# Documentation of Day 5

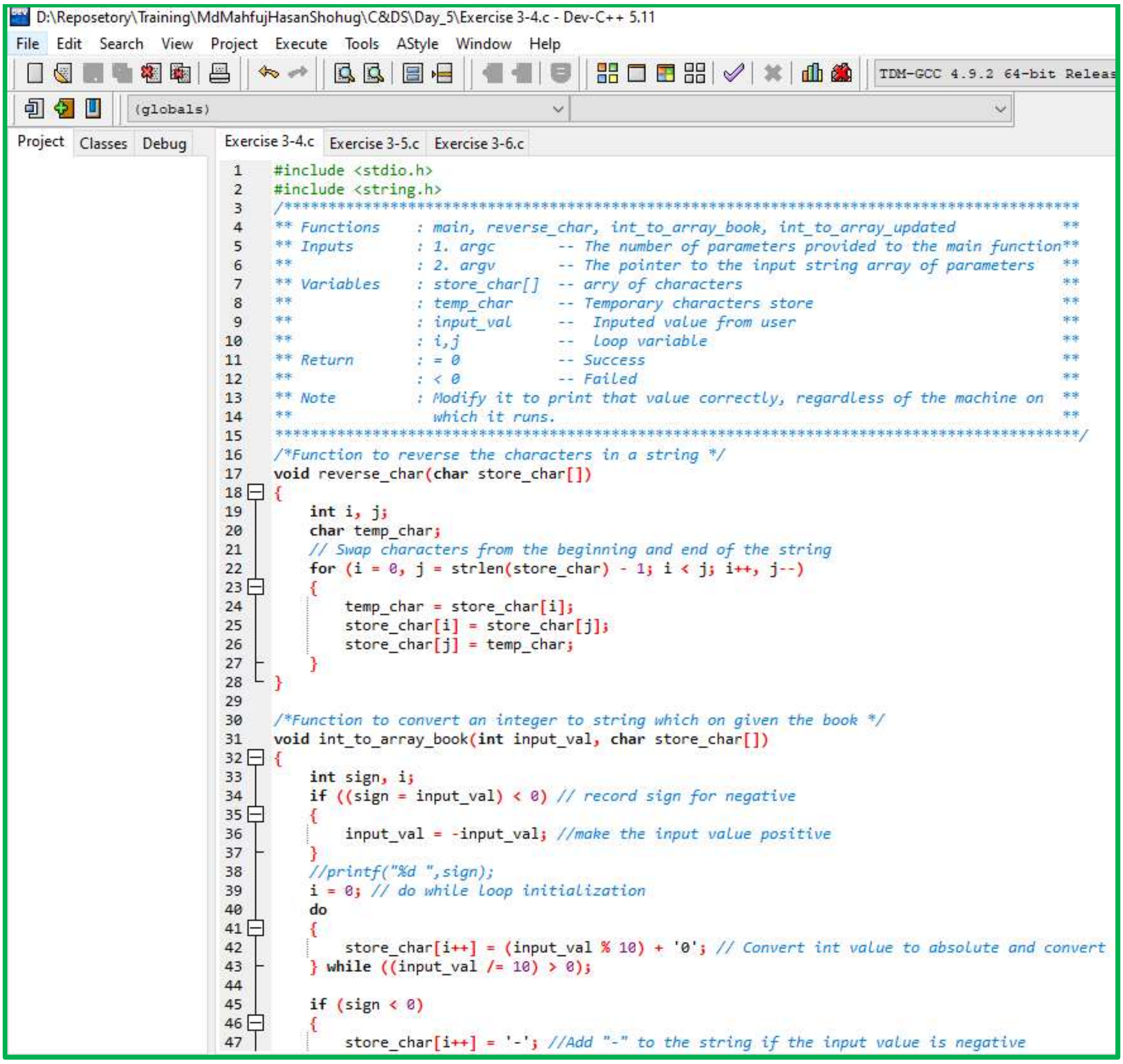
## Md. Mahfuj Hasan Shohug

### BDCOM0019

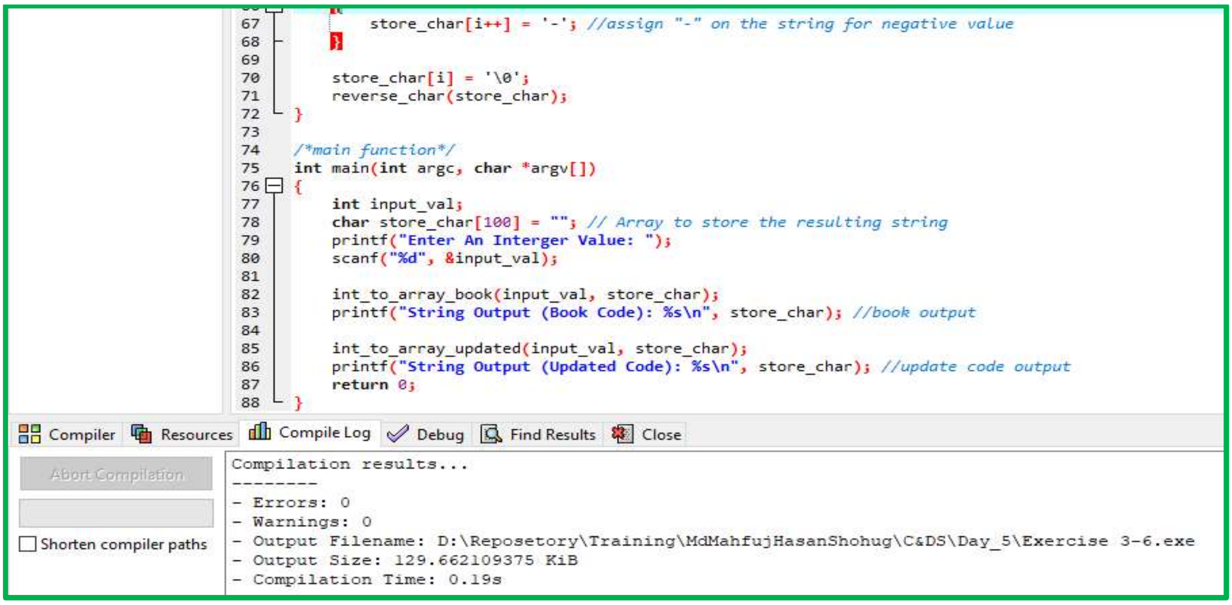
## 1. Exercise 3-4:

Question: In a two's complement number representation, our version of itoa does not handle the largest negative number, that is, the value of n equal to -(2^wordsize-1). Explain why not. Modify it to print that value correctly, regardless of the machine on which it runs.

