**Expt No.: 3 Name: Swaranjana Nayak**

**Date: 28/08/2020 Reg. No.: 19BCE0977**

**CYCLE SHEET 3**

**(Based on Functions)**

**Aim:**

To solve the given problem using C.

1. **Design a number converter system**

**Code:**

*// Number converter system in C*

*// Cycle sheet 3 program*

*// Maximum 50 digits for each representation*

*// Doesn't convert decimals*

# include <stdio.h>

# include <stdlib.h>

# include <math.h>

# define N 50

**int** m\_to\_decimal(**char** arr**[]**, **int** m, **int** l);

**void** decimal\_to\_n(**int** num, **int** n);

**int** valid(**char** arr**[]**, **int** m, **int** l);

**int** main()

{

**int** m, n, l = 0, status, QW;

**char** number[N];

    printf("\n------ PROGRAM TO CONVERT NUMBER FROM BASE M TO BASE N ------\n");

    printf("2 <= m,n <= 16\n\n");

    printf("Enter m: ");

    status = scanf("%d", &m);

    fflush(stdin);

    while (status == 0 || m < 2 || m > 16)

    {

        printf("Invalid input! Enter m: ");

        status = scanf("%d", &m);

        fflush(stdin);

    }

    printf("\nEnter n: ");

    status = scanf("%d", &n);

    fflush(stdin);

    while (status == 0 || n < 2 || n > 16)

    {

        printf("Invalid input! Enter n: ");

        status = scanf("%d", &n);

        fflush(stdin);

    }

    while(m == n)

    {

        printf("\nYou are converting to the same base you are entering the number in!\n");

        printf("Enter a different values of n: ");

        status = scanf("%d", &n);

        fflush(stdin);

        while (status == 0 || n < 2 || n > 16)

        {

            printf("\nInvalid input! Enter n: ");

            status = scanf("%d", &n);

            fflush(stdin);

        }

    }

    printf("\nEnter the number: ");

    fgets(number, N - 1, stdin);

for(l = 0; number[l] != '\n'; l++);

*//printf("%d\n", l);*

    if(valid(number, m, l) == 0)

    {

        printf("Invalid input!\n");

        return -1;

    }

    else

    {

        QW = m\_to\_decimal(number, m, l);

*//printf("%d\n", QW);*

        printf("\nThe resulting number is: \n");

        decimal\_to\_n(QW, n);

        return 0;

    }

}

*// Function to convert between base m and decimal*

*// Where 1 < m <= 16*

*// Input is character array of length l*

**int** m\_to\_decimal(**char** arr**[]**, **int** m, **int** l)

{

**char** residue[16] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};

**int** weight[N];

**int** i, j, num = 0;

*// Getting the weights*

    for(i = 0; i < l; i++)

    {

        for(j = 0; j < 16; j++)

        {

            if(arr[i] == residue[j])

            {

                weight[i] = j;

            }

        }

    }

*// Converting the number to decimal*

    for(i = 0; i < l; i++)

    {

        num += weight[i]\*pow(m, l-1-i);

    }

    return num;

}

*// Function to convert the number from base 10 to base n*

*// Where 1 < n <= 16*

*// Input is decimal number n*

**void** decimal\_to\_n(**int** num, **int** n)

{

**char** residue[16] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};

**char** c[N], converted[N];

**int** i = 0, j, rem, temp;

    while (num != 0)

    {

        rem = num % n;

        c[i] = residue[rem];

        num /= n;

        i++;

    }

    c[i] = '\0';

    converted[i] = '\0';

    for(j = 0; c[j] != '\0'; j++)

    {

        converted[i-1-j] = c[j];

    }

    printf("%s\n\n", converted);

}

*// Function to check is input is valid*

*// Takes input of given character array, length of string and base n and returns output.*

**int** valid(**char** arr**[]**, **int** m, **int** l)

{

**char** residue[16] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};

**int** i, j, flag = 0;

    for(i = 0; i < l; i++)

    {

        for(j = 0; j < m; j++)

        {

            if(arr[i] == residue[j])

            {

                flag = 1;

                break;

            }

        }

        if(flag == 0)

        {

            return flag;

        }

        if (i != l-1)

        {

            flag = 0;

