

[12주차 2강]

연산자/함수/자료형 심화(3)

학습 내용

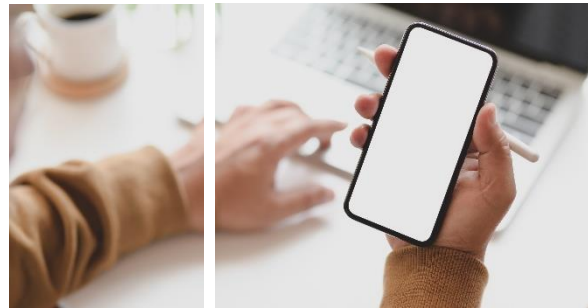
13.3 라이브러리 활용





학습 목표

📍 13.3 표준 라이브러리의 활용법을 익힌다.





1. 비트연산자
2. 재귀함수
3. 라이브러리 활용

3. 라이브러리 활용



난수(random number) 생성

- 난수란? 정의된 범위 내에서 임의로 추출되는 수
- C언어에서는 난수를 생성하는 함수 제공
- 관련 함수: rand() , srand() , time ()



실행 시간 측정

- C언어에서는 제공되는 시간 관련 함수를 사용하여 프로그램이 실행되는데 필요한 시간을 측정할 수 있음
- 관련 함수 : clock()



3. 라이브러리 활용

난수(random number) 생성

- `rand()` 함수 : `<stdlib.h>`에 선언되어 있음
 - ✓ 0~`RAND_MAX` 사이의 임의의 수 리턴
 - ✓ `RAND_MAX`는 `stdlib.h`에 정의된 상수 (`32767`)

```
#include <stdio.h>
#include <stdlib.h> // rand 함수 사용을 위해

int main() {
    int i;
    for( i=0 ; i < 5 ; ++i)    // 5개의 난수 생성
        printf(" %d", rand() );
    return 0;
}
```

✓ 위 프로그램을 여러 번 실행시켜보자.



3. 라이브러리 활용



난수(random number) 생성

- `srand()` 함수: `<stdlib.h>`에 선언되어 있음
 - ✓ `rand()` 함수의 시드(seed) 변경

```
#include <stdio.h>
#include <stdlib.h> // rand 함수 사용을 위해

int main() {
    int i;
    srand( 10 );    // 시드를 10으로 지정
    for( i=0 ; i < 5 ; ++i) // 5개의 난수 생성
        printf(" %d", rand() );
    return 0;
}
```

- ✓ `srand()` 함수의 인자를 변경해가면서 실행시켜보자.



3. 라이브러리 활용



난수(random number) 생성

- `time()` 함수 : `<time.h>`에 선언되어 있음
 - ✓ 현재 시스템의 시간에 의해 결정되는 정수 리턴
 - ✓ 실행할 때 마다 시드를 바꾸기 위해서 사용

```
#include <stdio.h>
#include <stdlib.h> // rand 함수 사용을 위해
#include <time.h>   // time 함수 사용을 위해

int main() {
    int i;
    srand( time(NULL) ); // 시드를 현재시간으로 지정
    for( i=0 ; i < 5 ; ++i) // 5개의 난수 생성
        printf(" %d", rand() );
    return 0;
}
```

✓ 실행할 때마다 결과가 달라짐



※ 실습하기



[예제 13.5] 0부터 100사이의 난수를 5개 출력하는 프로그램 작성

- 실행할 때 마다 다른 값이 나와야 함
- (hint) 난수로 생성된 값을 0~100 사이로 바꾸기 위해 나머지 연산자(%) 활용



3. 라이브러리 활용



[min, max) 사이의 난수 하나 생성하기

- ✓ % 연산자를 사용하는 방식 보다 아래 방식 권장

```
int random_num(int min, int max) {  
    int rand_num;  
  
    srand( time(NULL) );  
    rand_num = rand() / ((double)RAND_MAX + 1) * (max-min) + min;  
    return rand_num;  
}
```



3. 라이브러리 활용



[min, max) 사이의 난수 하나 생성하기

```
rand() / ((double)RAND_MAX + 1) * (max-min) + min
```

- ✓ `rand()` : 0~RAND_MAX 사이의 임의의 정수 생성
- ✓ `/ ((double)RAND_MAX + 1)` : 생성된 정수를 [0,1) 사이의 소수로 변환
- ✓ `* (max-min)` : [0,1) 사이의 소수를 [0,max-min) 사이의 값으로 변환
- ✓ `+ min` : 수를 min 만큼 이동시켜, [min,max) 사이의 수로 변환



3. 라이브러리 활용



실행 시간 측정

- `clock()` 함수 사용 : `<time.h>`에 선언되어 있음
 - ✓ 호출 당시의 시스템 시각을 `CLOCKS_PER_SEC` (`time.h`에 정의된 상수) 단위로 반환
 - ✓ 초 단위의 시간을 얻기 위해서는 `clock()` 함수에 의해 측정된 시각을 `CLOCKS_PER_SEC`로 나누어야 함



3. 라이브러리 활용



실행 시간 측정 코드

```
#include <stdio.h>
#include <time.h>      // clock() 함수 사용을 위해

int main( void ) {
    clock_t start, finish;
    double  duration;

    start = clock();    // 시작 시각
    ...                // 수행시간을 측정하고 하는 코드
    finish = clock();   // 종료 시각

    duration = (double)(finish-start) / CLOCKS_PER_SEC;

    printf("실행 시간: %lf 초\n", duration);
    return;
}
```





학습 정리

- **난수**란 정의된 범위 내에서 임의로 추출되는 수로 **rand()**, **srand()** 함수를 이용하여 난수를 생성할 수 있음
- 실행할 때마다 다른 난수를 만들기 위해 **time()** 함수를 활용함
- **clock() 함수**는 시스템 시각을 반환하는 함수로, 이를 사용하여 프로그램의 실행시간을 측정할 수 있음
- 초 단위의 시간을 얻기 위해서는 clock() 함수에 의해 측정된 시각을 LOCKS_PER_SEC로 나누어야 함