Source code:


D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-4.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Releas
(globals)
Project Classes Debug
Exercise 3-4.c Exercise 3-5.c Exercise 3-6.c

```c
#include <stdio.h>
#include <string.h>
/************************************************************************************
** Functions     : main, reverse_char, int_to_array_book, int_to_array_updated        **
** Inputs        : 1. argc        -- The number of parameters provided to the main function**
**               : 2. argv        -- The pointer to the input string array of parameters   **
** Variables     : store_char[]  -- arry of characters                                    **
**               : temp_char     -- Temporary characters store                             **
**               : input_val     --  Inputed value from user                               **
**               : i,j           --  Loop variable                                         **
** Return        : = 0           -- Success                                                **
**               : < 0           -- Failed                                                 **
** Note          : Modify it to print that value correctly, regardless of the machine on  **
**                 which it runs.                                                          **
************************************************************************************/
/*Function to reverse the characters in a string */
void reverse_char(char store_char[])
{
    int i, j;
    char temp_char;
    // Swap characters from the beginning and end of the string
    for (i = 0, j = strlen(store_char) - 1; i < j; i++, j--)
    {
        temp_char = store_char[i];
        store_char[i] = store_char[j];
        store_char[j] = temp_char;
    }
}

/*Function to convert an integer to string which on given the book */
void int_to_array_book(int input_val, char store_char[])
{
    int sign, i;
    if ((sign = input_val) < 0) // record sign for negative
    {
        input_val = -input_val; //make the input value positive
    }
    //printf("%d ",sign);
    i = 0; // do while loop initialization
    do
    {
        store_char[i++] = (input_val % 10) + '0'; // Convert int value to absolute and convert
    } while ((input_val /= 10) > 0);

    if (sign < 0)
    {
        store_char[i++] = '-'; //Add "-" to the string if the input value is negative
```

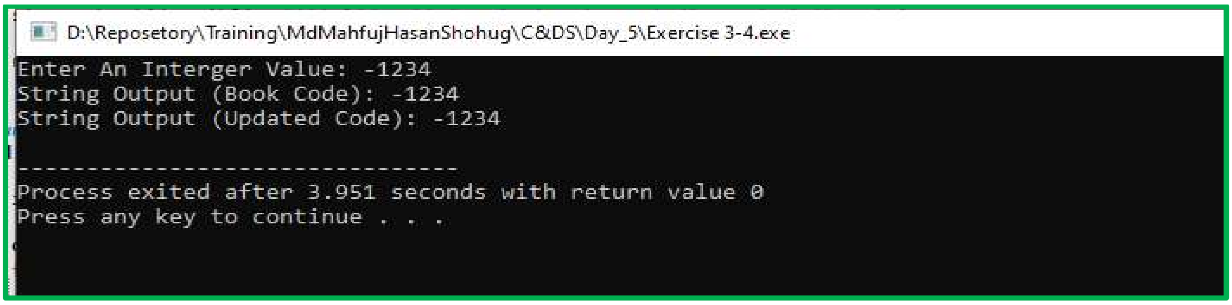


```
        store_char[i++] = '-'; //assign "-" on the string for negative value
    }

    store_char[i] = '\0';
    reverse_char(store_char);
}

/*main function*/
int main(int argc, char *argv[])
{
    int input_val;
    char store_char[100] = ""; // Array to store the resulting string
    printf("Enter An Interger Value: ");
    scanf("%d", &input_val);

    int_to_array_book(input_val, store_char);
    printf("String Output (Book Code): %s\n", store_char); //book output

    int_to_array_updated(input_val, store_char);
    printf("String Output (Updated Code): %s\n", store_char); //update code output
    return 0;
}
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-6.exe
- Output Size: 129.662109375 KiB
- Compilation Time: 0.19s
```

Description: The problem with the original code is that it believes that signed numbers should be represented using a two's complement notation, with the greatest negative value being equal to -(2wordsize-1). This presumption might not be accurate across the board, though. I may change the code to handle the largest negative integer as a special case, which will print the largest negative number accurately regardless of the machine's representation. Here is the code that has been modified to address this problem.

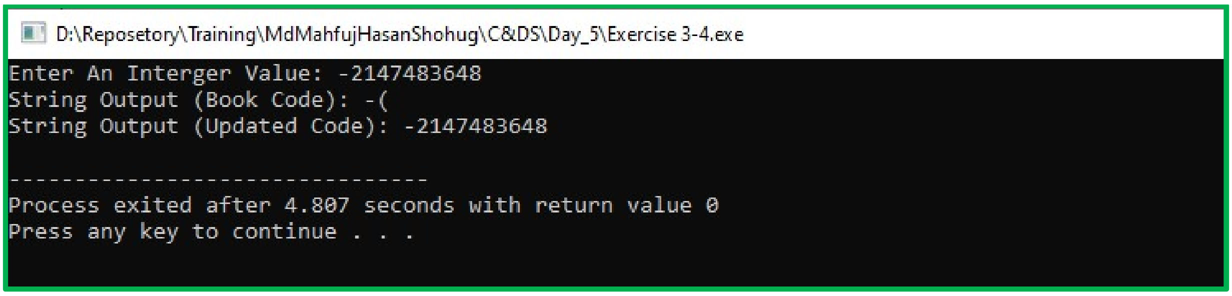
Some output: Inputting general number in the range:


D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-4.exe

```
Enter An Interger Value: -1234
String Output (Book Code): -1234
String Output (Updated Code): -1234

--------------------------------
Process exited after 3.951 seconds with return value 0
Press any key to continue . . .
```

It’s perfectly work booth book and update on the negative value.