        }

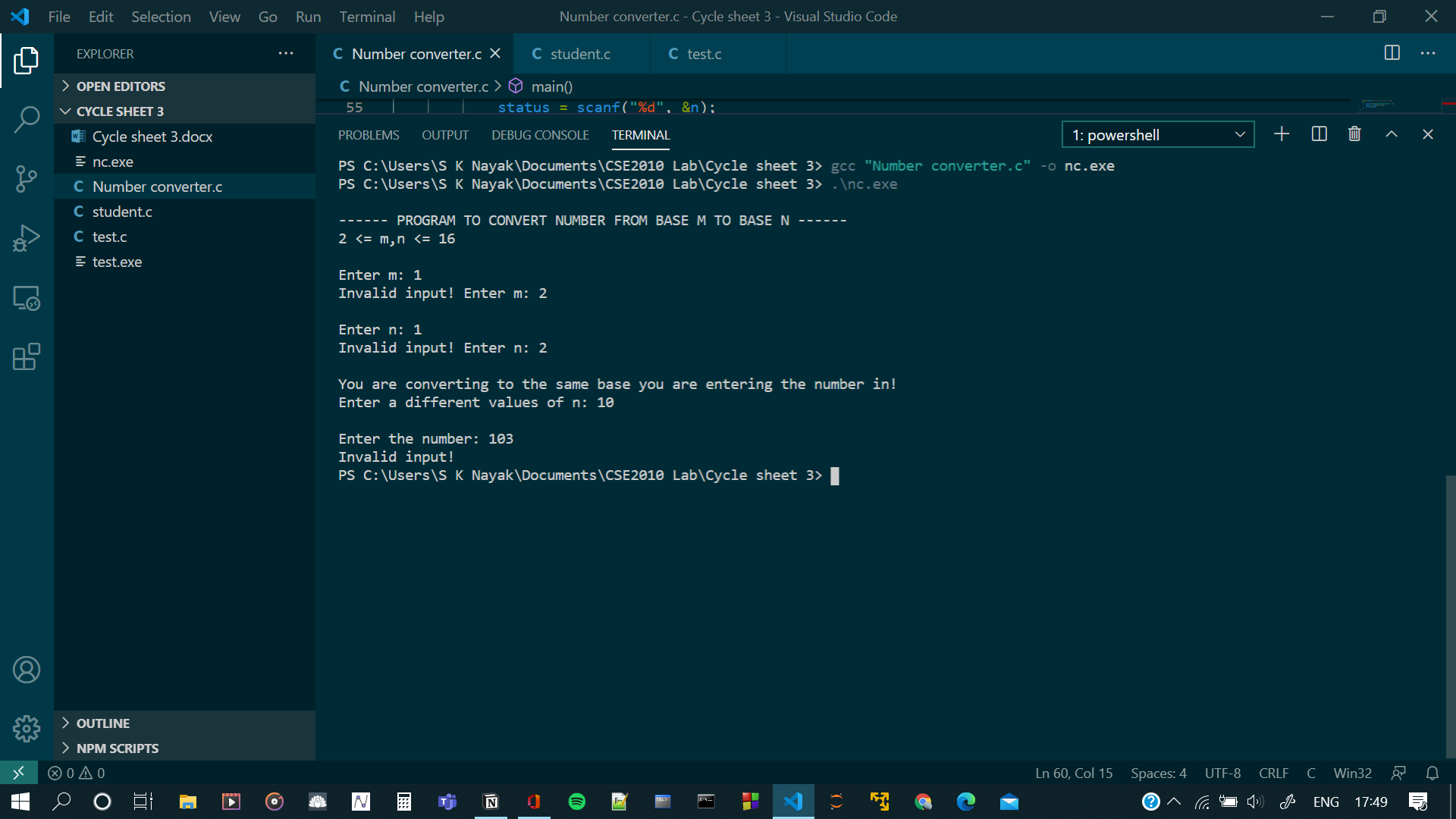
    }

    return flag;

