

Practice 4

컴퓨터프로그래밍1

4월 6일

1. 지난실습 리뷰

2. C프로그래밍 실습

(+ 실습 파일 제출 방법)

실습 review

실습 4

- 사용자에게 4개의 정수를 입력 받고 평균을 구하는 프로그램을 작성하시오
 - 입력과 출력은 main function에서 수행
 - main function 이외에 **평균을 구하는 2개의 함수를 정의**하여 사용
 1. 실수 평균을 구하는 함수 (4 정수를 전달 받아 평균을 **float**으로 return)
 2. 정수 평균을 구하는 함수 (4 정수를 전달 받아 평균을 **int**으로 return)
 - 실수 평균은 **소수점 아래 6자리** 까지 출력
- 출력 예시

```
input the first number: 23
input the second number: 12
input the third number: 45
input the fourth number: 23

Average(float) is 25.750000
Average(int) is 25
```

```
input the first number: 107
input the second number: 83
input the third number: 99
input the fourth number: 121

Average(float) is 102.500000
Average(int) is 102
```

실습 4

- 정답 예시

```
1 #include <stdio.h>
2
3 float mean(int, int, int, int);
4 int int_mean(int, int, int, int);
5
6 int main(void){
7     int n1, n2, n3, n4;
8
9     printf("input the first number: ");
10    scanf("%d", &n1);
11    printf("input the second number: ");
12    scanf("%d", &n2);
13    printf("input the third number: ");
14    scanf("%d", &n3);
15    printf("input the fourth number: ");
16    scanf("%d", &n4);
17
18    printf("\nAverage(float) is %.6f", mean(n1, n2, n3, n4));
19    printf("\nAverage(int) is %d\n", int_mean(n1, n2, n3, n4));
20
21    return 0;
22 }
23
24 float mean(int n1, int n2, int n3, int n4){
25     int sum = 0;
26     sum = n1 + n2 + n3 + n4;
27
28     return (float)sum / 4;
29 }
30
31 int int_mean(int n1, int n2, int n3, int n4){
32     int sum = 0;
33     sum = n1 + n2 + n3 + n4;
34
35     return sum / 4;
36 }
```

실습 5

- 실수 2개를 입력 받아 user defined function을 이용해 두 값을 교환하여 출력하는 프로그램을 작성하시오
 - main function에서는 입력과 출력, 그리고 함수 호출만을 담당
 - user defined function에서는 **pass by reference** 방식으로 두 변수를 전달 받고, 두 변수의 값을 바꿈
 - 함수의 형태는 void swap (?x, ?y); 형태
 - 전역변수는 사용 금지
- 출력 예시

```
input two number
x: 1.3
y: 2.8

Before: x [1.300000] y [2.800000]
After : x [2.800000] y [1.300000]
```

```
input two number
x: 11.5
y: 92.149

Before: x [11.500000] y [92.149002]
After : x [92.149002] y [11.500000]
```

실습 5

- 정답 예시

```
1 #include <stdio.h>
2
3 void swap(float*, float*);
4
5 int main(void){
6     float x, y;
7
8     printf("input two number\n");
9     printf("x: ");
10    scanf("%f", &x);
11    printf("y: ");
12    scanf("%f", &y);
13
14    printf("\nBefore: x [%f] y [%f]\n", x, y);
15    swap(&x, &y);
16    printf("After : x [%f] y [%f]\n", x, y);
17
18    return 0;
19 }
20
21 void swap(float* x, float* y){
22     float tmp;
23     tmp = *x;
24     *x = *y;
25     *y = tmp;
26 }
```

C프로그래밍 실습

실습 6

- 사용자가 입력한 년도가 윤년인지 확인하고 결과를 출력해주는 프로그램을 작성하시오.
 - 입력한 년도가 윤년이 아니라면 그 이전 년도 중 가장 가까운 윤년을 출력
 - 윤년 확인방법:
 - 년도가 4로 나누어 떨어지면서 100으로는 나누어 떨어지지 않거나
 - 혹은, 400으로 나누어 떨어지는 해