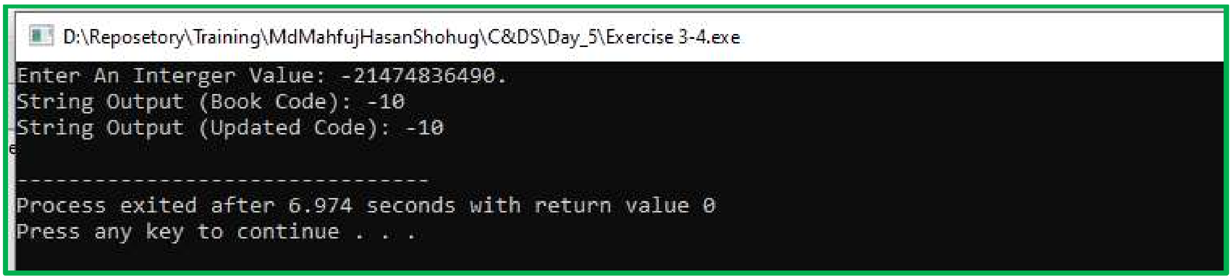
Now inputting negative highest value of integer:


D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-4.exe

```
Enter An Interger Value: -2147483648
String Output (Book Code): -(
String Output (Updated Code): -2147483648

--------------------------------
Process exited after 4.807 seconds with return value 0
Press any key to continue . . .
```

Book code is not work on this output, but updated function is work properly and successfully convert a integer into a string.

But in this input here I am inputting value out of range then output will be random string or other lets see:



D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-4.exe

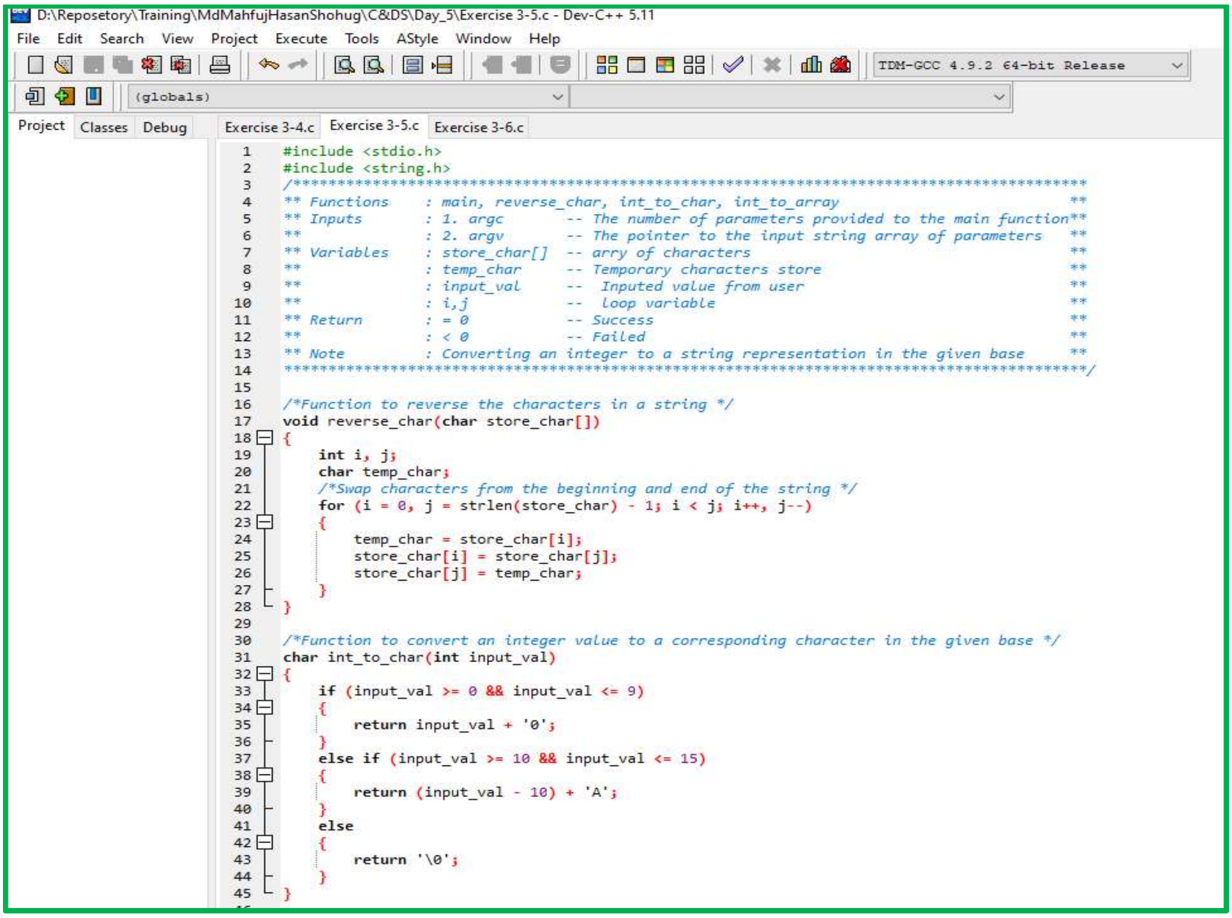
```
Enter An Interger Value: -21474836490.
String Output (Book Code): -10
String Output (Updated Code): -10

--------------------------------
Process exited after 6.974 seconds with return value 0
Press any key to continue . . .
```

which is outside the range of a 32-bit signed integer, the behavior of the program becomes undefined. the value will exceed the limit, leading to unpredictable behavior. The value might wrap around in this situation and be saved as a different integer. It may, for instance, be saved as -21474836490 modulo 232, which is equal to -10. (Output will show depends on the platform).