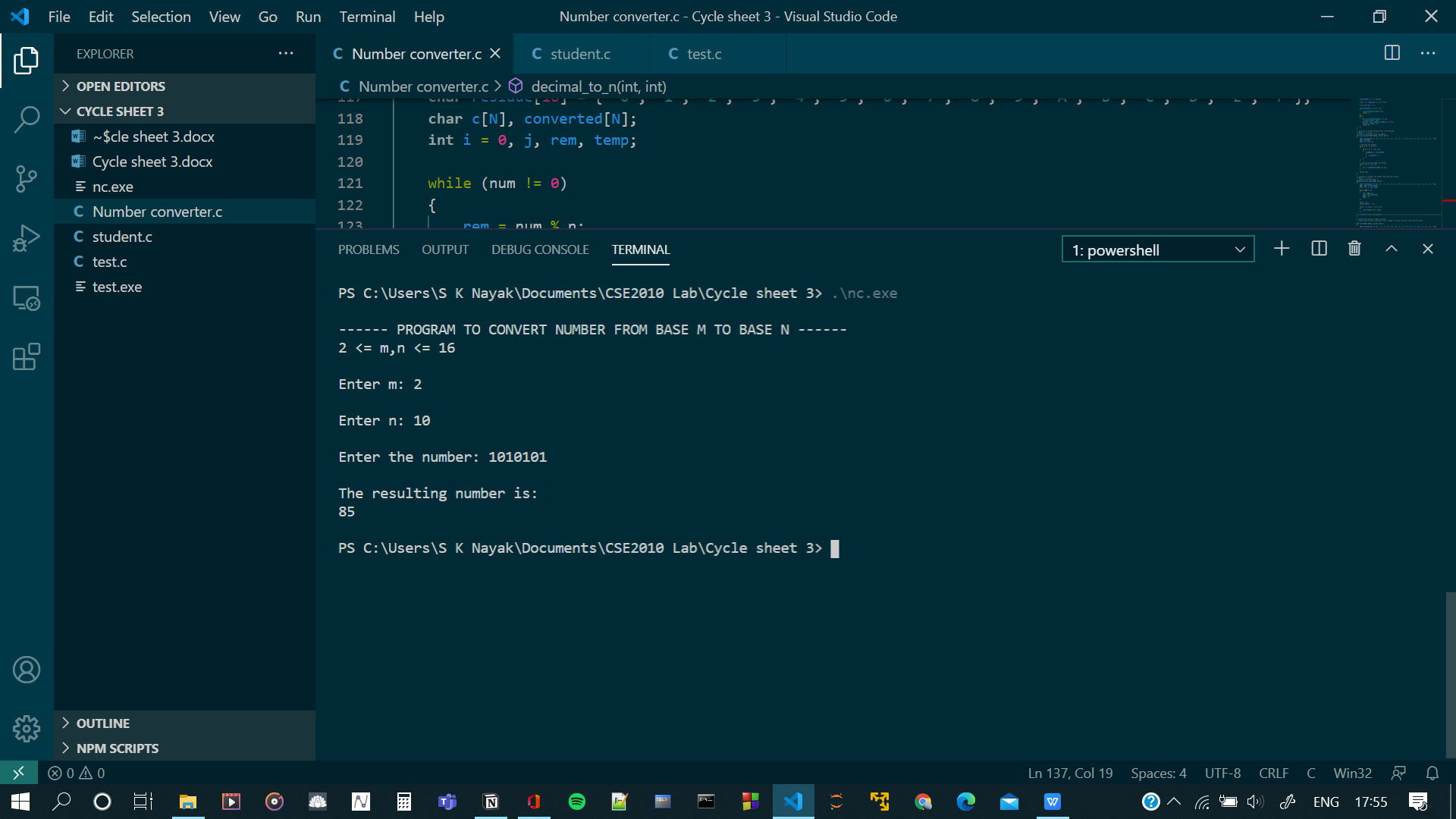
}

**Output:**

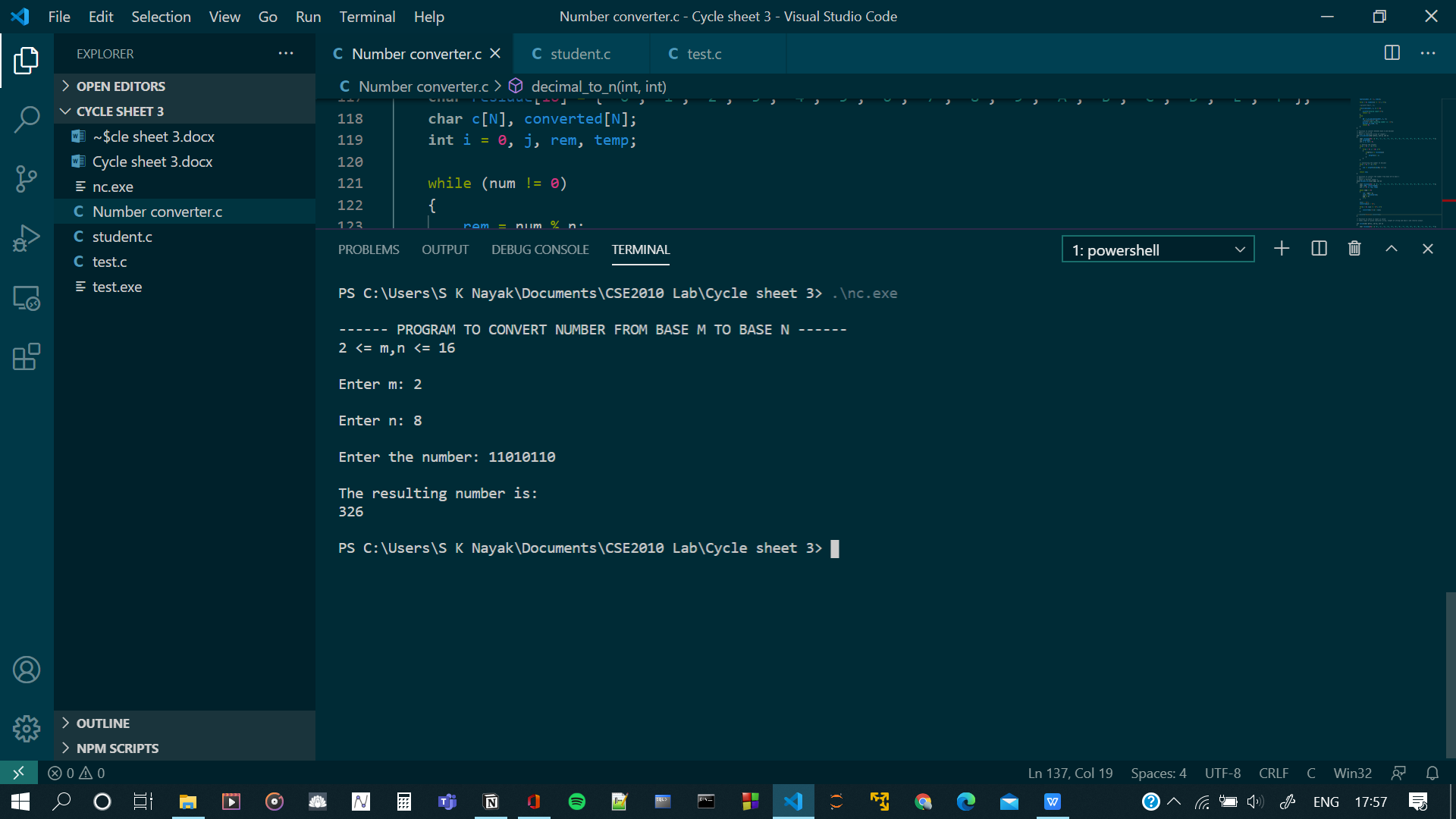
