프로그래밍 연습

실습 #3

실습환경

martini.snucse.org

- 자신의 martini 계정/임시계정으로 실습컴퓨터 로그인
- window키>왼쪽 ‘terminal’ 켜기
- ssh {id}@martini.snucse.org
  ex)ssh PPmccl27@martini.snucse.org

Linux basic

I. man(manual) page 의 활용 (표준 함수 사용 방법 관련 문서)

```
~$ man –s 3 printf
```

```
PRINTF(3)                  Linux Programmer's Manual                  PRINTF(3)

NAME
       printf,  fprintf,  dprintf,  sprintf,  snprintf,  vprintf,  vfprintf, vdprintf,
       vsprintf, vsnprintf - formatted output conversion

SYNOPSIS
       #include <stdio.h>

       int printf(const char *format, ...);
       int fprintf(FILE *stream, const char *format, ...);
       int dprintf(int fd, const char *format, ...);
       int sprintf(char *str, const char *format, ...);
       int snprintf(char *str, size_t size, const char *format

       #include <stdarg.h>

       int vprintf(const char *format, va_list ap);
       int vfprintf(FILE *stream, const char *format, va_list
       int vdprintf(int fd, const char *format, va_list ap);
       int vsprintf(char *str, const char *format, va_list ap)
       int vsnprintf(char *str, size_t size, const char *forma

   Feature Test Macro Requirements for glibc (see feature_test
```

단락	설명
1	일반 명령어
2	시스템 호출
3	C 표준 라이브러리 함수들
4	특수 파일 (보통 /dev 에서 발견되는 장치 파일)과 드라이버
5	파일 형식과 convensions
6	게임과 화면 보호기
7	기타
8	시스템 관리 명령어와 데몬

실습

실습1

연산자 우선 순위를 코드 실행으로 직접 확인하기 위한 예제

1. vim ex3-1.c 입력합니다.
2. i를 눌러 입력모드로 변경한 후 다음 코드를 작성해 보세요.

```
#include <stdio.h>

int main(void)
{
    float a = 1, b = 2, c = 3, d = 4;
    printf("a = %f, b = %f, c = %f, d = %f₩n₩n", a, b, c, d);
    printf("a*b/c = %f₩n", a*b/c);
    printf("a*b/c = %.3f₩n₩n", a*b/c);
    printf("1+a*b = %f₩n", 1+a*b);
    printf("1+a*b = %03.0f₩n₩n", 1+a*b);
    printf("a = %2.1f, b = %2.1f, c = %2.1f, d = %2.1f₩n", a, b, c, d);
    printf("++a*b-c-- = %f₩n", ++a*b-c--);
    printf("a = %2.1f, b = %2.1f, c = %2.1f, d = %2.1f₩n", a, b, c, d);

    return 0;
}
```

실습1

➢ 코드분석

```
float a = 1, b = 2, c = 3, d = 4;
```

- 실수형 변수 a, b, c, d를 선언하고, 초기값 설정

```
printf("a = %f, b = %f, c = %f, d = %f₩n₩n", a, b, c, d);
```

- 변수 a, b, c, d 에 할당된 값 출력

✓ 출력 결과

```
a = 1.000000, b = 2.000000, c = 3.000000, d = 4.000000
```

실습1

➢ 코드분석

```
printf("a*b/c = %f₩n", a*b/c);
```

- *, / 연산자 우선 순위가 같으므로 associativity에 따라 왼쪽 부터 계산

```
printf("a*b/c = %.3f₩n₩n", a*b/c);
```

- 위 코드와 연산은 동일. %.3f format 으로 출력 (소수점 3자리까지)

✓ 출력 결과

```
a*b/c = 0.666667
a*b/c = 0.667
```

Operator precedence and associativity	
Operator	Associativity
() ++ *(postfix)* -- *(postfix)*	left to right
+ (*unary*) - (*unary*) ++ (*prefix*) -- (*prefix*)	right to left
* / %	left to right
+ -	left to right
= += -= *= /= etc.	right to left