## 2. Exercise 3-5:

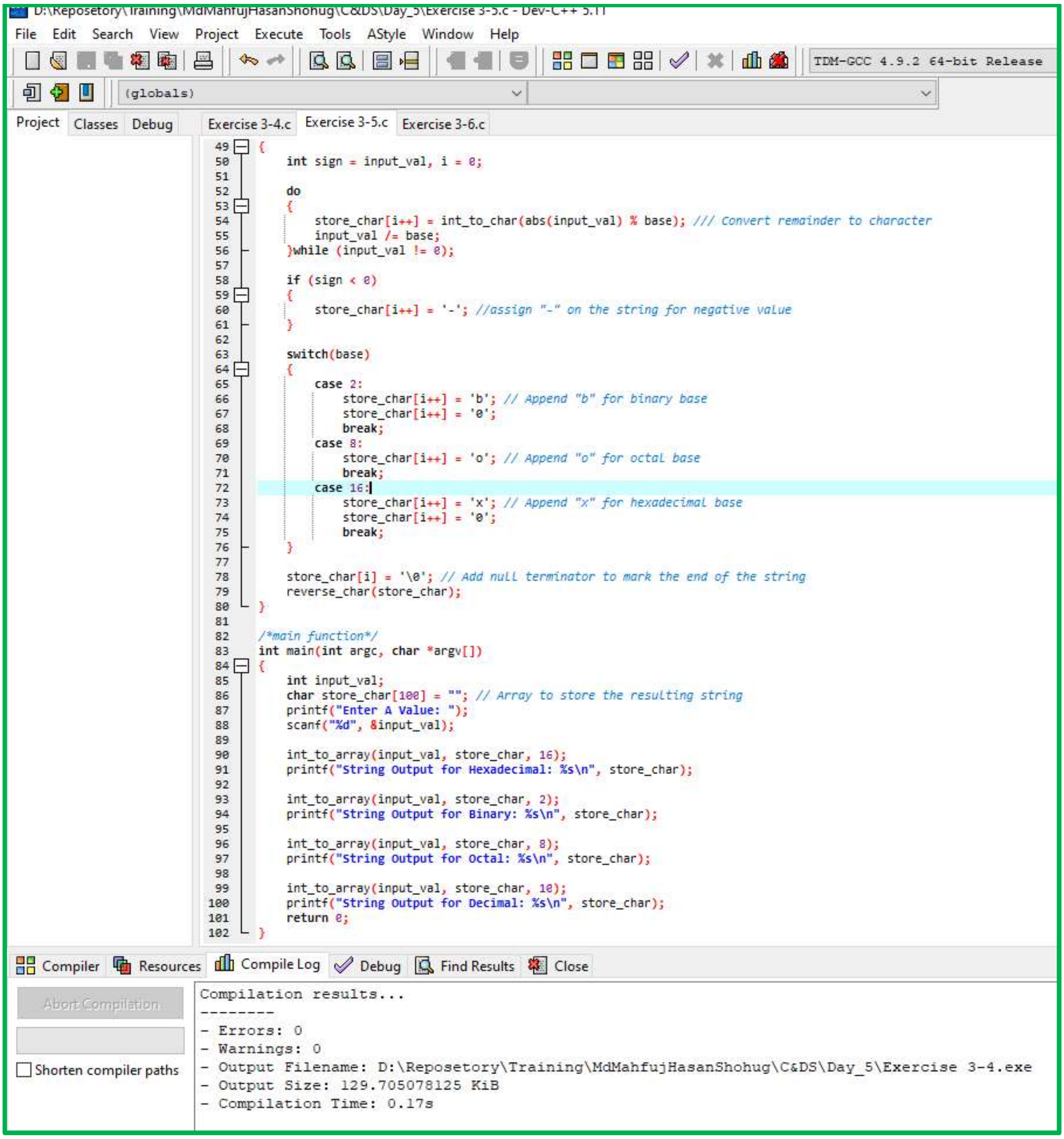
Question: Write the function itob(n,s,b) that converts the integer n into a base b character representation in the string s. In particular, itob(n,s,16) formats s as a Hexadecimal integer in s.

Solving source code with description:



D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-5.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
Exercise 3-4.c Exercise 3-5.c Exercise 3-6.c

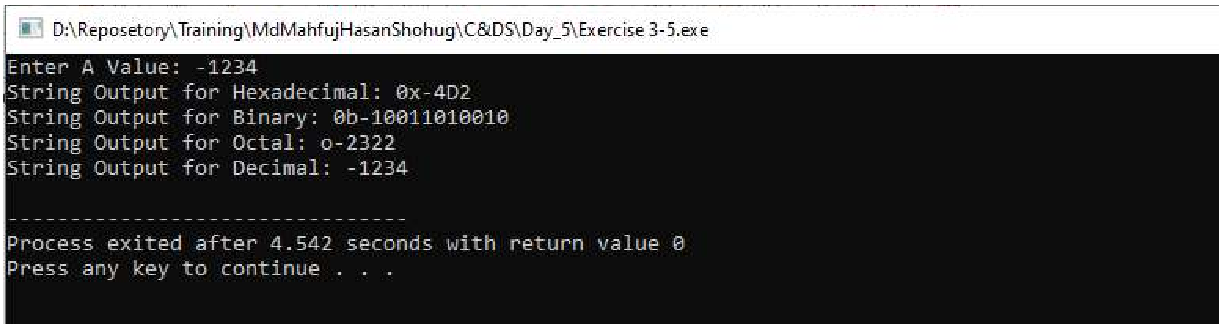
```c
#include <stdio.h>
#include <string.h>
/*******************************************************************************************
** Functions    : main, reverse_char, int_to_char, int_to_array                          **
** Inputs       : 1. argc         -- The number of parameters provided to the main function**
**              : 2. argv         -- The pointer to the input string array of parameters   **
** Variables    : store_char[]  -- arry of characters                                    **
**              : temp_char     -- Temporary characters store                            **
**              : input_val     --  Inputed value from user                              **
**              : i,j           --  loop variable                                        **
** Return       : = 0           -- Success                                               **
**              : < 0           -- Failed                                                **
** Note         : Converting an integer to a string representation in the given base     **
*******************************************************************************************/

/*Function to reverse the characters in a string */
void reverse_char(char store_char[])
{
    int i, j;
    char temp_char;
    /*Swap characters from the beginning and end of the string */
    for (i = 0, j = strlen(store_char) - 1; i < j; i++, j--)
    {
        temp_char = store_char[i];
        store_char[i] = store_char[j];
        store_char[j] = temp_char;
    }
}

/*Function to convert an integer value to a corresponding character in the given base */
char int_to_char(int input_val)
{
    if (input_val >= 0 && input_val <= 9)
    {
        return input_val + '0';
    }
    else if (input_val >= 10 && input_val <= 15)
    {
        return (input_val - 10) + 'A';
    }
    else
    {
        return '\0';
    }
}
```