**Input validation :-**



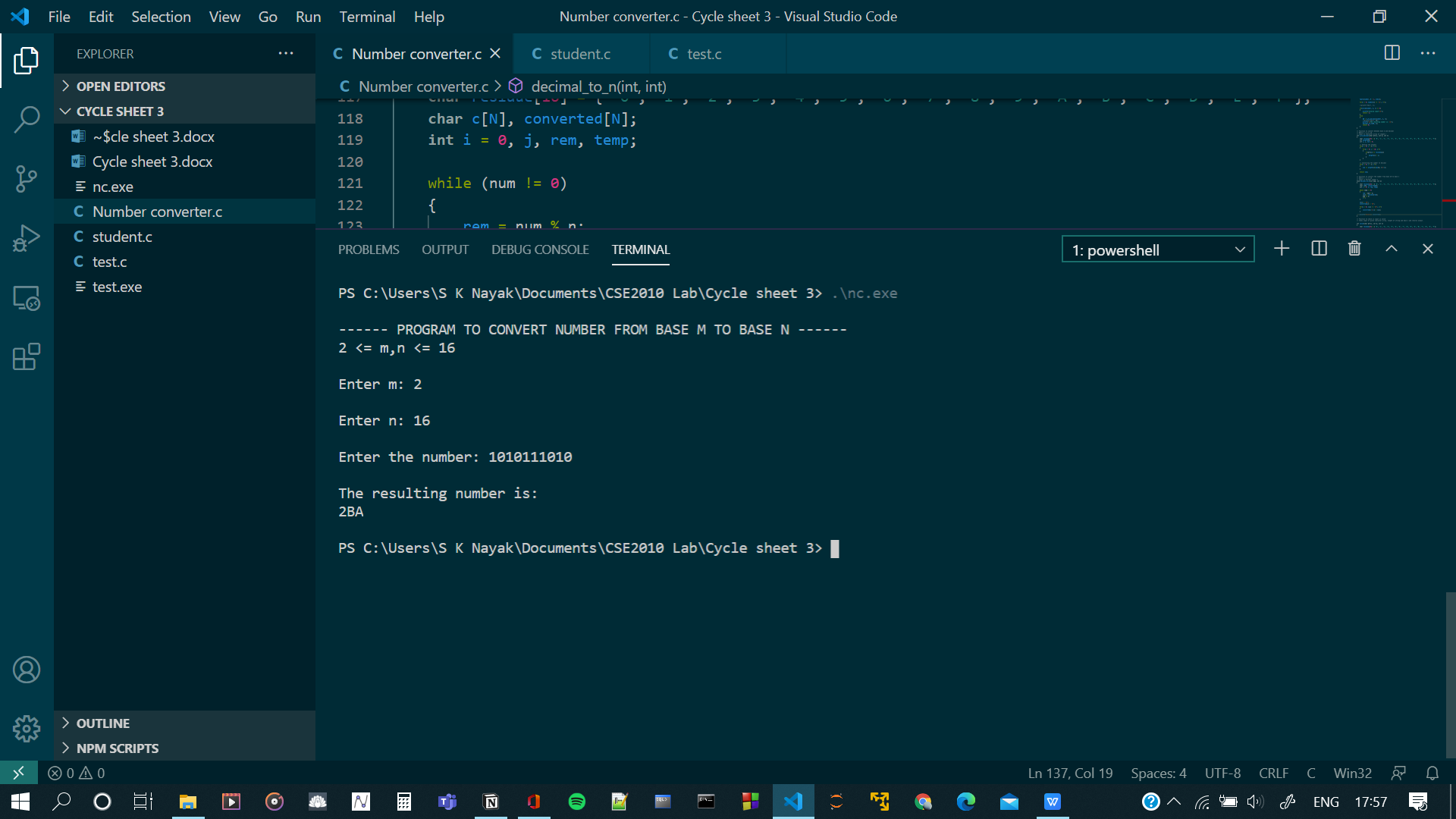
