

2015년 2학기

프로그래밍과 문제해결

Assignment #2

담당교수: 윤은영

학번: 20150000

학과: 컴퓨터공학과

이름: 홍길동

POVIS ID: xxxxxxxx

명예서약(Honor code)

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

Problem: 온도 변환 프로그램

1. 문제의 개요

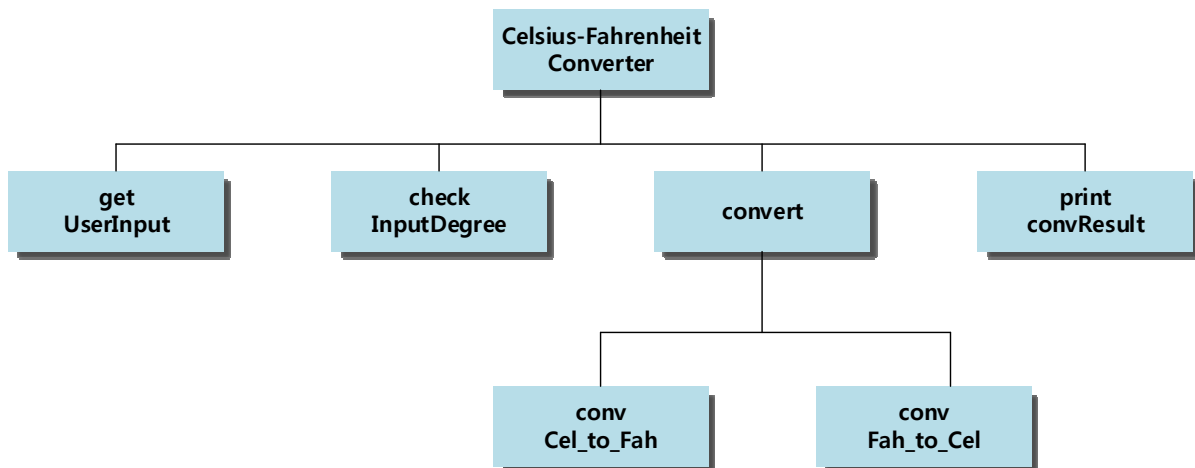
본 프로그램을 간략히 설명하면 다음과 같다.

- 사용자가 단위 변환을 원하는 온도 값을 실수 형태로 입력 받는다.
- 입력한 온도 값의 단위를 문자 형태로 입력 받는다.
- 사용자로부터 입력 받은 두 종류의 값(실수 및 문자)을 활용하여 온도 변환을 수행하여 출력한다.
- 이 때 사용되는 섭씨(Celsius) 및 화씨(Fahrenheit) 간 변환 공식은 아래와 같다.

$$C = (F - 32.0) / 1.8$$

$$F = C * 1.8 + 32.0$$

이 때 사용되는 구상 가능한 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.



- 입력부: 사용자가 입력하는 온도 값과 그에 해당하는 단위를 입력 받고 저장한다.
- 처리부: 입력 예외 처리를 통해 사용자가 입력한 적절한 온도 단위를 반환한다.

반환된 단위 값을 바탕으로 온도 값을 변환하여 결과를 반환한다.

- 출력부: 변환이 완료된 온도 값을 바뀐 단위와 함께 출력한다.