D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-5.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
Exercise 3-4.c Exercise 3-5.c Exercise 3-6.c
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
□ Shorten compiler paths

```c
{
    int sign = input_val, i = 0;

    do
    {
        store_char[i++] = int_to_char(abs(input_val) % base); /// Convert remainder to character
        input_val /= base;
    }while (input_val != 0);

    if (sign < 0)
    {
        store_char[i++] = '-'; //assign "-" on the string for negative value
    }

    switch(base)
    {
        case 2:
            store_char[i++] = 'b'; // Append "b" for binary base
            store_char[i++] = '0';
            break;
        case 8:
            store_char[i++] = 'o'; // Append "o" for octal base
            break;
        case 16:
            store_char[i++] = 'x'; // Append "x" for hexadecimal base
            store_char[i++] = '0';
            break;
    }

    store_char[i] = '\0'; // Add null terminator to mark the end of the string
    reverse_char(store_char);
}

/*main function*/
int main(int argc, char *argv[])
{
    int input_val;
    char store_char[100] = ""; // Array to store the resulting string
    printf("Enter A Value: ");
    scanf("%d", &input_val);

    int_to_array(input_val, store_char, 16);
    printf("String Output for Hexadecimal: %s\n", store_char);

    int_to_array(input_val, store_char, 2);
    printf("String Output for Binary: %s\n", store_char);

    int_to_array(input_val, store_char, 8);
    printf("String Output for Octal: %s\n", store_char);

    int_to_array(input_val, store_char, 10);
    printf("String Output for Decimal: %s\n", store_char);
    return 0;
}
```

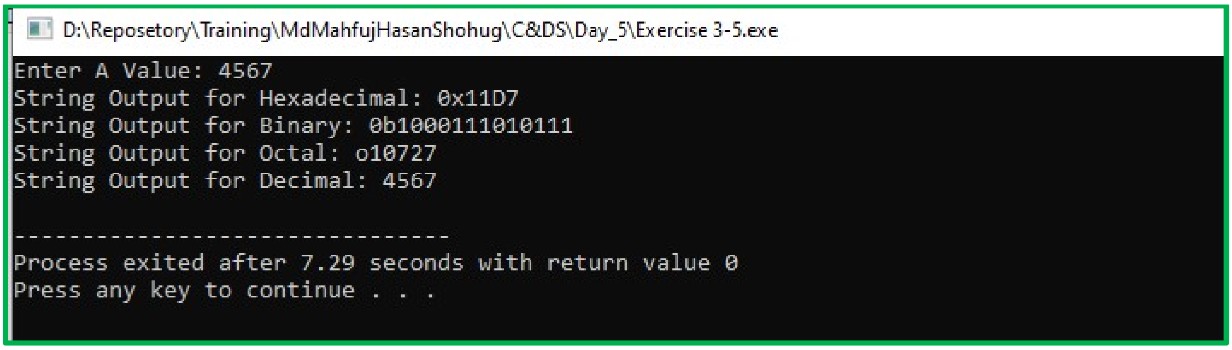
```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-4.exe
- Output Size: 129.705078125 KiB
- Compilation Time: 0.17s
```

Let’s on this code show some output according to the question:


D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-5.exe

```
Enter A Value: -1234
String Output for Hexadecimal: 0x-4D2
String Output for Binary: 0b-10011010010
String Output for Octal: o-2322
String Output for Decimal: -1234

--------------------------------
Process exited after 4.542 seconds with return value 0
Press any key to continue . . .
```

Here on this program one input is directly convert with different number system. Above entering the negative value then its return string with negative value.


D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-5.exe

```
Enter A Value: 4567
String Output for Hexadecimal: 0x11D7
String Output for Binary: 0b1000111010111
String Output for Octal: o10727
String Output for Decimal: 4567

--------------------------------
Process exited after 7.29 seconds with return value 0
Press any key to continue . . .
```