- 출력 예시

```
gr120200190@cspiro:~$ ./runs/6
Enter the year to be tested: 2013
It is not a leap year.
2012 is a leap year.
gr120200190@cspiro:~$ ./runs/6
Enter the year to be tested: 2012
It is a leap year.
gr120200190@cspiro:~$ ./runs/6
Enter the year to be tested: 2001
It is not a leap year.
2000 is a leap year.
gr120200190@cspiro:~$ ./runs/6
Enter the year to be tested: 2101
It is not a leap year.
2096 is a leap year.
```

실습 7

- 하나의 실수를 입력 받고 반올림을 할 때 소수점 아래를 버려야 하는지, 올려야 하는지 확인하는 프로그램을 작성하시오.
- 모든 내용을 main function에 구현
- 소수점 아래를 버려야 한다면 0, 올려야 한다면 1을 출력
- printf 문 안에서 필요한 연산과, 다음 4개의 관계연산자 각각만을 사용하여 결과를 출력
>, >=, ==, != (총 4번 출력)
- Main function의 형태

```
printf ("Use \">>\": %d\n", , > ) ;
printf ("Use \">>=\": %d\n", , >= ) ;
printf ("Use \"==\": %d\n", , == ) ;
printf ("Use \"!=\": %d\n", , != ) ;
```

hint) casting을 잘 활용하여야 함

출력 예시

```
gr120200190@cspiro:~$ ./runs/7
Input: 0.5
Use ">": 1
Use ">=": 1
Use "==" : 1
Use "!=": 1
gr120200190@cspiro:~$ ./runs/7
Input: 0.1
Use ">": 0
Use ">=": 0
Use "==" : 0
Use "!=": 0
```

```
gr120200190@cspiro:~$ ./runs/7
Input: 3.4
Use ">": 0
Use ">=": 0
Use "==" : 0
Use "!=": 0
gr120200190@cspiro:~$ ./runs/7
Input: 12.999
Use ">": 1
Use ">=": 1
Use "==" : 1
Use "!=": 1
```

실습 8

- 두 개의 operand (실수 a, b)와 하나의 operator를 입력 받아 해당 operator에 맞는 연산을 해주는 계산기 프로그램을 작성하시오.
 - 출력 예시
- operand와 operator는 띄어쓰기로 구분
- +, -, *, / 연산만 구현
- 계산 결과는 소수 2번째 자리까지 출력
- operand는 항상 소수(2번째 자리까지)라 가정
- 잘못된 operator에 대한 예외 처리 필요
- 0으로 나눗셈을 하는 경우 error message를 출력
- 사칙 연산을 구분해주는 조건문으로 반드시 switch문을 이용

```
gr120200190@cspiro:~$ ./runs/8
Operator (+, -, *, /)
Please enter the formula. (ex: 12.3 + 5.5)
12.3 + 5.5
12.30 + 5.50 = 17.80
gr120200190@cspiro:~$ ./runs/8
Operator (+, -, *, /)
Please enter the formula. (ex: 12.3 + 5.5)
22.2 - 32.8
22.20 - 32.80 = -10.60
gr120200190@cspiro:~$ ./runs/8
Operator (+, -, *, /)
Please enter the formula. (ex: 12.3 + 5.5)
4.2 * 2.0
4.20 * 2.00 = 8.40
gr120200190@cspiro:~$ ./runs/8
Operator (+, -, *, /)
Please enter the formula. (ex: 12.3 + 5.5)
10.4 / 4.0
10.40 / 4.00 = 2.60
gr120200190@cspiro:~$ ./runs/8
Operator (+, -, *, /)
Please enter the formula. (ex: 12.3 + 5.5)
3.2 / 0
Division by zero
gr120200190@cspiro:~$ ./runs/8
Operator (+, -, *, /)
Please enter the formula. (ex: 12.3 + 5.5)
3.4 @ 1.2
Invalid operator: @
```

실습 제출 방법 - 사이버캠퍼스

- 각 실습의 파일명은 {학번}_{실습번호}.c로 저장
e.g.) 실습1의 파일명은 20211234_1.c
- 작성한 c파일 코드들 (.c 파일들)을 사이버캠퍼스 과제란에 업로드
- 제출기한 : 사이버캠퍼스에 명시된 기한까지
- **지각제출은 불가능**
- **파일제목 등 제출 양식이 틀리면 오답처리**