2. 알고리즘

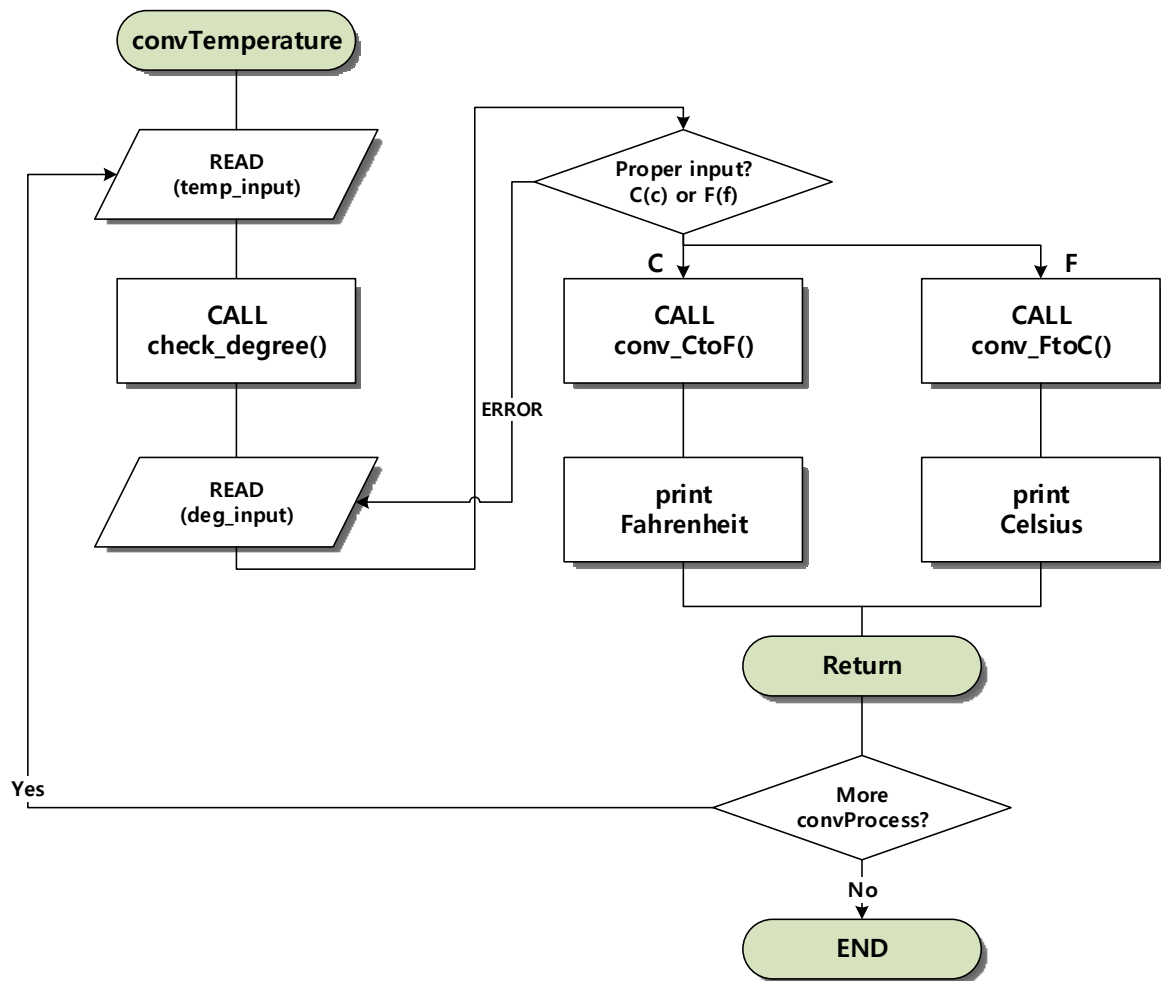
본 프로그램 작성을 위한 알고리즘을 Pseudo 코드 형태로 나타내면 다음과 같다.

Pseudo-algorithm for Convert Temperature

// 프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

- 1 include essential headers <stdio.h>
 - 2 prompt user and read value of temperature in float type
 - 3 prompt user and read degree of temperature in character type
 - 4 check input degree
 - 5 if (deg != 'c' || 'C' || 'f' || 'F')
 repeat until proper input
 else return input degree
 - 6 call convert function
 - 7 if (input degree == 'c' || 'C')
 call conv_CtoF & return convResult
 else
 call conv_FtoC & return convResult
 - 8 print convResult
 - 9 if user want more convert
 go to step 2. until EXIT
-

위의 의사 알고리즘을 Flowchart를 통해 표현하면 아래와 같다.



3. 프로그램 구조 및 설명

a) 변환을 원하는 온도 값 입력

- 프로그램을 실행하면 변환에 사용될 온도 값에 대한 입력을 요청한다. 사용자가 온도 값을 입력하면 scanf를 통해 실수형(float) 변수 temp에 저장된다.

b) 온도의 단위 입력

- 온도의 입력을 완료하게 되면 프로그램은 입력한 온도의 단위를 확인하는 함수 check_degree()함수를 호출한다.
- 해당 함수에서는 정확한 온도 단위의 입력이 이루어 질 때까지 단위 입력 의사를 do-while 구문을 통해 반복하게 된다.
- 적절한 온도 단위의 입력이 이루어지면, 해당 값을 반환하여 준다.

