

## Day 8 Documentation

Md. Mahfuj Hasan Shohug

BDCOM0019

.....

### 1. Exercise 1.15:

Problem: Rewrite the temperature conversion program of Section 1.2 to use a function For conversion.

Solution: temperature conversion program of Section 1.2 was given bellow:

```
#include <stdio.h>
/* print Fahrenheit-Celsius table
for fahr = 0, 20, ..., 300; floating-point version */
main()
{
    float fahr, celsius;
    float lower, upper, step;
    lower = 0; /* lower limit of temperatuire scale */
    upper = 300; /* upper limit */
    step = 20; /* step size */
    fahr = lower;
    while (fahr <= upper) {
        celsius = (5.0/9.0) * (fahr-32.0);
        printf("%3.0f %6.1f\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

For this temperature conversation in my program I am using one function called "temperature\_conversion" which one can convert both calculation Celsius to Fahrenheit and Fahrenheit to Celsius by calling function from the main function. Using a specified temperature range and step size, this application intends to display temperature conversion tables for Celsius to Fahrenheit and Fahrenheit to Celsius. The output will be a structured table with temperature values and their converted equivalents.

The function temperature\_conversion calculate this two equation:

1 – For Celsius to Fahrenheit ->  $(temp * 9 / 5) + 32$

2 – For Fahrenheit to Celsius ->  $(temp - 32) * 5 / 9$

Source code is here:

```

1  #include <stdio.h>
2  #include <stdbool.h>
3  /**
4   * Function Name: main, temperature_conversion
5   * Inputs       : 1. argc -- The number of parameters provided to the main function**
6   *              : 2. argv -- The pointer to the input string array of parameters **
7   * Variable     : temp -- variable for tempture
8   *              : low_temp -- Lowest value of tempture 0
9   *              : high_temp -- highest value of tempture 300
10  *              : step_num -- step is decrease by 20
11  * Return       : = 0 -- Success
12  *              : < 0 -- Failed
13  * Note        : Temperature conversion program using a function for conversion.
14  */
15  /**tempture convater function both celsius to fahrenheit and fahrenheit to celsius
16  ** with declaring parameter**/
17  void temperature_conversion (int high_temp, int low_temp, int step_num, bool is_celsiToFahr)
18  {
19      float temp = low_temp; // declare the float variable for tempture
20
21      /*using the ternary operator conditionally choose the appropriate string*/
22      printf("%s\t%s\n", is_celsiToFahr ? "Celsius" : "Fahrenheit", is_celsiToFahr ? "Fahrenheit" : "Celsius");
23      printf("-----\t-----\n");
24      while(temp <= high_temp)
25      {
26          if(is_celsiToFahr)
27          {
28              printf("%.2f\t%.2f\n", temp, (temp * 9 / 5) + 32); // celsius to fahrenheit formula
29          }
30          else
31          {
32              printf("%.2f\t%.2f\n", temp, (temp - 32) * 5 / 9); // fahrenheit to celsius formula
33          }
34          temp += step_num;
35      }
36  }
37
38  //Main Function
39  int main(int argc, char *argv[])
40  {
41      temperature_conversion (300, 0, 20, true); // for bool value true calculate Celsius to Fahrenheit
42      printf("\n");
43      temperature_conversion (300, 0, 20, false); //for bool value false calculate Fahrenheit to Celsius
44      return 0;
45  }

```

Resources | Compile Log | Debug | Find Results | Close

Compilation results...

