```c
/************************************************/
#include <stdio.h>
/************************************************/

/* Author: Daniel Gonzalez P#4926400 */
/* Assignment 1: Table temperature (Fahrenheit and Celsius) */

/* program will accept and validate input values
 * to be one character in length in the range from 1 to 9 */
/* compilation: gcc TConverter.c -o myprogram.out */

/*   execute with ./myprogram.out */
/*   To exit execution press [ctrl + Z] */

/*********************************************
    I Daniel Gonzalez #4926400 hereby certify that this collective work is
    my own and none of it is the work of any other person or entity.
**********************************************/









/************************************************/
/* program defines */

#define LOWER -20.0                             /* Lower limit of table */
#define UPPER 280.0                             /* Upper limit */
#define NOERRORS 0                              /* For clean return  */
#define TEMPERATURE_FORMULA_VALUE_5 5.0         /* Constant formula value 5.0 */
#define TEMPERATURE_FORMULA_VALUE_9 9.0         /* Constant formula value 9.0*/
#define TEMPERATURE_FORMULA_VALUE_32 32.0       /* Constant formula value 32.0*/
#define INPUT_LOWEST 1                          /* Minimum step value*/
#define INPUT_HIGHEST 9                         /* Maximum step value*/
#define EOL '\n'                                /* End of line character*/
#define ZERO_CHAR '0'                           /* Character 0 */
#define SPACE_CHAR ' '                          /* Space character*/









/************************************************/
/*   Function prototypes:
*    Funtion declarations are listed as below.
*/
float convertToCelsius(float fahrenheit);
float convertToFahrenheit(float celsius);
void tableDisplay(float, float, int);
int validateInput();
/************************************************/
```

```c


/**
 * [main program function]
 * @return [NOERRORS when program finishes ok]
 */
int main()
{
    /* Declare and initialize variables  */

    float fahrenheit = LOWER;
    float celsius = LOWER;
    int step = 0;


    /* Input values */
    printf( "Enter a value from %d to %d for the table temperature steps:\n",
            INPUT_LOWEST,
            INPUT_HIGHEST
            );
    step = validateInput();


    /* Main process */
    tableDisplay(fahrenheit, celsius, step);


    return NOERRORS;
}
```

```c

/**
 * [convertToCelsius description] converts Fahrenheit to Celsius
 * @param  Fahrenheit [float Fahrenheit temperature value to convert to Celsius]
 * @return            [float Celsius converted value]
 * Conversion formula used f(x) = 5(x - 32)/9
 */
float convertToCelsius(float fahrenheit){

    return
    TEMPERATURE_FORMULA_VALUE_5
    * (fahrenheit - TEMPERATURE_FORMULA_VALUE_32)
    / TEMPERATURE_FORMULA_VALUE_9;
}



/**
 * [convertToFahrenheit Converts Celsius to Fahrenheit]
 * @param  celsius [float value to be converted]
 * @return         [float Converted value to Fahrenheit]
 * Conversion formula used f(x) = 9x/5 + 32
 */
float convertToFahrenheit(float celsius){

    return
    TEMPERATURE_FORMULA_VALUE_9
    * celsius
    /TEMPERATURE_FORMULA_VALUE_5
    + TEMPERATURE_FORMULA_VALUE_32;
}



/**
 * [tableDisplay prints a conversion temperature table with the specified
 * increment]
 * @param fahrenheit [fahrenheit initial Value]
 * @param celsius    [celsius initial value]
 * @param step       [Step increment for intervals of table]
 */
void tableDisplay(float fahrenheit, float celsius, int step){

    printf(
        "---------------------------------------"
        "---------------------------------------\n"
        "\tFahrenheit\tCelsius\t\tCelsius\t\tFahrenheit\n"
        "---------------------------------------"
        "---------------------------------------\n"
        );

    while (fahrenheit <= UPPER)
    {
        printf(
            "\t%6.1f\t\t %6.1f\t\t %6.1f\t\t %6.1f\n",
            fahrenheit,
            convertToCelsius(fahrenheit),
            celsius,
            convertToFahrenheit(celsius)
            );

        celsius = fahrenheit += step;
    }

}
```

```
210
211 /**
212  * [validateInput validates user input to be in range]
213  * @return [int validated input]
214  */
215 int validateInput(){
216
217     // getchar() - ZERO_CHAR : converts input char to int usable value.
218     int step = getchar() - ZERO_CHAR;
219     int ok = 0;
220     int charCount = 0;
221
222     /*while next character is different from end of line,
223      count characters if different from space.*/
224     while( (ok = getchar()) != EOL){
225
226         charCount += (ok == SPACE_CHAR)? 0 : 1;
227     }
228
229     printf("\n" );
230     if (step < INPUT_LOWEST | step > INPUT_HIGHEST | charCount > 0){
231
232         printf(
233             "Enter a value from %d to %d for the table temperature steps:\n",
234             INPUT_LOWEST,
235             INPUT_HIGHEST
236             );
237         return validateInput();
238
239     }
240     else{
241
242         return step;
243     }
244
245 }
```