실습1

➢ 코드분석

```
printf("1+a*b = %f₩n", 1+a*b);
```

- 연산자 우선 순위가 다르므로 우선 순위가 높은 * 부터 계산

```
printf("1+a*b = %03.0f₩n₩n", 1+a*b);
```

- 위 코드와 연산은 동일. %03.0f format 으로 출력
  (전체 3자리를 유지하고 빈 공간은 0으로 표기, 소수점 표기하지 않음)

✓ 출력 결과

```
1+a*b = 3.000000
1+a*b = 003
```

Operator precedence and associativity	
Operator	Associativity
() ++ *(postfix)* -- *(postfix)*	left to right
+ *(unary)* - *(unary)* ++ *(prefix)* -- *(prefix)*	right to left
* / %	left to right
+ -	left to right
= += -= *= /= etc.	right to left

실습1

➢ 코드분석

```
printf("a = %2.1f, b = %2.1f, c = %2.1f, d = %2.1f₩n", a, b, c, d);
```

- 변수 a, b, c, d를 소수 첫째자리까지 출력

```
printf("++a*b-c-- = %f₩n", ++a*b-c--);
```

- 연산자 우선 순위에 따라 ((++a)*b)-(c--) 와 같은 순위로 계산

```
printf("a = %2.1f, b = %2.1f, c = %2.1f, d = %2.1f₩n", a, b, c, d);
```

- ++, -- 연산자에 의해 변경된 변수 값을 출력

✓ 출력 결과

```
a = 1.0, b = 2.0, c = 3.0, d = 4.0
++a*b-c-- = 1.000000
a = 2.0, b = 2.0, c = 2.0, d = 4.0
```

Operator precedence and associativity	
Operator	Associativity
() ++ *(postfix)* -- *(postfix)*	left to right
+ (*unary*) - (*unary*) ++ (*prefix*) -- (*prefix*)	right to left
* / %	left to right
+ -	left to right
= += -= *= /= etc.	right to left

실습2

정수 2개를 입력 받아 +, *, % 연산 결과를 보여주는 코드를 작성

- 결과값을 10진수 뿐만 아니라 8진수, 16진수로도 보여줌
- 결과값을 "*" 의 개수로도 표현

```
ppmccl30@martini:~$ ./ex3-2
2개의 정수를 입력하세요 : 3 5
3 + 5 = 8 (8진수 : 10) (16진수 : 8)
********
3 * 5 = 15 (8진수 : 17) (16진수 : f)
***************
3 % 5 = 3 (8진수 : 3) (16진수 : 3)
***
ppmccl30@martini:~$
```

과제

과제1

양의 정수를 입력 받아 1부터 해당 값 사이의 소수(prime number) 개수 및 해당 소수를 출력하는 코드를 작성한다.

```
ppmccl30@martini:~$ ./hw3-1
정수를 입력하세요 : 1000

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101
103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193
197 199 211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 293
307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409
419 421 431 433 439 443 449 457 461 463 467 479 487 491 499 503 509 521
523 541 547 557 563 569 571 577 587 593 599 601 607 613 617 619 631 641
643 647 653 659 661 673 677 683 691 701 709 719 727 733 739 743 751 757
761 769 773 787 797 809 811 821 823 827 829 839 853 857 859 863 877 881
883 887 907 911 919 929 937 941 947 953 967 971 977 983 991 997

1부터 1000 사이에는 총 168 개의 소수가 존재합니다.
ppmccl30@martini:~$
```

과제2

양의 정수(홀수)를 입력 받아 해당 값 높이의 마름모를 그려 주는 코드를 작성한다.

```
ppmccl30@martini:~$ ./hw3-2
숫자를 입력하세요 : 9
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
ppmccl30@martini:~$
```