```

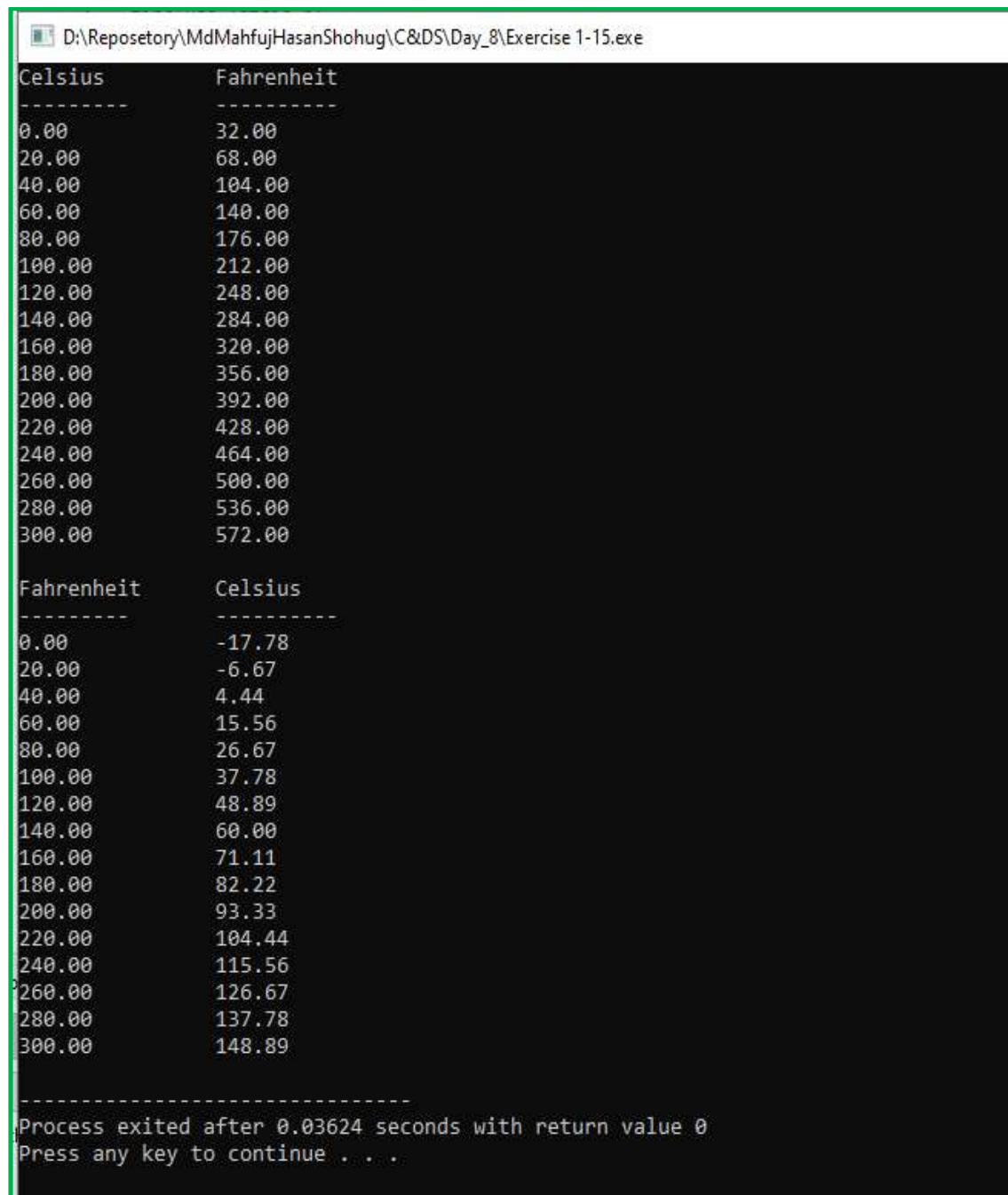
-----
- Errors: 0
- Warnings: 0
- Output Filename: D:\Reposetory\MdMahfujHasanShohug\C&DS\Day_8\Exercise 1-15.exe
- Output Size: 130.3125 KiB
- Compilation Time: 0.16s

```

Here is this code the temperature\_conversion function's internal structure low\_temp is used as the initial value of the temp variable. Based on the is\_celsiToFahr argument, the table headers are produced, showing the relevant column headers. The supplied step\_num is used in a while loop to iterate through the temperature range from low\_temp to high\_temp. Based on the is\_celsiToFahr parameter, the temperature value and its converted value are printed in the loop in a formatted manner. Each repetition increases the temp value by step\_num.

In the main function the temperature\_conversion function called two times, once with is\_celsiToFahr set to true (for converting from Celsius to Fahrenheit) and once with is\_celsiToFahr set to false (for converting from Fahrenheit to Celsius). The temperature range (300 to 0), step size (20), and conversion direction are all provided by the function calls. Between the two sets of temperature conversion tables, a newline character is printed. The main function returns 0, indicating that the program has run successfully.

Here is the sample of the output on this code.



```
D:\Reposetory\MdMahfujHasanShohug\C&DS\Day_8\Exercise 1-15.exe
Celsius          Fahrenheit
-----
0.00             32.00
20.00            68.00
40.00            104.00
60.00            140.00
80.00            176.00
100.00           212.00
120.00           248.00
140.00           284.00
160.00           320.00
180.00           356.00
200.00           392.00
220.00           428.00
240.00           464.00
260.00           500.00
280.00           536.00
300.00           572.00

Fahrenheit        Celsius
-----
0.00              -17.78
20.00             -6.67
40.00              4.44
60.00             15.56
80.00             26.67
100.00            37.78
120.00            48.89
140.00            60.00
160.00            71.11
180.00            82.22
200.00            93.33
220.00           104.44
240.00           115.56
260.00           126.67
280.00           137.78
300.00           148.89

-----
Process exited after 0.03624 seconds with return value 0
Press any key to continue . . .
```