1. Binary to Decimal



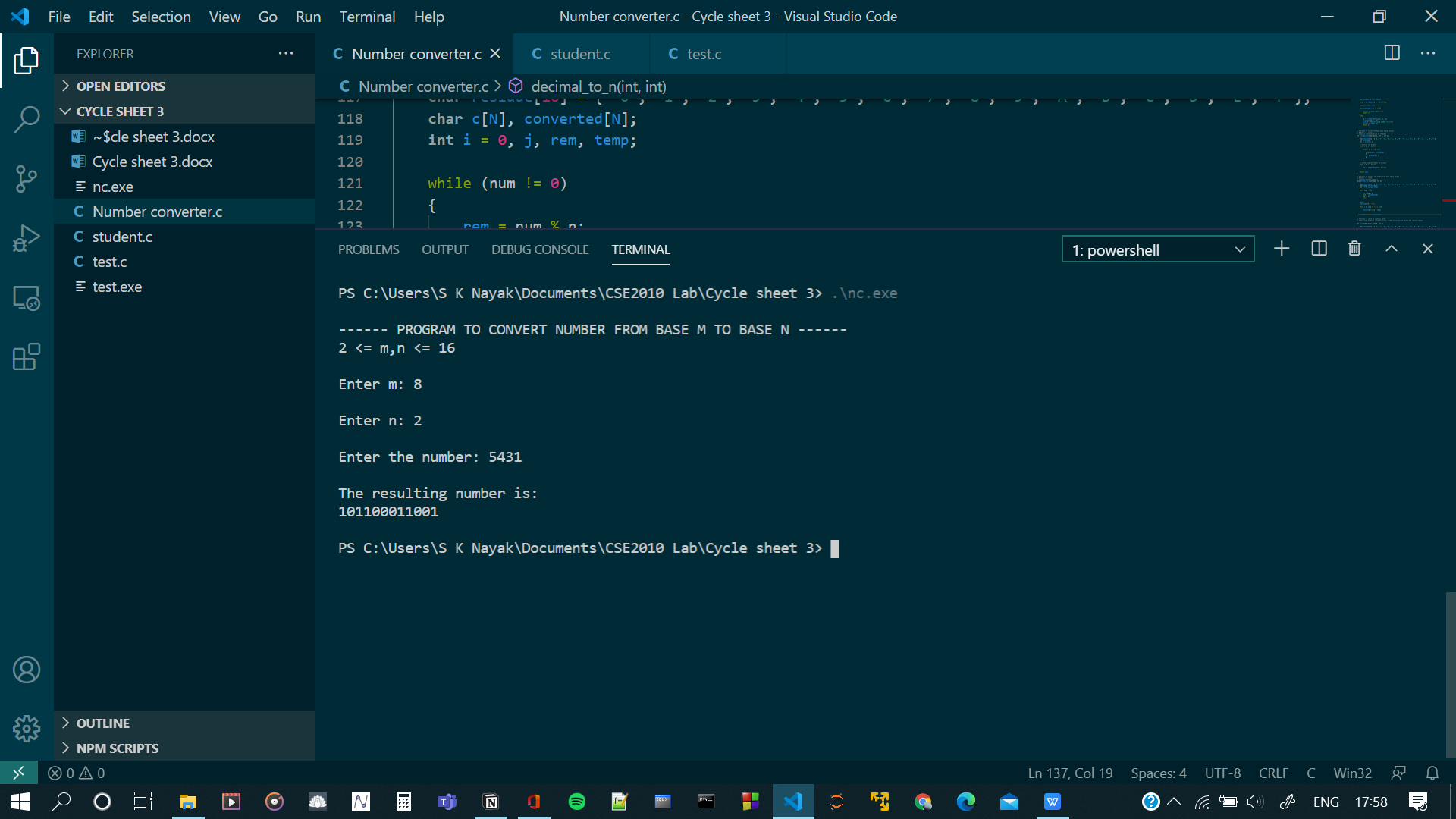
1. Binary to Octal



