FUNCTION

AIM:

To create a parameterized function for bit manupulation and reverse the string and conversion of integer to string.

ALGORITHM:

· stdio.h is included to perform input and output operations.

· Function declarations are done for bit manipulation, integer-to-string conversion, and string reversal.

· An unsigned character variable is declared to store the input number.

· The user enters an integer value which is stored in Input.

· The user enters the bit position to be operated on.

· SetBit() is called to set the specified bit and the result is displayed.

· ResetBit() is called to clear the specified bit and the result is displayed.

· ToggleBit() is called to toggle the specified bit and the result is displayed.

· ReadBit() is called to read the value of the specified bit and the result is displayed.

· IntToString() is called to convert the input number into a string.

· The converted string is displayed.

· Reverse() is called to reverse the converted string.

· The reversed string is displayed.

· The program ends after displaying all results.

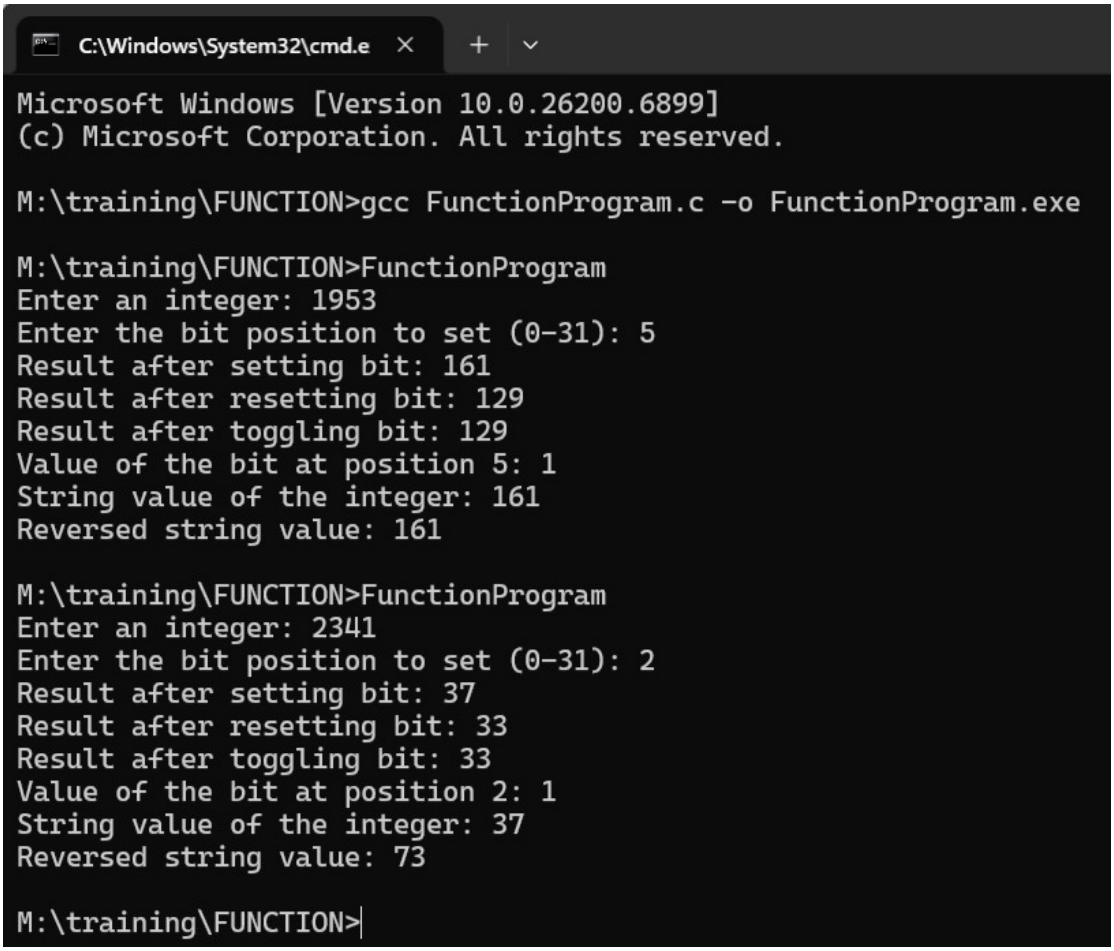
PROGRAM:

```c
/*
* Program to demonstrate a function with parameters and return  value
* Author   : MUTHUGANESH S
* Date     : 16/01/2026
* Filename : FunctionProgram.c
* retval   : void
*/

#include <stdio.h>

//function declarations
unsigned char SetBit(unsigned char Num, int Pos);
unsigned char ResetBit(unsigned char Num, int Pos);
unsigned char ToggleBit(unsigned char Num, int Pos);
int ReadBit(unsigned char Num, int Pos);
void IntToString(int Num, char Result[]);
```

```c
void Reverse(char Str[], char Reversed[]);

int main(void){
    int Position, Result;
    unsigned char Input;

    printf("Enter an integer: ");
    scanf("%hhu", &Input);

    printf("Enter the bit position to set (0-31): ");
    scanf("%d", &Position);

    // Function call to set the bit
    Result = SetBit(Input, Position);
    printf("Result after setting bit: %d\n", Result);

    // Function call to reset the bit
    Result=ResetBit(Input, Position);
    printf("Result after resetting bit: %d\n", Result);

    // Function call to toggle the bit
    Result=ToggleBit(Input, Position);
    printf("Result after toggling bit: %d\n", Result);

    // Function call to read the bit
