**LAB ASSESSMENT**

**COE - KIOT - C Programming - Week 1**

**NAME : NAVEETH KHAN J**

**DATA : 04/09/2024**

**Problem Statement:**

1. Write a C program to get any numbers from 1-9999 and print the numbers in words. (example : Number = 123 then print "One Two Three"). get the input by using scanf . Use only operators, loops, and Decision statements. Should not use functions, array.

**Solution:**

#include<stdio.h>

int main(void) {

    //getting the user input

    unsigned int InputNumber;

    scanf("%d", &InputNumber);

    //using digit place, to extract the single digit individually

    unsigned int tens = 1;

    int DuplicateInputNumber = InputNumber / 10; // Used to achieve the digit place

    while (DuplicateInputNumber > 0) { // runs to calculate the tens place

        tens \*= 10;

        DuplicateInputNumber /= 10;

    }

    while (tens > 0) {  // tens is setting to handle all the edge cases in the problem statement, especially trailing zeros

        int num = InputNumber / tens; //extract individual digits

        switch (num)

        {   //checking all the necessary condition to print the integer in str using switch.

        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;

        default:

            printf("Zero ");

            break;

        }

        InputNumber %= tens;    //every iteration reducing the inputnumber

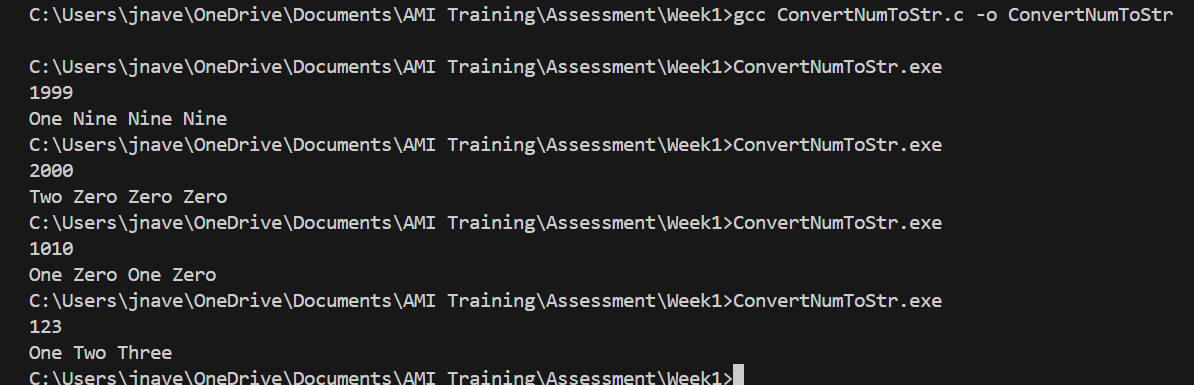
        tens /= 10; //every iteration need to reduce the digit place,

    }

    return 0;   // Indicating successfully execution of a C Program.

}

Output:



**Explanation:**

1. Get the Input Number from the User.
2. Defined tens variable to determine and store the number of digits in a 10 multiple.
3. Traverse the InputNumber individually from starting digit to ending with using loops,
4. Inside while, defined the switch statement to print the integer in string format.
5. In every iteration, reduce the Input Number and tens variable by modulo by tens and divided by 10.

Write a C program to solve below statements

1. Get any number between 0-255 from user and store that in unsigned char variable "Input"

1. In "Input" D0 is Ignition Switch, D1 - Right Indicator Switch, D2 Left indicator Switch, D3&D4 - not used , D5 - Right Indicator status , D6 - Ignition Status, D7 - Left Indicator status
2. By using MACROS read bits D0, D1 & D2 modify the bits D5, D6 & D7. keep D3 & D4 as it is.
3. if D0 is high then set D6 as high and check for indicator switch value and modify the indicator status, if D0 is low only modify the D6 , Indicator status should be OFF even indicator switches are high
4. print outputs as if Ignition is high the IGINITION ON and if any indicator switch is High then print the corresponding indicator is On, if both indicator switch is in High state then print PARKING ON
5. also print the Input value after modification

**Solution:**

#include<stdio.h>

#define READBIT(Input, Position) (Input & (1 << Position))

#define SETBIT(Input, Position) (Input ^ (1 << Position))

#define RESETBIT(Input, Position) (Input & ~(1 << Position))

int main(void) {

    // Used for reading the user input

    unsigned char Input;

    scanf("%hhu", &Input);

    // Need to read individual bit for checking the status

    int D0 = READBIT(Input, 0);

    int D1 = READBIT(Input, 1);

    int D2 = READBIT(Input, 2);

    int D5 = READBIT(Input, 5);

    int D6 = READBIT(Input, 6);

    int D7 = READBIT(Input, 7);

    // If our D0 is set or not, we need to modify on both the condition for D6,

    D6 = D0;

    // checking the status of D0

    if (D0) {

        D5 = D0;    // Set D5, if D0 is high

        D7 = D0;    // Set D7, if D0 is high

        Input = SETBIT(Input, 5);   // Since the we taken seperately for checking, we need to update it in Input Variable

        Input = SETBIT(Input, 7);   // We need to update in Input Variable in the position of 7.

        Input = SETBIT(Input, 6);   // We need to update in Input Variable in the position of 6

    } else {

        D5 = 0; // If the D0 is low, we need to reset the bit for D5.

        D7 = 0; // If the D0 is low, we need to reset the bit for D7.

        Input = RESETBIT(Input, 5); // We need to update as well in the input, the above changes.

        Input = RESETBIT(Input, 7); // Update in Input Number for D7.

        Input = RESETBIT(Input, 6); // Update in Input Number for D6.

    }

    if (D6) {   // If D6 is high, printf Ignition status is "On"

        printf("IGNITION ON\n");

    }

    if (D5) {   // If the D5 is high, it means right indicator is "On"

        printf("Right Indicator is ON \n");

    }

    if (D7) {   // If the D7 is high, it means left indicator in "On"

        printf("Left Indicator is ON \n");

    }

    if (D5 && D7) {     // If both indicator is high, print "PARKING ON"

        printf("PARKING ON\n");

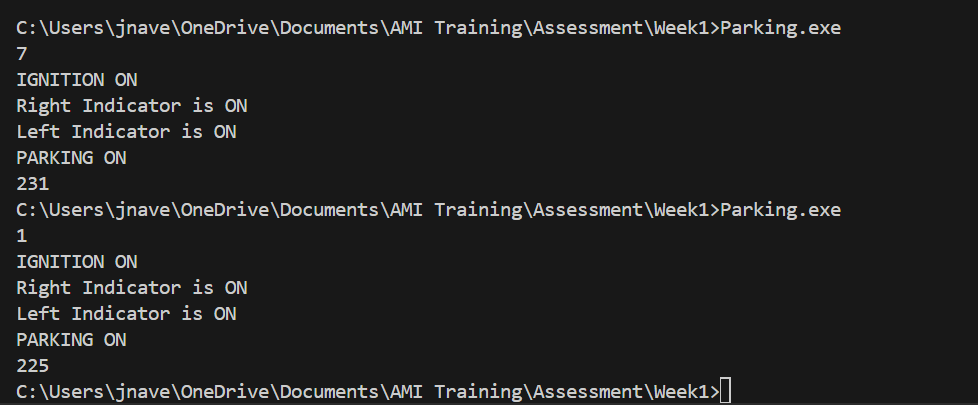
    }

    printf("%d", Input); // Show the original value

    return 0; // Indicating successfull execution of a C Program.

}

**Output:**

****