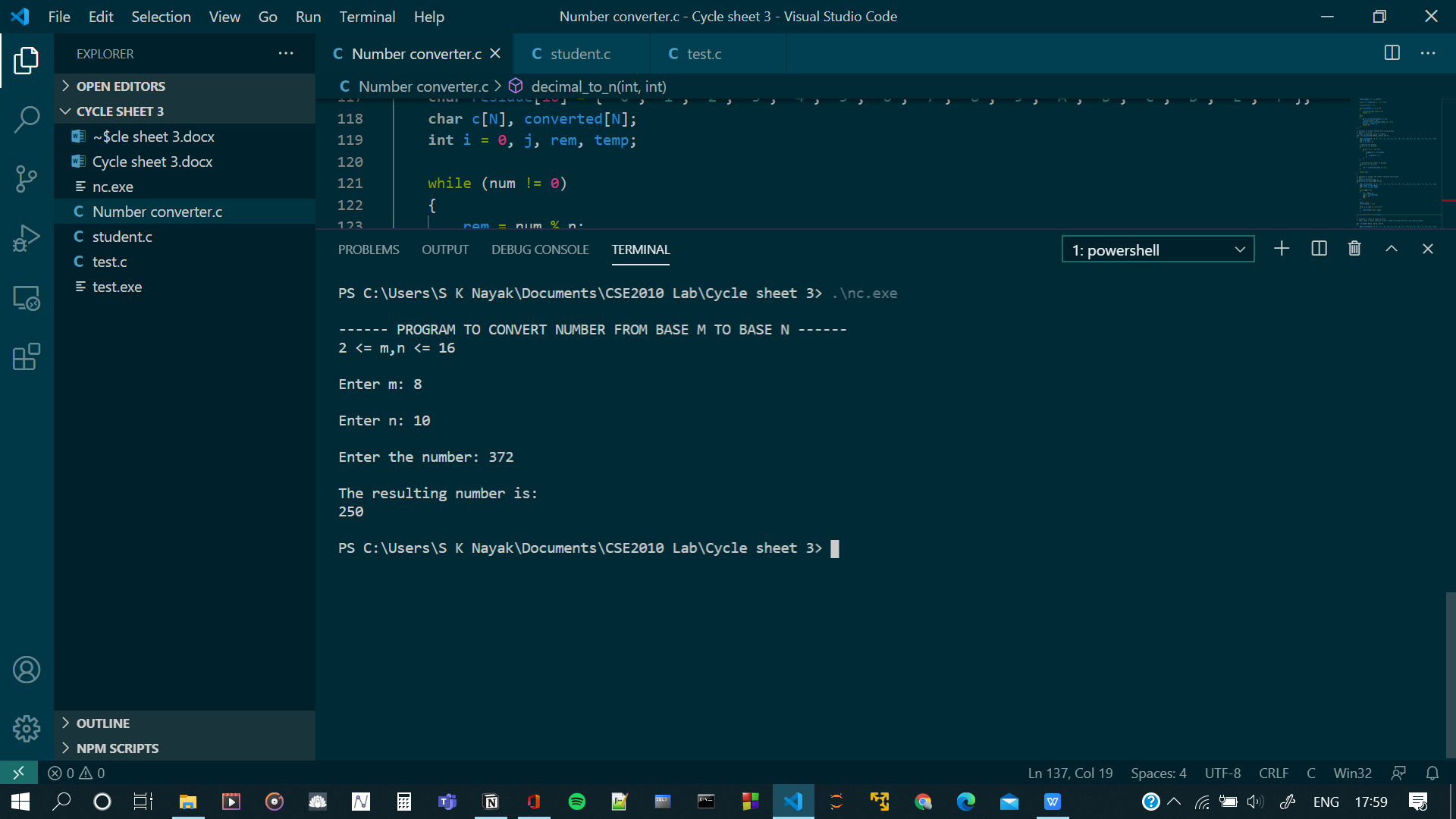
1. Binary to Hexadecimal



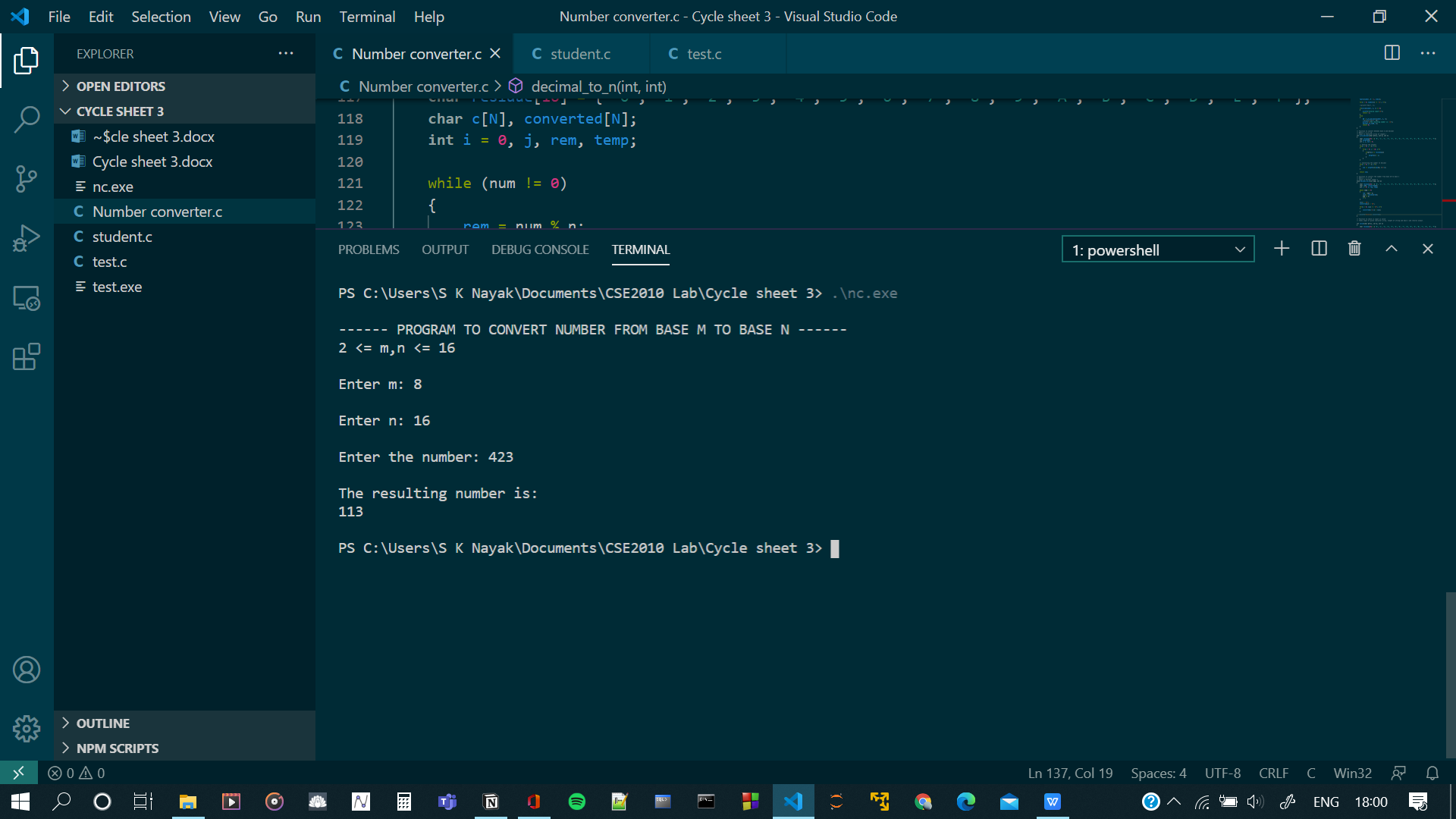