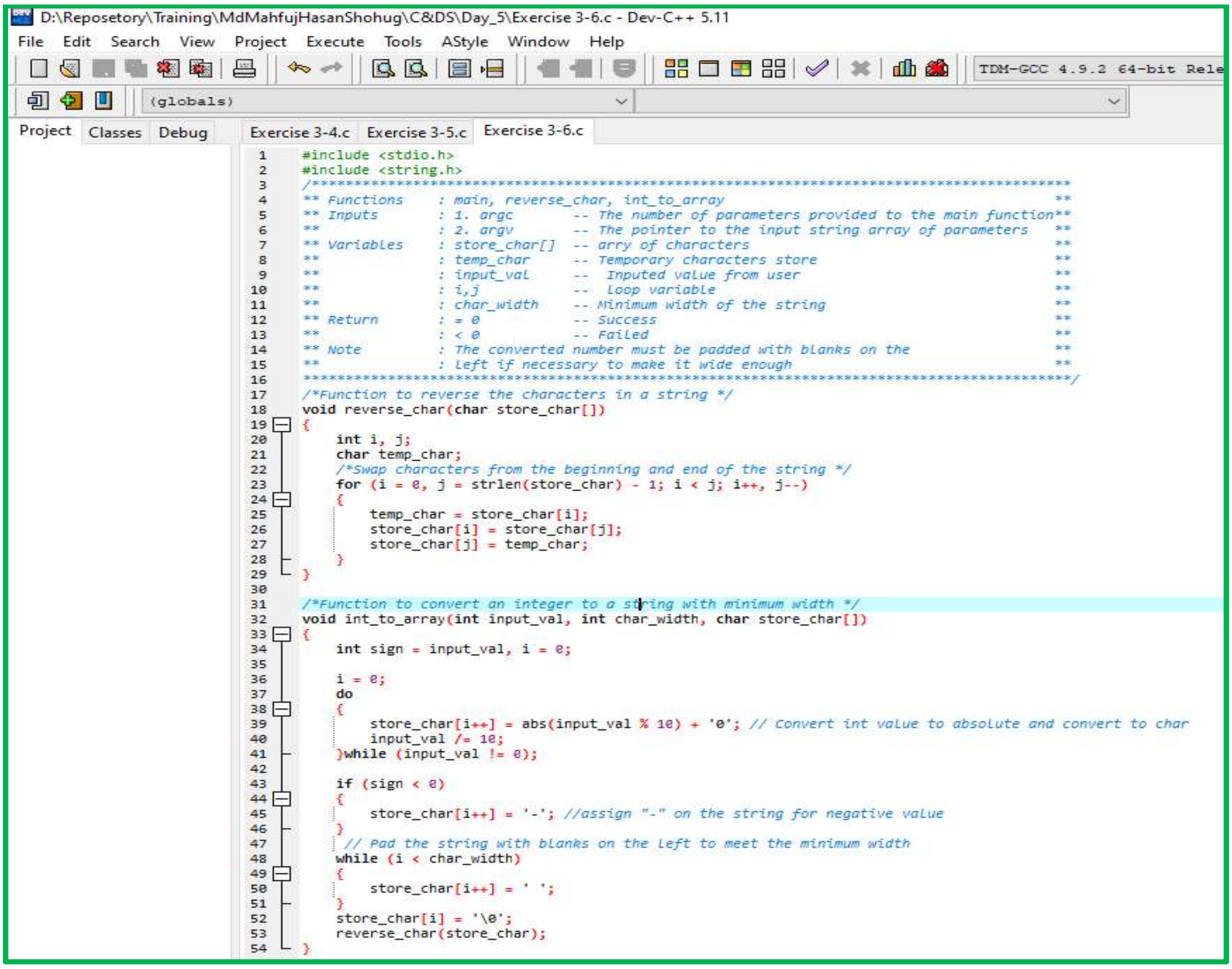
On this ss I input positive value also got the positive value return as a string.

Here I am showing all the number system with call the function with their base value.

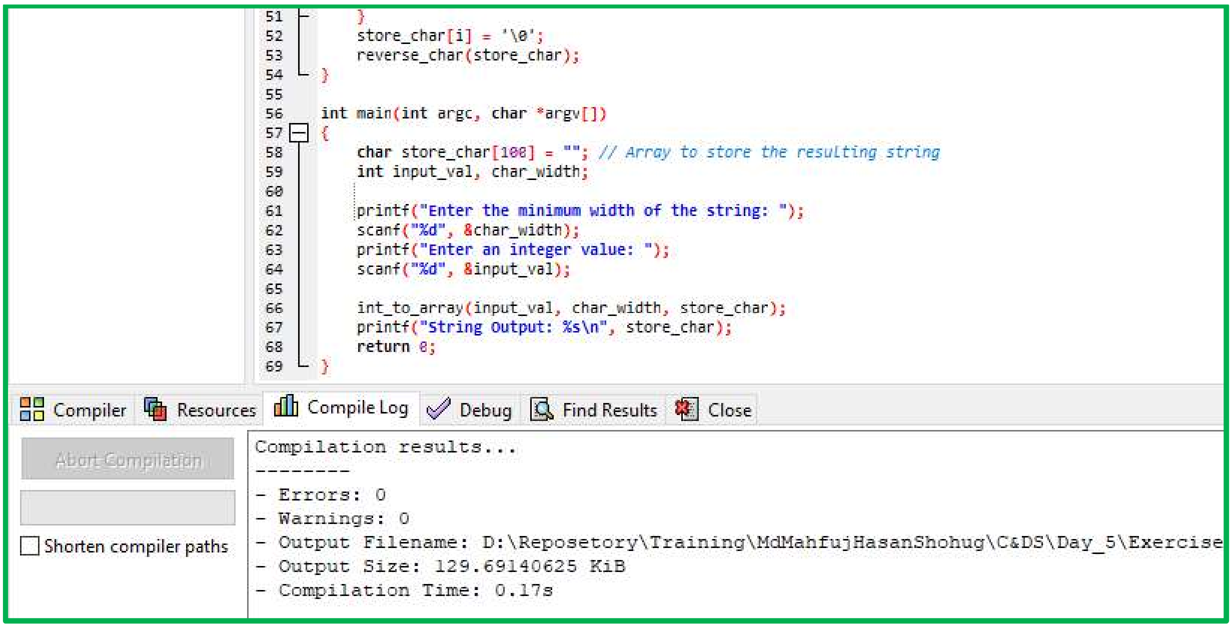
## 3. Exercise 3-6:

Problem: Write a version of itoa that accepts three arguments instead of two. The third Argument is a minimum field width; the converted number must be padded with blanks on the left if necessary to make it wide enough.

Source Code:



```c
#include <stdio.h>
#include <string.h>
/*********************************************************************************
** Functions      : main, reverse_char, int_to_array                            **
** Inputs         : 1. argc        -- The number of parameters provided to the main function**
**                : 2. argv        -- The pointer to the input string array of parameters   **
** Variables      : store_char[]   -- arry of characters                         **
**                : temp_char      -- Temporary characters store                 **
**                : input_val      --  Inputed value from user                   **
**                : i,j            --  Loop variable                             **
**                : char_width     -- Minimum width of the string                **
** Return         : = 0            -- Success                                    **
**                : < 0            -- Failed                                     **
** Note           : The converted number must be padded with blanks on the       **
**                : left if necessary to make it wide enough                     **
*********************************************************************************/
/*Function to reverse the characters in a string */
void reverse_char(char store_char[])
{
    int i, j;
    char temp_char;
    /*Swap characters from the beginning and end of the string */
    for (i = 0, j = strlen(store_char) - 1; i < j; i++, j--)
    {
        temp_char = store_char[i];
        store_char[i] = store_char[j];
        store_char[j] = temp_char;
    }
}

/*Function to convert an integer to a string with minimum width */
void int_to_array(int input_val, int char_width, char store_char[])
{
    int sign = input_val, i = 0;

    i = 0;
    do
    {
        store_char[i++] = abs(input_val % 10) + '0'; // Convert int value to absolute and convert to char
        input_val /= 10;
    }while (input_val != 0);

    if (sign < 0)
    {
        store_char[i++] = '-'; //assign "-" on the string for negative value
    }
     // Pad the string with blanks on the left to meet the minimum width
    while (i < char_width)
    {
        store_char[i++] = ' ';
    }
    store_char[i] = '\0';
    reverse_char(store_char);
}
```

