**Computer Algorithm**

**Chapter 01**

**C Practice**

**[실습] C언어를 이용한 간단한 알고리즘 구현**

이번 실습을 통해 간단한 알고리즘을 사용하여 문제를 해결하는 연습을 한다.

**실습 목표**

* **Event-driven 시스템을 학습하고 이해한다.**
* **간단한 Event-driven 시스템을 구현할 수 있다.**
* **랜덤 수를 활용한 Monte-Carlo법을 이용하여 문제를 해결할 수 있다.**
* **rand()함수를 사용할 때 시드를 주는 방법을 파악하고 이를 해결할 수 있다.**

**요구사항**

**실습과제 1) (20점)**

**사용자의 입력을 게속 받는 프로그램을 구현한다.**

**1을 입력 시 “Welcome to Woong-Sup’s Algorithm Class”**

**2을 입력 시 “GoodLuck for 2020 class”**

**3을 입력 시 프로그램 종료**

**입력과 출력과 프로그램 종료는 C언어 표준입출력 라이브러리를 사용하고 각 문장은 출력 후 줄바꿈이 되어야 한다.**

**실습과제 2) (80점)**

**0과 10사이의 정수의 쌍 (x, y)를 난수로 1000번 발생시켜 1/4원 내부의 난수 쌍과 외부의 난수 쌍을 이용하여 원주율** π**를 구한다.ㅇ**

**배경지식**

1)

Event-driven 시스템이란 사용자의 행동 (마우스 클릭, 키보드 입력), 센서, 또는 다른 프로그램에서 전송된 메시지에 의해 프로그램의 실행 내용이 결정되는 시스템을 말한다. 대부분의 컴퓨터 시스템은 event driven 방법에 의해 동작한다.

Windows 용 프로그램의 경우 Event driven 시스템을 구현 할 때 무한 루프를 사용하며 아래의 그림과 같은 형태로 동작한다.

No input

waiting

Keyboard event *e*

*e*

Routine 1

실행

Routine 2

실행

Routine *n*

실행

위 형태에 대한 의사(pseudo)코드는 다음과 같다

**while**(e = 키보드 입력 값)

**if** (e == 1)

run 'routine 1'

**else** (e == 2)

run 'routine 2'

**else**

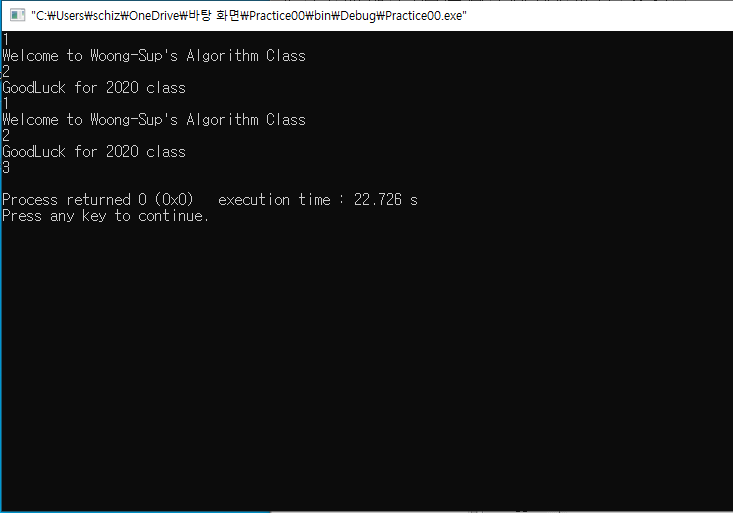
exit while loop

키보드 입력값은 scanf 함수를 통해 얻을 수 있다. Scanf 함수는 입력 값이 키보드에서 주어질 때 까지 프로그램을 대기 상태로 만들어준다.

예)

**int** e;

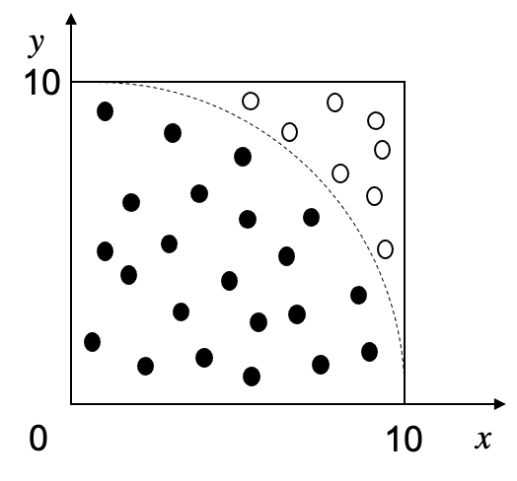
scanf(%d, &i);



소스코드 ex-1의 실행화면

2)

0 < *x,y* < 10 의 조건을 가지는 점 (*x,y*)를 랜덤하게 다량으로 생성할 경우 각 점들은 아래의 그림과 0과 10 사이의 사각형 안에 위치하게 된다. 이 때 사각형 내에 1/4원을 만들고 원 안에 생성되는 점들을 검은 색으로 원 밖에 생성되는 점들을 흰색으로 표시하면 아래 그림과 같다.