1. Octal to Binary



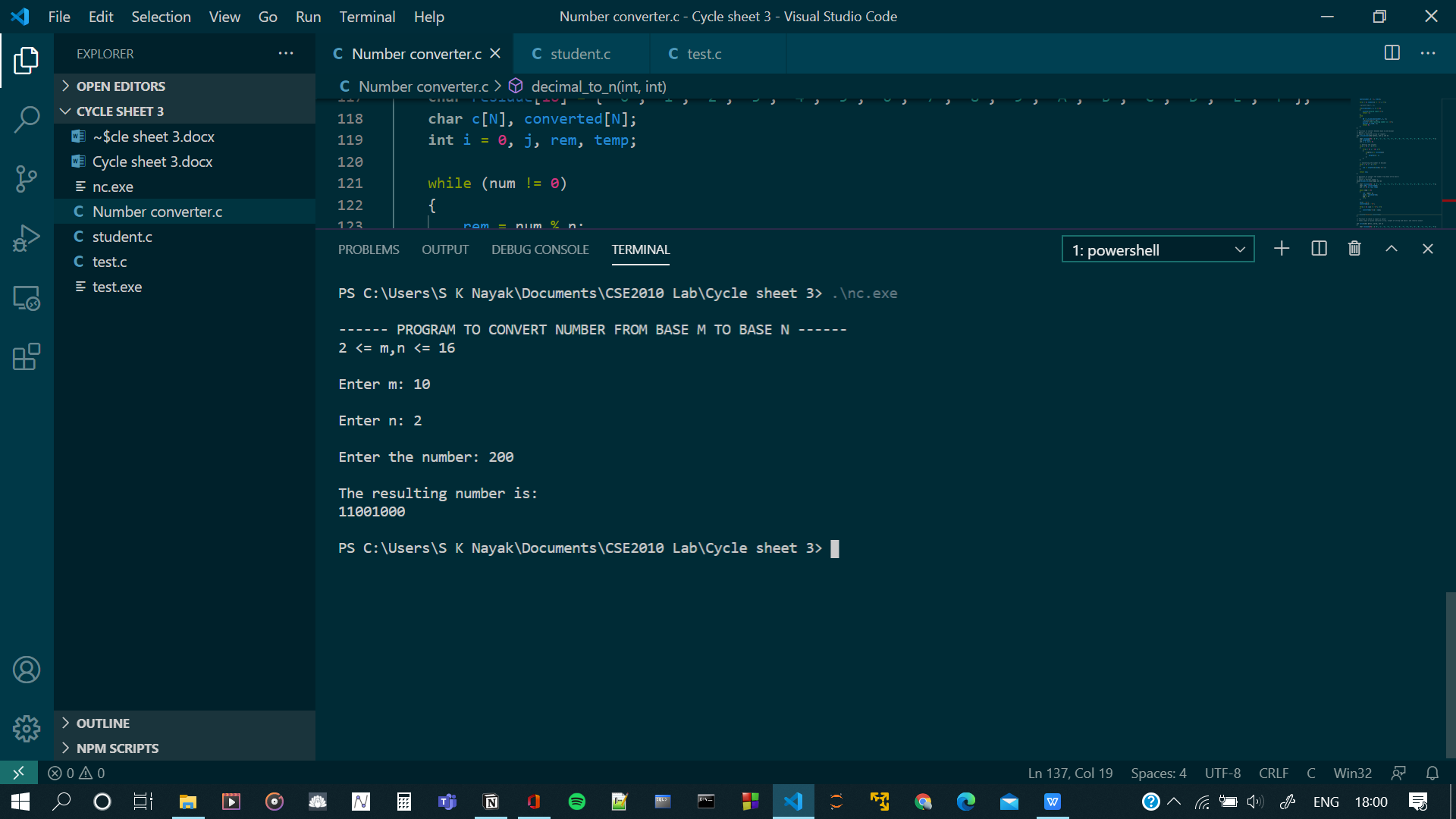