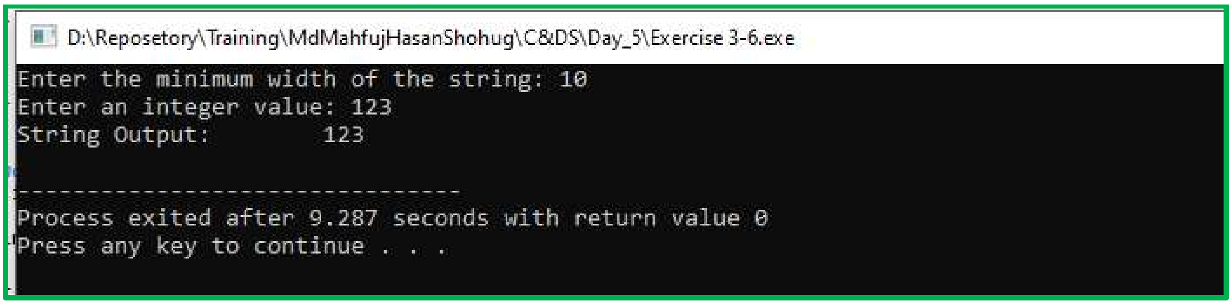


```
    }
    store_char[i] = '\0';
    reverse_char(store_char);
}

int main(int argc, char *argv[])
{
    char store_char[100] = ""; // Array to store the resulting string
    int input_val, char_width;

    printf("Enter the minimum width of the string: ");
    scanf("%d", &char_width);
    printf("Enter an integer value: ");
    scanf("%d", &input_val);

    int_to_array(input_val, char_width, store_char);
    printf("String Output: %s\n", store_char);
    return 0;
}
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise
- Output Size: 129.69140625 KiB
- Compilation Time: 0.17s
```

In this code the program will scan the string minimum size and the  show the output and compare with the string size. Here is some output on this with different input

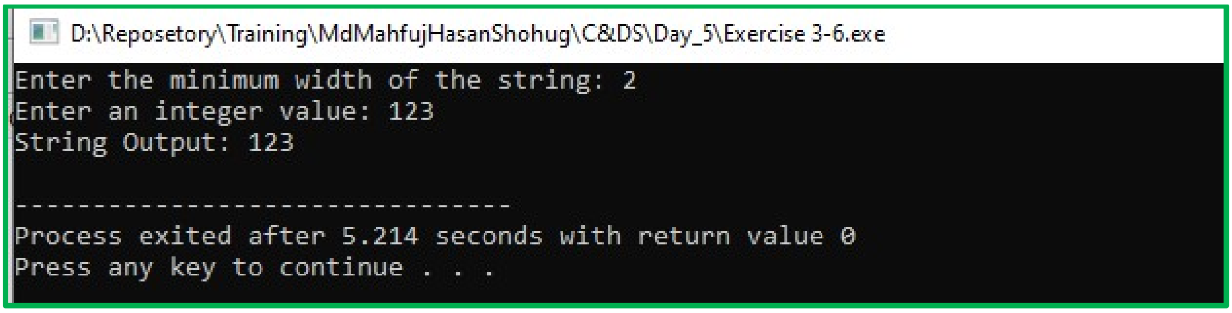
First input correctly with 10 size to 3 size int showing 7 spaces on that:



D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-6.exe

```
Enter the minimum width of the string: 10
Enter an integer value: 123
String Output:        123

--------------------------------
Process exited after 9.287 seconds with return value 0
Press any key to continue . . .
```

Lets get input 2 size and input 3 size int let's see what happened:



D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise 3-6.exe

```
Enter the minimum width of the string: 2
Enter an integer value: 123
String Output: 123

--------------------------------
Process exited after 5.214 seconds with return value 0
Press any key to continue . . .
```

It got show properly because of my code maximum size is 100.

