1.NUMBER TO WORD

AIM:

To convert the number to words without using fuctions and array by only using loops,decision statements and operators.

PROCEDURE:

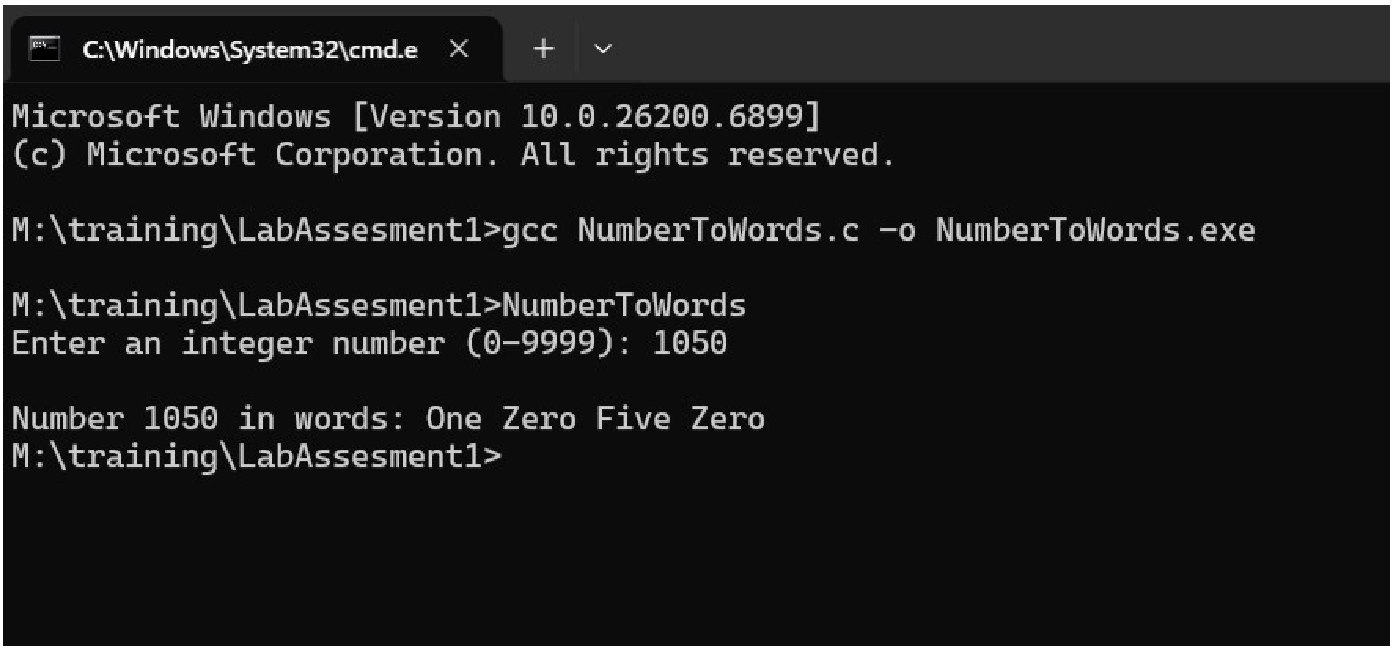
- stdio.h is included to use input and output functions.
- Integer variables are created to store the number, reversed number, and zero counters.
- The user enters an integer number between 0 and 9999.
- The entered number is checked to see if it is within the given range.
- The number is processed digit by digit to create its reverse.
- Zeros at the end of the number are counted while reversing.
- The reversed number is used to keep the original digit order.
- Each digit is extracted from the reversed number one by one.
- A switch statement is used to convert each digit into its word form.
- The word of each digit is printed on the screen.
- The reversed number is reduced by removing the last digit each time.
- After all digits are printed, the counted zeros are printed.
- The complete number in words is displayed.
- The program ends after showing the result.