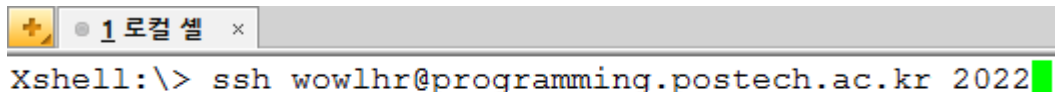
c) 온도 값의 변환

- check() 함수를 통한 단위 입력 결과에 따라 적절한 변환 함수를 호출하게 된다. 단위 입력이 C 또는 c인 경우, convert_CtoF(temp) 함수를 호출하고 그 반대의 경우 convert_FtoC(temp) 함수를 호출한다.
- 각 변환 함수 내에는 입력 변수에 따라 온도 값을 변환하는 계산을 수행하며, 그 결과를 즉시 출력해준다.
- 이 때, 소수점 아래 둘째 자리까지 결과를 나타내기 위해 "%.2f" 포맷 형식을 사용한다.

d) 추가 변환 작업 실행

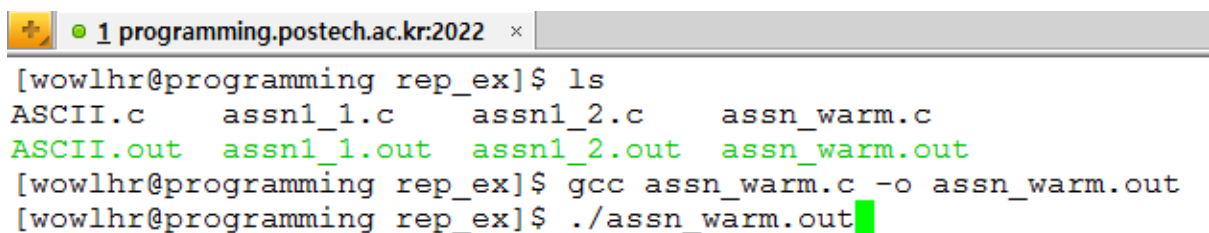
- 사용자에게 추가적인 변환 작업에 대한 의사문을 출력하게 되고, 입력에 따라 a) 과정으로 돌아갈 지 또는 프로그램을 종료할 지 결정하게 된다.
- 의사 변수를 정수형으로 선언하고, while 문을 통해 1의 입력이 있으면 프로그램을 반복적으로 수행하도록 작성하였다.

4. 프로그램 실행방법 및 예제



```
Xshell:\> ssh wowlhr@programming.postech.ac.kr 2022
```

- Xshell 환경에서 위 명령어를 통해 비밀번호를 입력하고, 본인에게 할당된 리눅스 서버에 접속한다. 이 후 본 프로그램이 저장되어 있는 폴더로 이동한다.



```
[wowlhr@programming rep_ex]$ ls
ASCII.c      assn1_1.c    assn1_2.c    assn_warm.c
ASCII.out    assn1_1.out  assn1_2.out  assn_warm.out
[wowlhr@programming rep_ex]$ gcc assn_warm.c -o assn_warm.out
[wowlhr@programming rep_ex]$ ./assn_warm.out
```

- 이동한 폴더에서 작성한 프로그램의 소스코드가 있는지 확인을 한 후 컴파일을 수행한다. 이 후, 컴파일이 완료된 파일을 실행시킨다.

```

1 programming.postech.ac.kr:2022 x
[wowlhr@programming rep_ex]$ ls
ASCII.c      assn1_1.c    assn1_2.c    assn_warm.c
ASCII.out    assn1_1.out  assn1_2.out  assn_warm.out
[wowlhr@programming rep_ex]$ gcc assn_warm.c -o assn_warm.out
[wowlhr@programming rep_ex]$ ./assn_warm.out
Enter the degree you want to convert:

```

- 프로그램이 정상적으로 동작하여 변환을 요구하는 온도 값 입력을 요청한다.

```

1 programming.postech.ac.kr:2022 x
[wowlhr@programming rep_ex]$ ls
ASCII.c      assn1_1.c    assn1_2.c    assn_warm.c
ASCII.out    assn1_1.out  assn1_2.out  assn_warm.out
[wowlhr@programming rep_ex]$ gcc assn_warm.c -o assn_warm.out
[wowlhr@programming rep_ex]$ ./assn_warm.out
Enter the degree you want to convert:100.0
Is this a Fahrenheit or Celsius? [F/C]