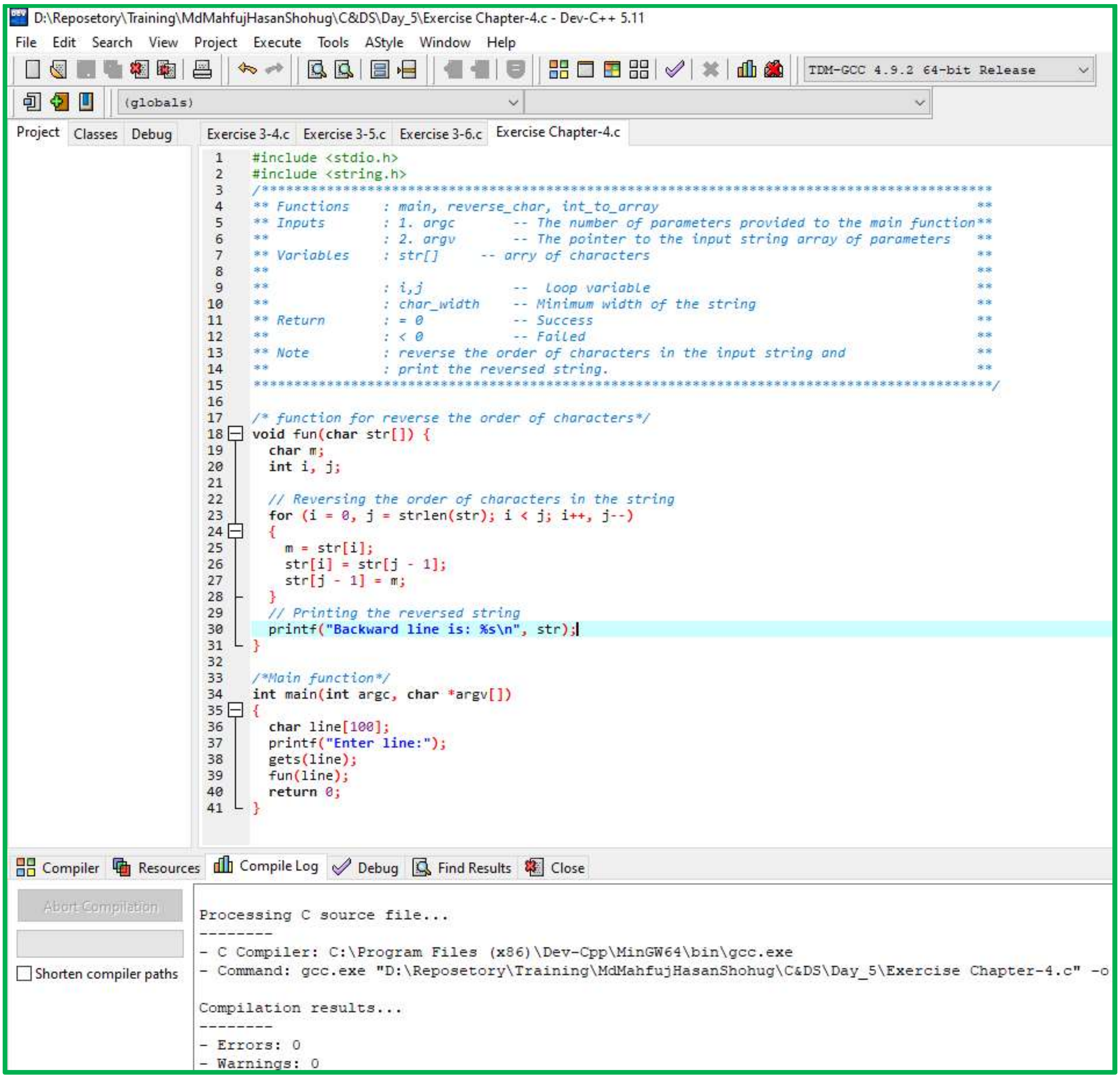
The revised itoa function is used in the main function to show how to use it. It creates a string from the decimal number -123 and inserts blanks to make it at least 10 characters wide. Next, the padded string and the original number are printed.

## 4. Chapter 4 Question:

1.function fun()try to store a string to backward sequence, fill in the blanks

```
void fun (char str[])
{ char m; int i,j;
 for(i=0,j=strlen(str);i<__________;i++,j--)
 { m=str[i];
   str[i]=__________;
   str[j-1]=m;
 }
 printf("%s\n",str);
}
```

Source code:



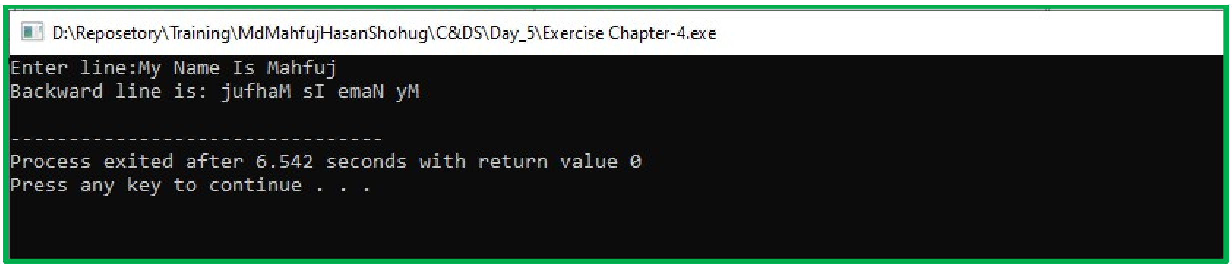
D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise Chapter-4.c - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
Exercise 3-4.c Exercise 3-5.c Exercise 3-6.c Exercise Chapter-4.c

```c
#include <stdio.h>
#include <string.h>
/*************************************************************************************
** Functions    : main, reverse_char, int_to_array                                  **
** Inputs       : 1. argc        -- The number of parameters provided to the main function**
**              : 2. argv        -- The pointer to the input string array of parameters   **
** Variables    : str[]     -- arry of characters                                   **
**                                                                                  **
**              : i,j            --  Loop variable                                  **
**              : char_width     -- Minimum width of the string                     **
** Return       : = 0            -- Success                                         **
**              : < 0            -- Failed                                          **
** Note         : reverse the order of characters in the input string and           **
**              : print the reversed string.                                        **
*************************************************************************************/

/* function for reverse the order of characters*/
void fun(char str[]) {
  char m;
  int i, j;

  // Reversing the order of characters in the string
  for (i = 0, j = strlen(str); i < j; i++, j--)
  {
    m = str[i];
    str[i] = str[j - 1];
    str[j - 1] = m;
  }
  // Printing the reversed string
  printf("Backward line is: %s\n", str);
}

/*Main function*/
int main(int argc, char *argv[])
{
  char line[100];
  printf("Enter line:");
  gets(line);
  fun(line);
  return 0;
}
```

```
Processing C source file...
--------
- C Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\gcc.exe
- Command: gcc.exe "D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise Chapter-4.c" -o

Compilation results...
--------
- Errors: 0
- Warnings: 0
```

Output:



D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_5\Exercise Chapter-4.exe

```
Enter line:My Name Is Mahfuj
Backward line is: jufhaM sI emaN yM

--------------------------------
Process exited after 6.542 seconds with return value 0
Press any key to continue . . .
```

Reversing the string: The fun function implements a loop to reverse the order of characters in the input string str using the following steps: The loop starts with i initialized to 0 and j initialized to the length of the string (strlen(str)). For each iteration of the loop, the characters at positions i and j-1 in the string are swapped using a temporary variable m. The loop continues until i becomes greater than or equal to j.

Input was: “My Name Is Mahfuj”

Output by reversing the string.