PROGRAM:

```c
/*
* Program to convert a given number into words using operators,loops and
decision statements
* Author   : MUTHUGANESH S
* Date     : 15/01/2026
* Filename : NumberToWords.c
* retval   : void
*/

#include <stdio.h>

int main(void){
    int Number,ReversedNumber=0;
    int ZeroFlag=0,ZeroCount=0;
    printf("Enter an integer number (0-9999): ");
```

```
scanf("%d",&Number);

printf("\nNumber %d in words: ",Number);

if(Number<0||Number>9999){
    printf("Out of range! Please enter a number between 0 and 9999.\n");
    return 0;
}
//Reverse the number to maintain the order of digits
while (Number>0)
{
    if(ZeroFlag==0 && Number%10==0){
        ZeroCount++;
    }
    else{
        ZeroFlag=1;
    }
    //Extract the last digit and build the reversed number
    ReversedNumber = (ReversedNumber * 10) + (Number % 10);
    Number /= 10;
}


//Loop to extract each digit and convert to words
while(ReversedNumber>0){
    //Extract the last digit
    int digit=ReversedNumber%10;

    //Convert digit to word using switch case
    switch(digit){

        case 0:{
            printf("Zero ");
            break;
        }
        case 1:{
            printf("One ");
            break;
        }
        case 2:{
            printf("Two ");
            break;
        }
        case 3:{
            printf("Three ");
            break;
        }
        case 4:{
            printf("Four ");
            break;
        }
        case 5:{
            printf("Five ");
            break;
            }
            case 6:{
                printf("Six ");
                break;
            }
            case 7:{
                printf("Seven ");
                break;
            }
            case 8:{
                printf("Eight ");
                break;
            }
            case 9:{
                printf("Nine ");
                break;
            }

        }

        //Remove the last digit
        ReversedNumber/=10; //Assignment operator
    }

    //Print the trailing zeros
        for(int i=0;i<ZeroCount;i++){
            printf("Zero ");
        }

    return 0;
}
```

OUTPUT:



```
Microsoft Windows [Version 10.0.26200.6899]
(c) Microsoft Corporation. All rights reserved.

M:\training\LabAssesment1>gcc NumberToWords.c -o NumberToWords.exe

M:\training\LabAssesment1>NumberToWords
Enter an integer number (0-9999): 1050

Number 1050 in words: One Zero Five Zero
M:\training\LabAssesment1>
```

2.PROGRAM USING MACROS

AIM:

To define the ingnition status using the an integer reading and make use macros and display the result.

PROCEDURE:

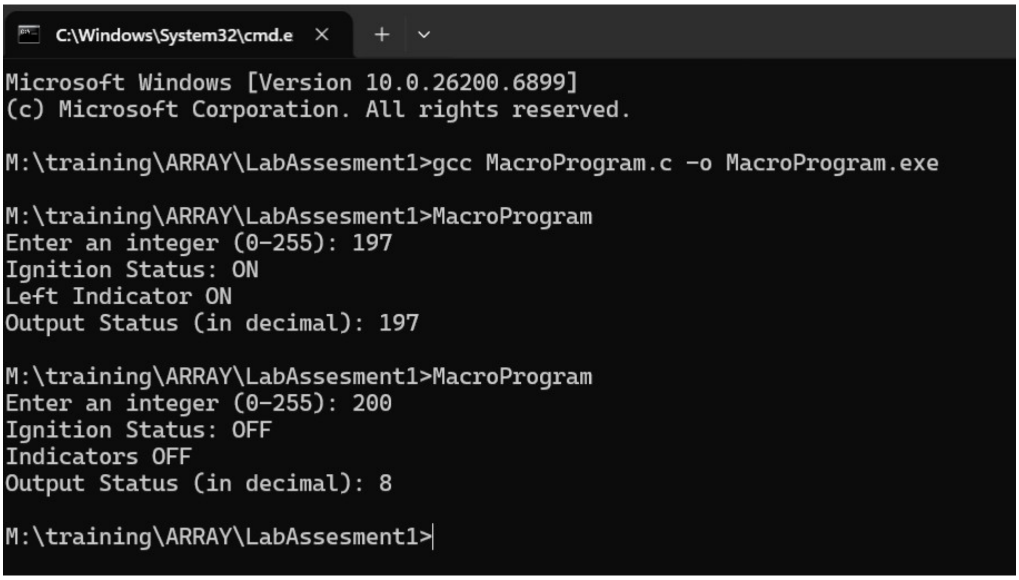
- stdio.h is included to perform input and output operations.
- Macros are defined to read, set, and clear individual bits in a variable.
- Macros ON and OFF are defined to represent logic values.
- An unsigned character variable Input is declared to store the 8-bit value.
- A number between 0 and 255 is read from the user and stored in Input.
- Bit D0 is read as the ignition switch.
- Bit D1 is read as the right indicator switch.
- Bit D2 is read as the left indicator switch.
- Bits D5, D6, and D7 are cleared to reset output status.
- Ignition status bit D6 is set if the ignition switch is ON.
- Right indicator status bit D5 is set if the right indicator switch is ON.
- Left indicator status bit D7 is set if the left indicator switch is ON.
- Both indicator status bits are set when parking condition is met.
- Ignition status is printed based on bit D6.
- Indicator status is printed based on bits D5 and D7.
- The final modified input value is displayed in decimal form.
- Program execution ends.

