아래의 순서로 작성하라.

**단계 1:** 알파벳 문자를 입력받아서 그에 해당하는 아스키값을 출력한다. 문자형 변수 ch 를 사용한다.

```
실행결과 1
Enter an alphabet: A
A 65
```

```
실행결과 2
Enter an alphabet: a
a 97
```

**단계 2:** 위의 실행결과에  
아스키 값을 입력받아서 그에 해당하는 문자를 출력하는 부분을 추가하라. int 변수 value 를 추가한다.

```
실행결과 1
Enter an alphabet: A
A 65
Enter a ascii value: 66
66 B
```

```
실행결과 2
Enter an alphabet: a
a 97
Enter a ascii value: 99
99 c
```

```
프로그램의 구성
변수 ch 와 value 의 선언;
ch 를 읽어들인다;
ch 와 그 아스키값을 출력;
value 를 읽어들인다;
value 와 이 값에 해당하는 문자를 출력;
```

■ **LAB5\_3**(정수, 실수의 변수, const 사용 상수)

반지름 값을 입력으로 받아서 넓이를 계산하여 출력하는 프로그램을 작성하라.

유의사항

- 3.141592 를 저장하는 심볼릭 실수형(double) 상수사용(const 키워드 사용)
- 반지름을 위한 정수형(int) 변수 radius
- 넓이를 위한 실수형(double) 변수 area

```
실행결과 1
Enter a radius: 10
The area of a circle with 10: 314.159200.
```

```
실행결과 2
Enter a radius: 5
The area of a circle with 5: 78.539800.
```

## HW 5

■ **HW5.1**(%c 의 사용, 아스키 코드 이해, 교재의 p102 참조)

알파벳 문자를 읽어들여서  
그에 해당하는 아스키값,  
다음 문자와 그 아스키 값,  
그리고 그 문자의 소문자와 그 아스키값을 출력하는 프로그램을 작성하라.  
- 변수는 char 형 ch 만을 사용한다.

```
실행결과 1
Enter an upper letter(A-Y): A
Character given is A(65)
The next character is B(66)
The lower case letter is a(97)
```

```
실행결과 2
Enter an upper case letter(A-Y): C
Character given is C(67)
The next character is D(68)
The lower case letter is c(99)
```

■ **HW5.2**(정수, 실수의 변수)

LAB5\_3 에 둘레를 계산하여 출력하는 부분을 추가하라.

```
실행결과 1
Enter a radius: 10
The area of a circle with 10: 314.159200
The circumference of a circle with 10: 62.831840
```

```
실행결과 2
Enter a radius: 5
The area of a circle with 5: 78.539800
The circumference of a circle with 5: 31.415920
```

■ **HW5.3**(double 형 변수, %lf %f 의 사용 연습, const 사용 상수)

섭씨 온도를 읽어들여서 화씨로 출력하는 프로그램을 작성하라.  
식은 다음과 같다. 화씨 = 섭씨 \* 9.0 / 5.0 + 32

- 변수로는 double 형 celsius 와 fahrenheit 를 사용하라.
- const 를 사용하여 심볼릭 상수 BASE(정수형, 값 32)와 CONVERSION\_FACTOR(double 형, 값 9.0/5.0)를 선언하고 이를 사용하라.(#define 사용)

```
실행결과 1
Enter a celsius temperature: 24.9
Fahrenheit Equivalent : 76.820000
```

```
실행결과 2
Enter a celsius temperature: 40.5
Fahrenheit Equivalent : 104.900000
```