    Result=ReadBit(Input, Position);
    printf("Value of the bit at position %d: %d\n", Position, Result);

    // Function call to convert integer to string
    char StringValue[100];
    IntToString(Input, StringValue);
    printf("String value of the integer: %s\n", StringValue);

    char ReversedString[100];
    Reverse(StringValue, ReversedString);
    printf("Reversed string value: %s\n", ReversedString);

    return 0;

}

unsigned char SetBit(unsigned char Num, int Pos){
    unsigned char Value;
    Value=Num | (1 << Pos);
    return Value;
}

unsigned char ResetBit(unsigned char Num, int Pos){
    unsigned char Value;
    Value=Num & ~(1 << Pos);
    return Value;
}

unsigned char ToggleBit(unsigned char Num, int Pos){
```

```
    unsigned char Value;
    Value=Num ^ (1 << Pos);
    return Value;
}
int ReadBit(unsigned char Num, int Pos){
    int Value;
    Value=(Num >> Pos) & 1;
    return Value;
}

void IntToString(int Num, char Result[])
{
    int i = 0, j;
    char Temp[20];

    if (Num == 0)
    {
        Result[i++] = '0';
        Result[i] = '\0';
        return;
    }

    while (Num > 0)
    {
        Temp[i++] = (Num % 10) + '0';
        Num /= 10;
    }

    for (j = 0; j < i; j++)
        Result[j] = Temp[i - j - 1];

    Result[i] = '\0';
}

void Reverse(char Str[], char Reversed[])
{
    int len = 0, i;
    while (Str[len] != '\0')
        len++;

    for (i = 0; i < len; i++)
        Reversed[i] = Str[len - i - 1];

    Reversed[len] = '\0';
}
```

OUTPUT:


C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.26200.6899]
(c) Microsoft Corporation. All rights reserved.
M:\training\FUNCTION>gcc FunctionProgram.c -o FunctionProgram.exe
M:\training\FUNCTION>FunctionProgram
Enter an integer: 1953
Enter the bit position to set (0-31): 5
Result after setting bit: 161
Result after resetting bit: 129
Result after toggling bit: 129
Value of the bit at position 5: 1
String value of the integer: 161
Reversed string value: 161
M:\training\FUNCTION>FunctionProgram
Enter an integer: 2341
Enter the bit position to set (0-31): 2
Result after setting bit: 37
Result after resetting bit: 33
Result after toggling bit: 33
Value of the bit at position 2: 1
String value of the integer: 37
Reversed string value: 73
M:\training\FUNCTION>