



시스템프로그래밍기초 실습

Ch3. The Fundamental Data Types

실습 예제 1) types.c

```
#include <stdio.h>
int main(int argc, char *argv[])
{
    printf("The size of some fundamental types.\n\n");
    printf("char:%31u byte \n", sizeof(char));
    printf("short:%31u bytes\n", sizeof(short));
    printf("int:%31u bytes\n", sizeof(int));
    printf("long:%31u bytes\n", sizeof(long));
    printf("unsigned:%31u bytes\n", sizeof(unsigned));
    printf("float:%31u bytes\n", sizeof(float));
    printf("double:%31u bytes\n", sizeof(double));
    printf("long double:%31u bytes\n", sizeof(long double));
    return 0;
}
```

실습 예제 2) capitalize.c

```
1 #include <stdio.h>
2 int main(int argc, char *argv[])
3 {
4     int c;
5     while ((c = getchar()) != EOF) {
6         if (c >= 'a' && c <= 'z')
7             putchar(c + 'A' - 'a');
8         else
9             putchar(c);
10    }
11    return 0;
12 }
```

Math library (math.h)

Math 라이브러리는 표준라이브러리의 일종이지만 다음과 같이 라이브러리를 포함시킨다는 것을 컴파일러에게 알려줘야 한다.

gcc -o [결과파일명] [소스코드(.c)] -lm

예) **gcc -o power_square power_square -lm**

실습 예제 3) power_square.c

```
#include <math.h>
#include <stdio.h>
int main(int argc, char *argv[])
{
    double x;
    while (1) {
        printf("Input x: ");
        if (scanf("%lf", &x) != 1)
            break;
        if (x >= 0.0) {
            printf("\n%15s %22.15e\n", "sqrt(x)", sqrt(x));
            printf("\n%15s %22.15e\n", "pow(x,x)", pow(x,x));
        } else {
            printf("\nNumber must be nonnegative.\n");
            break;
        }
    }
    return 0;
}
```

과제 1) my_sqrt.c

```
#include <stdio.h>
#include <math.h>
double my_sqrt(double number)
{
    double sqrt;
    /* 이 부분을 작성하시오 */
    return sqrt;
}
int main(int argc, char *argv[])
{
    double x;
    printf("Insert non-negative number x : ");
    scanf("%lf", &x);
    printf("\nsqrt(x) = %10lf, sqrt(x)^2 = %.30e\n",
        my_sqrt(x), my_sqrt(x)*my_sqrt(x));
    return 0;
}
```

제곱근 구하기 힌트

양의 실수 a 에 대하여 다음 과정을 따라 \sqrt{a} 의 근삿값을 구할 수 있다.

1. 임의의 양의 실수 x_0 를 택한다. 이 값이 \sqrt{a} 에 가까울수록 더 빨리 근삿값을 구할 수 있다.
2. $x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right)$ 라 한다.
3. 원하는 정밀도까지 위의 과정을 반복한다.

위에서 구한 수열 $\{x_n\}$ 은 $\lim_{n \rightarrow \infty} x_n = \sqrt{a}$ 를 만족한다.

터미널 명령어 스크립트

모든 파일을 한꺼번에 컴파일 시키고 싶고
각 컴파일마다 **-lm**과 같이 조건이 까다롭고 기억하기 쉽지 않을 때,
다음과 같이 스크립트를 작성한다. 단번에 실행하는 명령어도 추가한다.

```
# vim compile-chap-3
```

compile-chap-3 내용

```
gcc -o types types.c
gcc -o capitalize capitalize.c
gcc -o power_square power_square.c -lm
gcc -o my_sqrt my_sqrt.c -lm
./types
./capitalize
./power_square
./my_sqrt
```

```
# chmod +x compile-chap-3
# ./compleie-chap-3
```