위 그림에서 검은색 점들의 수를 a, 흰색 점들의 수를 b라고 가정할 때, a와 b의 관계를 유추함에 의해 값을 유추할 수 있다. 아래는 a와 b의 관계를 나타내는 관계식이며 아래의 식에 따라 값을 구할 수 있다.

단 위의 방법은 난수를 통해 값을 유추하는 방법이므로 충분한 수의 난수를 발생시켜야 적절한 값을 얻을 수 있다는 점을 잊지 말아야 한다.

몬테카를로법은 난수(random number)를 발생시켜 문제를 해결하는 방법이다. 따라서 C언어에서 난수를 발생시키는 방법을 알고 있어야 한다. 난수란 특정 배열순서나 규칙을 가지지 않는 연속적인 임의의 수를 말하는 것으로 C언어에서는 rand() 함수를 사용하여 난수를 만들 수 있다.

rand() 함수는 <stdlib.h>에서 제공하고 있으므로 rand() 함수를 사용하기 위해서는 프로그램에 <stdlib.h> 파일을 헤더에 포함하여야 한다. 하지만 rand() 함수를 단독으로 사용한다면 rand() 함수 구현의 특성상 rand() 결과값이 언제나 일정한 패턴으로 생성되는 결과를 가진다. 따라서 시드값을 주는 방법을 사용하여 rand() 함수의 값이 일정한 패턴으로 생성되지 않도록 한다. 시드값은 srand() 함수를 사용하며 프로그램 실행시 한번만 선언되도록 한다. 처음에 선언된 시드값에 의해 랜덤 값을 반복적으로 생성할 수 있다.

//stdlib[스탠다드 라이브러리, 시드값이란 난수표에서 행을 의미, srand()는 행을 지정해준다.

ex)

// time(NULL)함수의 결과값을 사용하여 srand()에서 시드값 발생.

srand((unsigned int)time(NULL));

…

**for**( index = 0 ; index < MAX ; index++ )

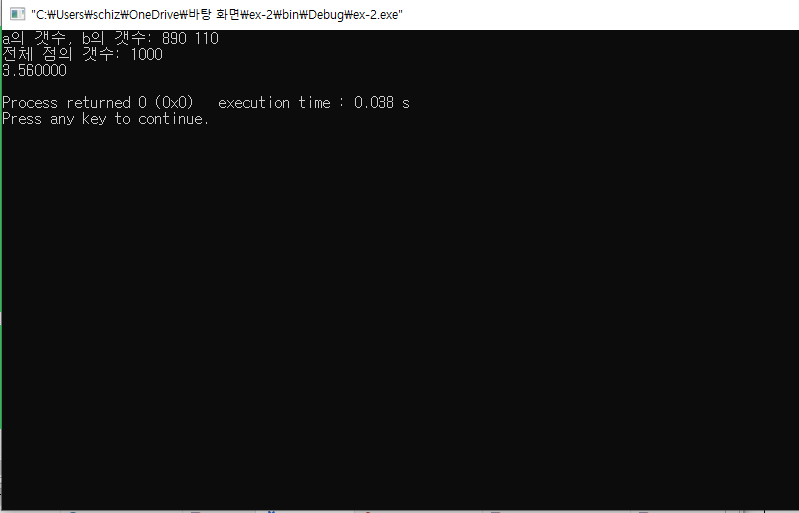
printf( "%d ", rand() ); // rand()를 사용하여 난수 발생

printf( "\n" ) ;

랜덤 함수에 의해 구해진 난수 값은 그 범위가 0 ~ RAND\_MAX인 정수 값으로 제한되어 있다. RAND\_MAX 값은 시스템에 따라 다른데 보통 32767 정도의 값을 가진다. 따라서 특정 범위(예컨대, 0 ~ 10)의 난수 값을 구하기 위해서는 다음과 같은 방식으로 난수 값을 구해야 한다.

(**int**)((**double**)rand()/RAND\_MAX \* 10)

위의 명령어는 rand()를 통해 구해진 값 (0에서 RAND\_MAX 사이의 값)을 RAND\_MAX로 나누어 0 ~ 1 사이의 값을 가지도록 하고 이를 10으로 곱하여 0 ~ 10 사이의 값을 가지도록 한 후 정수형으로 형변환하여 0 ~ 10 사이의 정수값을 가지도록 만드는 역할을 한다.



**제출방법**

* 보고서 작성방법: 각 실습문제 번호별로 결과가 나온 화면의 내용을 캡쳐하여 보고서에 붙여 놓는다.
* 소스코드의 파일이름에 연습문제 번호를 붙이는 것을 잊지 않는다.
  + 예) ex-1.c, ex-2.c
* 결과 보고서에 이름과 작성 날짜를 기입하는 것을 잊지 않는다.
  + 예) 김웅섭\_2020\_09\_01.doc
* 실행결과를 보고서에 작성하여 소스코드와 함께 e-class에 제출한다.
* 제출 마감 : e-class 제출 마감시간까지