```

- 성공적으로 입력을 마치게 되면 프로그램은 온도 값에 해당하는 단위를 입력할 것을 요청한다. 이 때, 적절하지 않은 단위를 입력하게 되면 예외 처리를 통해 의사문이 반복되는 것을 확인할 수 있다.

```

1 programming.postech.ac.kr:2022 x
[wowlhr@programming rep_ex]$ ls
ASCII.c      assn1_1.c    assn1_2.c    assn_warm.c
ASCII.out    assn1_1.out  assn1_2.out  assn_warm.out
[wowlhr@programming rep_ex]$ gcc assn_warm.c -o assn_warm.out
[wowlhr@programming rep_ex]$ ./assn_warm.out
Enter the degree you want to convert:100.0
Is this a Fahrenheit or Celsius? [F/C]a
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius? [F/C]b
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius? [F/C]c
    100.00 C = 212.00 F
//////////
    1.continue      2.EXIT:

```

- 적절한 온도 단위를 입력하는 경우, 반한 함수를 통해 온도 값과 단위가 정확하게 변환되는 것을 확인할 수 있다.
- 프로그램 반복 수행에 대해 추가 작업을 원하는 경우 아래와 같이 온도 입력 부분부터 동작하는 것을 볼 수 있다.

```

1 programming.postech.ac.kr:2022 x
[wowlhr@programming rep_ex]$ ls
ASCII.c      assn1_1.c    assn1_2.c    assn_warm.c
ASCII.out    assn1_1.out  assn1_2.out  assn_warm.out
[wowlhr@programming rep_ex]$ gcc assn_warm.c -o assn_warm.out
[wowlhr@programming rep_ex]$ ./assn_warm.out
Enter the degree you want to convert:100.0
Is this a Fahrenheit or Celsius? [F/C]a
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius? [F/C]b
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius? [F/C]c
      100.00 C = 212.00 F
////////////////////////////////////
      1.continue      2.EXIT: 1
Enter the degree you want to convert:

```

```

1 programming.postech.ac.kr:2022 x
[wowlhr@programming rep_ex]$ ls
ASCII.c      assn1_1.c    assn1_2.c    assn_warm.c
ASCII.out    assn1_1.out  assn1_2.out  assn_warm.out
[wowlhr@programming rep_ex]$ gcc assn_warm.c -o assn_warm.out
[wowlhr@programming rep_ex]$ ./assn_warm.out
Enter the degree you want to convert:100.0
Is this a Fahrenheit or Celsius? [F/C]a
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius? [F/C]b
Incorrect temperature scale!!!
Is this a Fahrenheit or Celsius? [F/C]c
      100.00 C = 212.00 F
////////////////////////////////////
      1.continue      2.EXIT: 1
Enter the degree you want to convert:50.23
Is this a Fahrenheit or Celsius? [F/C]F
      50.23 F = 10.13 C
////////////////////////////////////
      1.continue      2.EXIT: 2
[wowlhr@programming rep_ex]$

```

- 추가적인 변환 작업을 성공적으로 수행 후, 프로그램을 종료하게 된다.

5. 토론

- 변환을 원하는 온도가 입력된 후, 온도의 단위를 확인하는 함수를 불러오게 되는데, 이때 온도 입력에 사용되었던 엔터(Wn)가 입력 버퍼에 여전히 남아있어서 "Incorrect temperature scale!!!"이라는 경고문구가 한 차례 나온 후 동작하게 된다.

- 이러한 문제를 해결하기 위해, `fflush(stdin)`을 사용하여 입력 버퍼를 비워주는 방법보다 간단하게 온도의 단위를 입력 받을 때 "%c%c" 처리를 함으로써 'WnW'가 정당한 입력으로 처리되게 하는 방법이 있다.
- 위 방식과는 반대로 온도의 단위 입력 전 " %c"와 같이 공백을 임의로 추가해줌으로써 버퍼에 남아있는 'Wn' 값을 무시하는 방법도 손쉽게 적용할 수 있다.

6. 결론

- 본 과제에서는 프로그램 작성시에 필요한 함수들을 작성하고 사용하는 방법을 주로 학습하게 되었다. 특히, 함수의 선언에 따른 반환형 및 반환되는 값이 중요함을 알게 되었다. 뿐만 아니라, 원하는 값이 입력될 때까지 입력부를 반복 출력하는 기본적인 예외 처리에 대해 알 수 있었다.

7. 개선방향

- 본 과제 수행을 위해 온도 변환 함수를 두 개 (CtoF / FtoC)로 작성하였는데, 이를 다중 함수 입력(온도 값과 단위)을 활용하여 하나의 함수로 구현함으로써 더욱 간결한 프로그램이 작성될 것으로 기대된다.