



시스템프로그래밍기초 실습 3주차

Ch2, 3.

C언어 Data Type 및 printf() 형식지정자

Type	형식지정자	의미	범위
<u>Arithmetic type specifiers</u>			
char	%c	문자	$[-2^7, 2^7-1]$
int	%d, %i	부호가 있는 정수, 10진수	$[-2^{31}, 2^{31}-1]$
long	%l	부호가 있는 정수,	$[-2^{63}, 2^{63}-1]$
float	%f	단일 정밀도 부동소수	유효숫자 7자리
double	%lf	2배 정밀도 부동소수	유효숫자 15자리
long double	%Lf, %LF	4배 정밀도 부동소수	유효숫자 33자리
<u>Modifiers</u>			
short	%h		$[-2^{15}, 2^{15}-1]$
signed		부호가 있는 타입	$[-2^{n/2}, 2^{n/2}-1]$
unsigned	%u	부호가 없는 타입	$[0, 2^n-1]$

실습 예제 1) types.c

- 유의사항: '1' 이 아닌 'l'

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("The size of some fundamental types.\n\n");
6      printf("char:%3lu byte\n", sizeof(char));
7      printf("short:%3lu bytes\n", sizeof(short));
8      printf("int:%3lu bytes\n", sizeof(int));
9      printf("long:%3lu bytes\n", sizeof(long));
10     printf("unsigned:%3lu bytes\n", sizeof(unsigned));
11     printf("float:%3lu bytes\n", sizeof(float));
12     printf("double:%3lu bytes\n", sizeof(double));
13     printf("long double:%3lu bytes\n", sizeof(long double));
14     return 0;
15 }
```

실습 예제 1) types.c 결과

```
spubuntu@sp:~/sysPro/spWeek3$ gcc -o types types.c
spubuntu@sp:~/sysPro/spWeek3$ ./types
The size of some fundamental types.

char:  1 byte
short: 2 bytes
int:   4 bytes
long:  8 bytes
unsigned: 4 bytes
float:  4 bytes
double: 8 bytes
long double: 16 bytes
```

실습 예제 2) rand.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int i, n;
7      printf("\n%s\n%s",
8             "Some randomly distributed integers will be printed.",
9             "How many do you want to See? ");
10     scanf("%d", &n);
11     for (i = 0; i < n; i++) {
12         if (i % 5 == 0)
13             putchar('\n');
14         printf("%12d", rand());
15     }
16     printf("\n\n");
17     return 0;
18 }
```

실습 예제 2) rand.c 결과

- 결과가 매번 같은 수로 나옴.

```
spubuntu@sp:~/sysPro/spWeek3$ ./rand
```

```
Some randomly distributed integers will be printed.  
How many do you want to See? 35
```

1804289383	846930886	1681692777	1714636915	1957747793
424238335	719885386	1649760492	596516649	1189641421
1025202362	1350490027	783368690	1102520059	2044897763
1967513926	1365180540	1540383426	304089172	1303455736
35005211	521595368	294702567	1726956429	336465782
861021530	278722862	233665123	2145174067	468703135
1101513929	1801979802	1315634022	635723058	1369133069

```
spubuntu@sp:~/sysPro/spWeek3$ ./rand
```

```
Some randomly distributed integers will be printed.  
How many do you want to See? 27
```

1804289383	846930886	1681692777	1714636915	1957747793
424238335	719885386	1649760492	596516649	1189641421
1025202362	1350490027	783368690	1102520059	2044897763
1967513926	1365180540	1540383426	304089172	1303455736
35005211	521595368	294702567	1726956429	336465782
861021530	278722862			

실습 예제 3) srand.c

- rand.c를 복사해 srand.c 파일을 생성.

```
$ cp rand.c srand.c
```

- srand()를 사용하여 seed(시작값)을 설정.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      int i, n, seed;
8      int count;
9      srand(seed);
10     printf("\n%s\n%s",
11           "Some randomly distributed integers will be printed.",
12           "How many do you want to See? ");
13     scanf("%d", &n);
14     count = 0;
15     while (1) {
16         count++;
17         printf("%12d", rand());
18         if (count == n) break;
19         if (count % 5 != 0) continue;
20         putchar('\n');
21     }
22     printf("\n\n");
23     return 0;
24 }
```

실습 예제 3) srand.c 결과

- 랜덤 숫자가 rand.c의 결과와 달리 매번 새로 갱신되어야 한다.

```
spubuntu@sp:~/sysPro/spWeek3$ ./srand
```

```
Some randomly distributed integers will be printed.
```

```
How many do you want to See? 15
```

```
323453843 1755314815 1155575753 1375735339 779333136
26274613 1468046456 513924813 1844536236 1951340876
523456081 109396356 1105244896 1351287848 2098549248
```

```
spubuntu@sp:~/sysPro/spWeek3$ ./srand
```

```
Some randomly distributed integers will be printed.
```

```
How many do you want to See? 12
```

```
1703526174 873433306 513492391 1764925903 2000486701
635321588 1770283233 2034176029 1199956173 1774489155
1583827449 849156870
```


Math library (math.h)

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

```
$ gcc -o [결과파일명] [소스코드(.c)] -lm
```

- 예시

```
$ gcc -o power_square power_square.c -lm
```

실습 예제 4) power_square.c

```
1  #include <stdio.h>
2  #include <math.h>    // sqrt(float), pow(float, float)
3
4  int main()
5  {
6      double x;
7      while (1) {
8          printf("Input x: ");
9          if (scanf("%lf", &x) != 1)
10             break;
11         if (x >= 0.0) {
12             printf("\n%15s %22.15e\n", "sqrt(x)=", sqrt(x));
13             printf("\n%15s %22.15e\n", "pow(x,x)=", pow(x, x));
14         } else {
15             printf("\nNumber must be nonnegative.\n");
16             break;
17         }
18     }
19     return 0;
20 }
```

실습 예제 4) power_square.c 결과

