



윈도우프로그래밍

Lab 02

- 난수 생성 함수를 이용한 Quiz 프로그램 -

김한슬

uo3359@sookmyung.ac.kr

난수 생성 함수 – rand()

- int rand(void)
 - <stdlib.h> 에 선언되어 있음
 - 0 ~ 32767 사이의 난수를 발생시킴
 - 난수의 범위 지정
 - $a \sim b$ 범위의 난수 생성: $a + \text{rand}() \% (b - a + 1)$
 - ex) $\text{rand}() \% 100$: 0 ~ 99 사이의 난수 생성
 $10 + \text{rand}() \% 11$: 10 ~ 20 사이의 난수 생성

난수 생성 함수 – rand() 예제

```
#include <iostream>
using namespace std;
#include <stdlib.h>

void main() {
    for (int i = 0; i < 9; i++)
        cout << rand() % 10 << "
";
    cout << rand() % 10 << endl;

    for (int i = 0; i < 9; i++)
        cout << rand() % 10 << "
";
    cout << rand() % 10 << endl;
}
```

- 첫 번째 실행

C:\Windows\system32\cmd.exe

```
1 7 4 0 9 4 8 8 2 4
5 5 1 7 1 1 5 2 7 6
계속하려면 아무 키나 누르십시오 . . .
```

- 두 번째 실행

C:\Windows\system32\cmd.exe

```
1 7 4 0 9 4 8 8 2 4
5 5 1 7 1 1 5 2 7 6
계속하려면 아무 키나 누르십시오 . . .
```

난수 초기 값 변경 – srand()

- void srand(unsigned seed)
 - <stdlib.h> 에 선언되어 있음
 - 난수 생성 공식의 초기 값을 seed를 반영하여 수정
- rand() 를 사용할 때, srand()를 사용하여 난수 초기 값을 변경시켜 주어야 프로그램 실행 할 때마다 다른 난수 열이 생성됨
- rand() 함수 실행 전 프로그램 앞 부분에 srand(unsigned) time(NULL))을 실행
- time()
 - <time.h> 에 선언되어 있음
 - 컴퓨터 내장 시계로부터 현재 시간을 읽어 옴

난수 초기 값 함수 – srand() 예제

```
#include <iostream>
using namespace std;
#include <stdlib.h>
#include <time.h>

void main() {
    for (int i = 0; i < 9; i++)
        cout << rand() % 10 << " ";
    cout << rand() % 10 << endl;

    srand((unsigned) time(NULL));
    for (int i = 0; i < 9; i++)
        cout << rand() % 10 << " ";
    cout << rand() % 10 << endl;
}
```

- 첫 번째 실행

C:\Windows\system32\cmd.exe

```
1 7 4 0 9 4 8 8 2 4
4 8 5 8 9 1 6 9 7 1
계속하려면 아무 키나 누르십시오 . . .
```

- 두 번째 실행

C:\Windows\system32\cmd.exe

```
1 7 4 0 9 4 8 8 2 4
9 4 4 0 7 7 7 6 0
계속하려면 아무 키나 누르십시오 . . .
```

과제 1: quiz1.cpp 작성

- 10 ~ 99 범위의 난수 2개 생성
- 생성된 난수를 사용한 덧셈 문제 제시
- 사용자가 답을 입력
- 입력한 답을 읽고 검사
- 답이 옳은지 여부에 따라 적절한 메시지를 출력 후 프로그램 종료

<예시 1>

$59 + 47 = 100$

Wrong answer!

<예시 2>

$76 + 56 = 132$

Correct!

과제 2: quiz2.cpp 작성

- quiz1.cpp 확장
- 사용자가 틀린 답을 입력: 다시 기회를 주도록 수정
- 정답이 나올 때까지 반복되도록 함

<실행 예시>
 $89 + 84 = 111$
Wrong answer! Try again.
 $89 + 84 = 123$
Wrong answer! Try again.
 $89 + 84 = 173$
Correct!

과제 3: quiz3.cpp 작성

- **quiz1.cpp 확장**
- 여러 덧셈 문제를 제시하는 반복 구조로 변환
- 각 반복 후 계속 진행할지 여부를 사용자에게 묻음 (입력: y or n)
- **quiz2.cpp의 재시도 기능은 포함시키지 않음**
- "M correct answers out of N problems" 결과 출력
 - N개의 문제 중 M개를 맞추었다는 의미

<실행 예시>

39 + 65 = 123

Wrong

계속하려면 Y, 아니면 아무 키나 : Y

15 + 50 = 65

Correct

계속하려면 Y, 아니면 아무 키나 : Y

24 + 41 = 65

Correct

계속하려면 Y, 아니면 아무 키나 : Y

49 + 99 = 123

Wrong

계속하려면 Y, 아니면 아무 키나 : n

2 correct answers out of 4 problems

과제 제출: 보고서

- 내용
 - 각 과제의 프로그램 소스
 - 각 과제의 실행 화면 캡처
- 제출 형식
 - 스노우보드에서 Lab02_이름_학번.doc 다운로드
 - 스노우보드에 제출: Lab02_이름_학번.zip
(솔루션 폴더: debug 폴더 제거, 보고서 파일)
- 제출 기한
 - 3월 23일(화) PM 11:59 까지