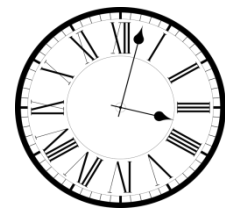


### 과제 #3 (총 10점)

#### [Q1] 시침과 분침 사이 각도 구하기 (4점)

##### 1. 설명

- 1) 아날로그 시계가 있다. 특정 시각에 시침과 분침이 이루는 사이의 각도를 구하는 프로그램을 완성하라. 두 개의 정수, 시와 분의 값을 입력 받아, 분침이 앞서 있거나 같은 위치이면 양수로, 아니면 음수로 각도를 표시하여 출력한다. 입력하는 시의 범위는 0에서 11까지, 분의 범위는 0에서 59까지의 정수이다.
- 2) 위와 같이 입력된 시간대에서 시침과 분침이 겹치지 않고 일직선이 되는 시각을 출력하라.



##### 2. 결과

※ 참고로 회색 배경이 칠해진 부분은 사용자가 입력하는 부분이다.

<입/출력 예시 1>

Enter two integer numbers (hour[0~11] & minute[0~59]): 3 15

1) Angle = -7.50° @ 03:15

2) Straight Line @ 03:49:05.45

Enter two integer numbers (hour[0~11] & minute[0~59]):

<입/출력 예시 2>

Enter two integer numbers (hour[0~11] & minute[0~59]): 12 30

\*\* Invalid Input! Hour should be an integer. 0 to 11.

Enter two integer numbers (hour[0~11] & minute[0~59]): 5 69

\*\* Invalid Input! Minute should be an integer. 0 to 59

Enter two integer numbers (hour[0~11] & minute[0~59]):

##### 3. 주의

- ① 프로그램 시작에 친절하게 "Enter two integer numbers (hour[0~11] & minute[0~59]):"를, 출력해 반복 입력할 수 있도록 하고, 두 숫자 모두 '0'이 입력되면 프로그램을 종료한다.
- ② 잘못된 입력에 대하여 친절한 문구를 출력하고, 상기 항과 같이 반복하여 입력을 받는다.
- ③ 입력 처리 후 결과 출력은 <입/출력 예시 1> ~ <입/출력 예시 2>를 따른다.
- ④ 각도와 초 단위는 소수점 아래 두 자리 실수로 출력한다. (예, 03:49:05.45)
- ⑤ 프로그램 종료 시, 친절한 문구를 출력해줄 것.

[Q2] 고정 소수점 실수의 이진수 변환 (3 점)

1. 설명

고정소수점으로 표기된 10진 실수를 고정소수점 이진수로 나타낸다.  $(\pm)0.D1D2D3D4$  형태의 소수점 이하 4자리까지의, 절대값이 1보다 작은, 고정소수점 실수를 입력 받아, 이진수  $(\pm)0.B1B2...B8$  형태의 소수점 이하 8자리까지 출력한다.

2. 결과

※ 참고로 회색 배경이 칠해진 부분은 사용자가 입력하는 부분이다.

<입/출력 예시 1>

Enter a fixed-point decimal number (0 for exit): 0,8125  
 \*\* Binary equivalent: +0.1101 0000

<입/출력 예시 2>

Enter a fixed-point decimal number (0 for exit): -0.8125  
 \*\* Binary equivalent: -0.1101 0000

<입/출력 예시 3>

Enter a fixed-point decimal number (0 for exit): 0.1  
 \*\* Binary equivalent: +0.0001 1001

<입/출력 예시 4>

Enter a fixed-point decimal number (0 for exit): 1.25  
 \*\* Invalid input! Its absolute value should be less than one.