```
spubuntu@sp:~/sysPro/spWeek3$ ./power_square
Input x: 4

      sqrt(x)=  2.0000000000000000e+00
      pow(x,x)=  2.5600000000000000e+02
Input x: 5

      sqrt(x)=  2.236067977499790e+00
      pow(x,x)=  3.1250000000000000e+03
Input x: 9

      sqrt(x)=  3.0000000000000000e+00
      pow(x,x)=  3.8742048900000000e+08
Input x: -1

Number must be nonnegative.
spubuntu@sp:~/sysPro/spWeek3$
```

실습 과제 1) my_sqrt.c

```
1  #include <stdio.h>
2  #include <math.h>    // fabs(float)
3
4  double my_sqrt(double number){
5      double sqrt = number / 2;
6      int count = 0;
7
8      // 이 부분을 작성하시오.
9      // 오차 범위 0.0001의 정밀도까지 연산을 반복할 것.
10     // (오차범위 0.0001이면 break)
11     // fabs(float)를 사용하여 오차 범위를 계산할 것.
12     // ( fabs(0.003 - 0.0543) == 0.540 )
13     // 매 loop마다 현재 count와 sqrt값을 출력할 것. (결과물 그림 참고.)
14
15 };
16
17 int main()
18 {
19     double x;
20     printf("Insert non-negative number x : ");
21     scanf("%lf", &x);
22     double result = my_sqrt(x);
23     printf("\nsqrt(x) = %10lf, sqrt(x)^2 = %.30e\n",
24         result, result*result);
25     return 0;
26 }
```

실습 과제 1) 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.

실습 과제 1) my_sqrt.c 결과

```
spubuntu@sp:~/sysPro/spWeek3$ ./my_sqrt
Insert non-negative number x : 1024
count : 1          current sqrt : 257.000000
count : 2          current sqrt : 130.492218
count : 3          current sqrt : 69.169715
count : 4          current sqrt : 41.986941
count : 5          current sqrt : 33.187738
count : 6          current sqrt : 32.021254
count : 7          current sqrt : 32.000007
count : 8          current sqrt : 32.000000

sqrt(x) = 32.000000, sqrt(x)^2 = 1.0240000000000050022208597511053e+03
```

실습 과제 2) script_ch3

- vi editor로 script_ch3 파일 작성.

```
$ vi script_ch3
```

- 아래 내용 그대로 작성.

```
1 gcc -o types types.c
2 gcc -o rand rand.c
3 gcc -o srand srand.c
4 gcc -o power_square power_square.c -lm
5 gcc -o my_sqrt my_sqrt.c -lm
6 ./types
7 ./rand
8 ./srand
9 ./power_square
10 ./my_sqrt
```

- 실행 권한 설정. (change mode)

```
$ chmod +x script_ch3
```

- 실행

```
$ ./script_ch3
```

**실행 중,
프로그램
종료 단축키
: Ctrl + W**

실습 과제 2) script_ch3 결과

```
spubuntu@sp:~/sysPro/spWeek3$ ./script_ch3
The size of some fundamental types.

char: 1 byte
short: 2 bytes
int: 4 bytes
long: 8 bytes
unsigned: 4 bytes
float: 4 bytes
double: 8 bytes
long double: 16 bytes

Some randomly distributed integers will be printed.
How many do you want to See? 12

1804289383 846930886 1681692777 1714636915 1957747793
424238335 719885386 1649760492 596516649 1189641421
1025202362 1350490027

Some randomly distributed integers will be printed.
How many do you want to See? 7
323453843 1755314815 1155575753 1375735339 779333136
26274613 1468046456

Input x: 4

sqrt(x)= 2.0000000000000000e+00
pow(x,x)= 2.5600000000000000e+02
Input x: 9

sqrt(x)= 3.0000000000000000e+00
pow(x,x)= 3.8742048900000000e+08
Input x: -1

Number must be nonnegative.
Insert non-negative number x : 25
count : 1 current sqrt : 7.250000
count : 2 current sqrt : 5.349138
count : 3 current sqrt : 5.011394
count : 4 current sqrt : 5.000013
count : 5 current sqrt : 5.000000

sqrt(x) = 5.000000, sqrt(x)^2 = 2.500000000016777690348135365639e+01
```


과제 제출 파일

1. types.c

2. types

3. rand.c

4. rand

5. srand.c

6. srand

7. power_square.c

8. power_square

9. my_sqrt.c

10.my_sqrt

11.script_ch3

과제 제출 형식(1) (실습 시간 내에 완료 못한 경우)

1. 모든 파일은 sys_03 학번.tar.gz으로 압축하여 제출한다.

- tar.gz 압축하기

```
$ tar -zcvf [sys_03_자기학번.tar.gz] [폴더명]
```

- tar.gz 압축풀기

```
$ tar -zxvf [압축파일명.tar.gz]
```

옵션	설명
-c	파일을 tar로 묶음
-p	파일 권한을 저장
-v	묶거나 파일을 풀 때 과정을 화면으로 출력
-f	파일 이름을 지정
-C	경로를 지정
-x	tar 압축을 풀
-z	gzip으로 압축하거나 해제함



과제 제출 형식(2) (실습 시간 내에 완료 못한 경우)

2. 이메일 제목 :

[시프기] 03 이름 학번

형식 다를 시,
메일 수신 불가.

양인식 (3공 318호) : inshik@hanyang.ac.kr

조윤수 (4공 412호) : suejoe@hanyang.ac.kr



감사합니다.