PROGRAM:

```
/*
* Program to demonstrate the Indicator using Macros
* Author   : MUTHUGANESH S
* Date     : 15/01/2026
* Filename : MacroProgram.c
* retval   : void
*/
```

```
//header file
#include <stdio.h>

//macro
#define READBIT(VAR,POS) ((VAR >> POS) & 1)
#define SETBIT(VAR,POS) (VAR |= (1 << POS))
#define CLEARBIT(VAR,POS) (VAR &= ~(1 << POS))
#define ON 1
#define OFF 0
int main(){
    unsigned char Input;
    printf("Enter an integer (0-255): ");
    scanf("%hhu", &Input);

    //Read Switches
    unsigned char IgnitionSwitch=READBIT(Input,0);
    unsigned char RightIndicatorSwitch=READBIT(Input,1);
    unsigned char LeftIndicatorSwitch=READBIT(Input,2);

    CLEARBIT(Input,5);//Right Indicator Status
    CLEARBIT(Input,6);//Ignition Status
    CLEARBIT(Input,7);//Left Indicator Status

    if(IgnitionSwitch==ON){
        SETBIT(Input,6); //Ignition ON

        if(RightIndicatorSwitch==ON && LeftIndicatorSwitch==OFF){
            SETBIT(Input,5); //Right Indicator ON
            SETBIT(Input,7); //Left Indicator ON
        }
        else if(RightIndicatorSwitch==ON){
            SETBIT(Input,5); //Left Indicator ON
        }
        else if(LeftIndicatorSwitch==ON){
            SETBIT(Input,7); //Right Indicator ON
        }
    }

    if(READBIT(Input,6)==ON){
        printf("Ignition Status: ON\n");
    }
    else{
        printf("Ignition Status: OFF\n");
    }

    if(READBIT(Input,5)==ON && READBIT(Input,7)==ON){
        printf("Parking On\n");
    }
    else if(READBIT(Input,5)==ON){
        printf("Right Indicator ON\n");
    }
    else if(READBIT(Input,7)==ON){
        printf("Left Indicator ON\n");
    }
    else{
        printf("Indicators OFF\n");
    }

    printf("Output Status (in decimal): %d\n", Input);
    return 0;

}
```

OUTPUT:



```
Microsoft Windows [Version 10.0.26200.6899]
(c) Microsoft Corporation. All rights reserved.

M:\training\ARRAY\LabAssesment1>gcc MacroProgram.c -o MacroProgram.exe

M:\training\ARRAY\LabAssesment1>MacroProgram
Enter an integer (0-255): 197
Ignition Status: ON
Left Indicator ON
Output Status (in decimal): 197

M:\training\ARRAY\LabAssesment1>MacroProgram
Enter an integer (0-255): 200
Ignition Status: OFF
Indicators OFF
Output Status (in decimal): 8

M:\training\ARRAY\LabAssesment1>
```