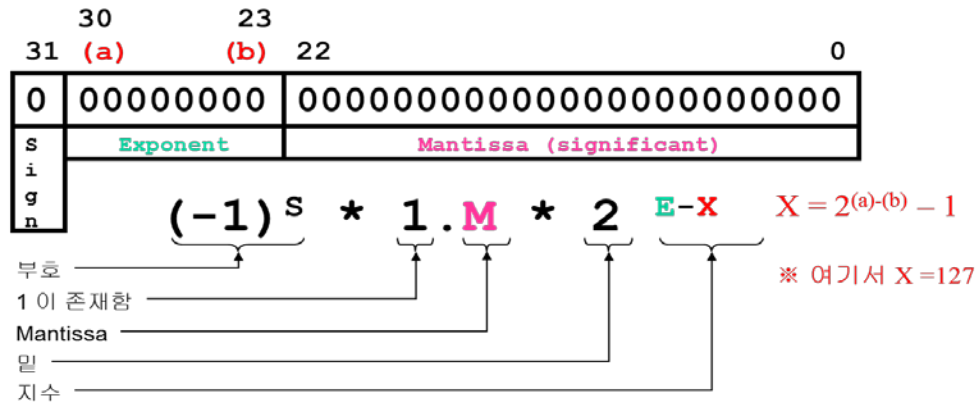
3. 주의

- ① 프로그램 시작에 친절하게 "Enter a fixed-point decimal number (0 for exit):"를 출력해 반복 입력할 수 있도록 하고, 숫자 '0'이 입력되면 프로그램을 종료한다.
- ② 잘못된 입력에 대하여 친절한 문구를 출력하고, 상기 항과 같이 반복하여 입력을 받는다.
- ③ 입력 처리 후 결과 출력은 <입/출력 예시 1> ~ <입/출력 예시 4>를 따른다.
- ④ 소수점 이하의 이진수는 4개의 수를 묶어, 사이에 공란을 두어 출력한다.
- ⑤ 프로그램 종료 시, 친절한 문구를 출력해줄 것.

[Q3] Float Type 실수 이진수로 출력 (3점)

1. 설명

float type의 내부 저장 32-비트 열에 따른 값 설정 방법 (IEEE 754)은 아래 그림과 같다. (컴프 강의자료 참조)



예로, 실수 **123.8125**는 아래와 같이 비트 열로 내부에 저장된다.

- ① 부호 (양수):  $s=0$  (1 bit)
- ② 이진수 변환:  $123.8125 \rightarrow 1111011.1101$
- ③ 소수점 이동:  $1.1110111101$  (좌로 6칸 이동)
  - $M = 1110\ 1111\ 0100\ 0000\ 0000\ 000$  (23 bits)
  - $E - X = 6$  where  $X = 127 \rightarrow E = 133 \rightarrow 1000\ 0101$  (8 bits)
- ④  $123.8125 \rightarrow 0\ 1000\ 0101\ 1110\ 1111\ 0100\ 0000\ 0000\ 000$

실수 형의 숫자를 입력 받아, 위의 내부 저장 형태의 비트 열을 출력하는 프로그램을 구현하라. 입력 된 수가 0이면, float-type 상수 0의 해당 비트 열과 더불어 float-type 최대 상수의 비트 열도 출력하고 프로그램을 종료한다.

2. 결과

※ 참고로 회색 배경이 칠해진 부분은 사용자가 입력하는 부분이다.

<입/출력 예시 1>

```
Enter a float-type real number (0 to exit): 0.1
** Binary representation = 0011 1101 1100 1100 1100 1100 1101
Sign           = 0
Exponent       = 0111 1011
Mantissa       = 1001 1001 1001 1001 1001 101
Enter a float-type real number : Enter a float-type real number (0 for exit):
```

<입/출력 예시 2>

```
Enter a float-type real number (0 for exit): -123..8125
** Binary representation = 1100 0010 1111 0111 1010 0000 0000 0000
Sign          = 1
Exponent      = 1000 0101
Mantissa      = 1110 1111 0100 0000 0000 000
Enter a float-type real number (0 for exit):
```

<입/출력 예시 3>

```
Enter a float-type real number (0 for exit): 0
** Binary representation = 0000 0000 0000 0000 0000 0000 0000 0000
Sign          = 0
Exponent      = 0000 0000
Mantissa      = 0000 0000 0000 0000 0000 000
NAX of float number = 340282346638528859811704183484516925440.000000
** Binary representation = 0111 1111 0111 1111 1111 1111 1111 1111
Sign          = 0
Exponent      = 1111 1110
Mantissa      = 1111 1111 1111 1111 1111 111
Bye!!!
```

### 3. 주의

- ① 프로그램 시작에 친절하게 "Enter a float-type real number (0 to quit):"를 출력해 반복 입력할 수 있도록 하고, 숫자 '0'이 입력되면 프로그램을 종료한다.
- ② 입력 처리 후 결과 출력은 <입/출력 예시 1> ~ <입/출력 예시 3>를 따른다.
- ③ 이진수는 4개의 숫자를 묶어, 사이에 공란을 두어 출력한다.
- ④ 프로그램 종료 시, 친절한 문구를 출력해줄 것.
- ⑤ Float type의 최대 상수는 표준 라이브러리 <float.h> 내 FLT\_MAX 로 정의되어 있다.