1. Octal to Decimal



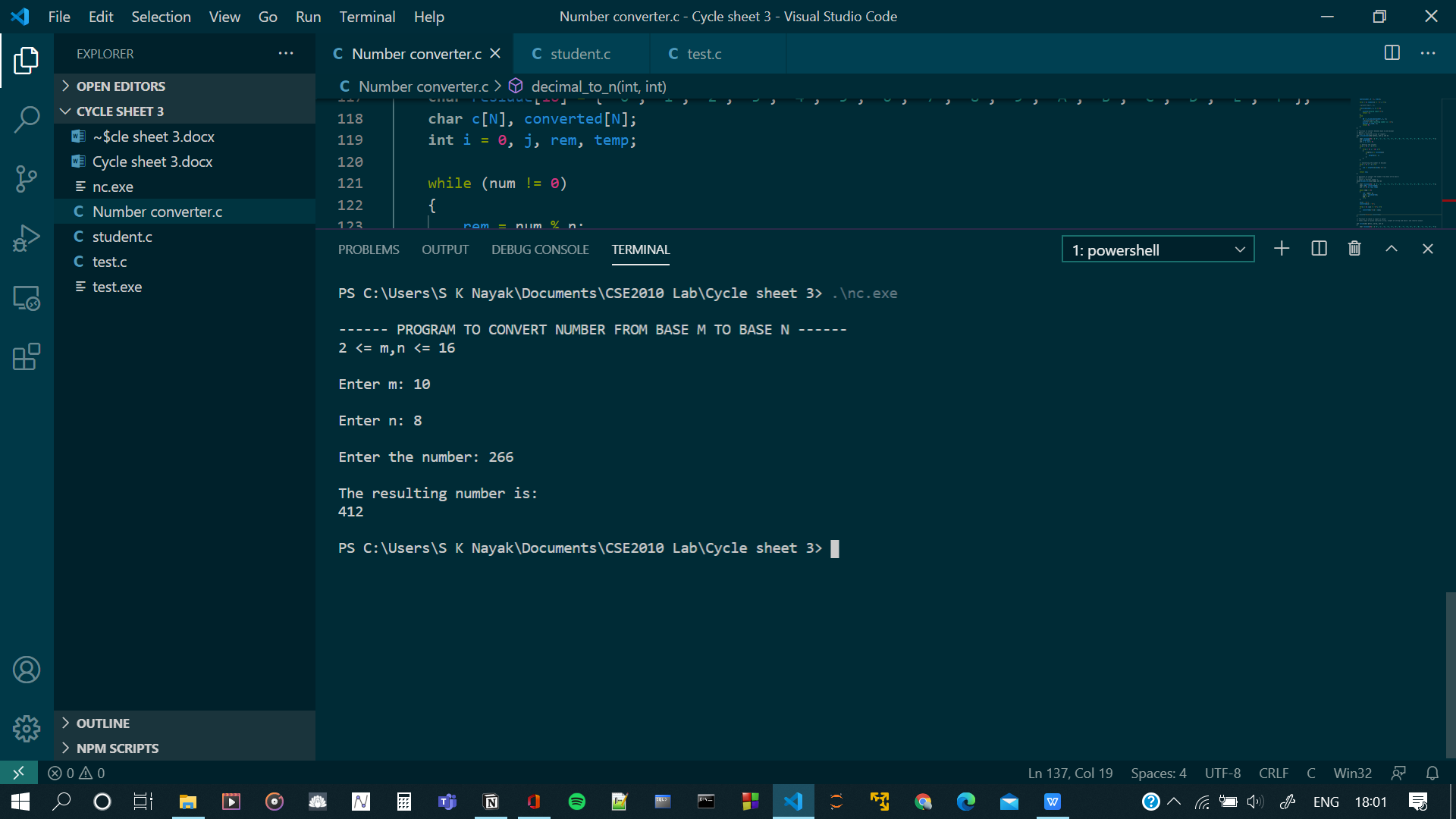
1. Octal to Hexadecimal



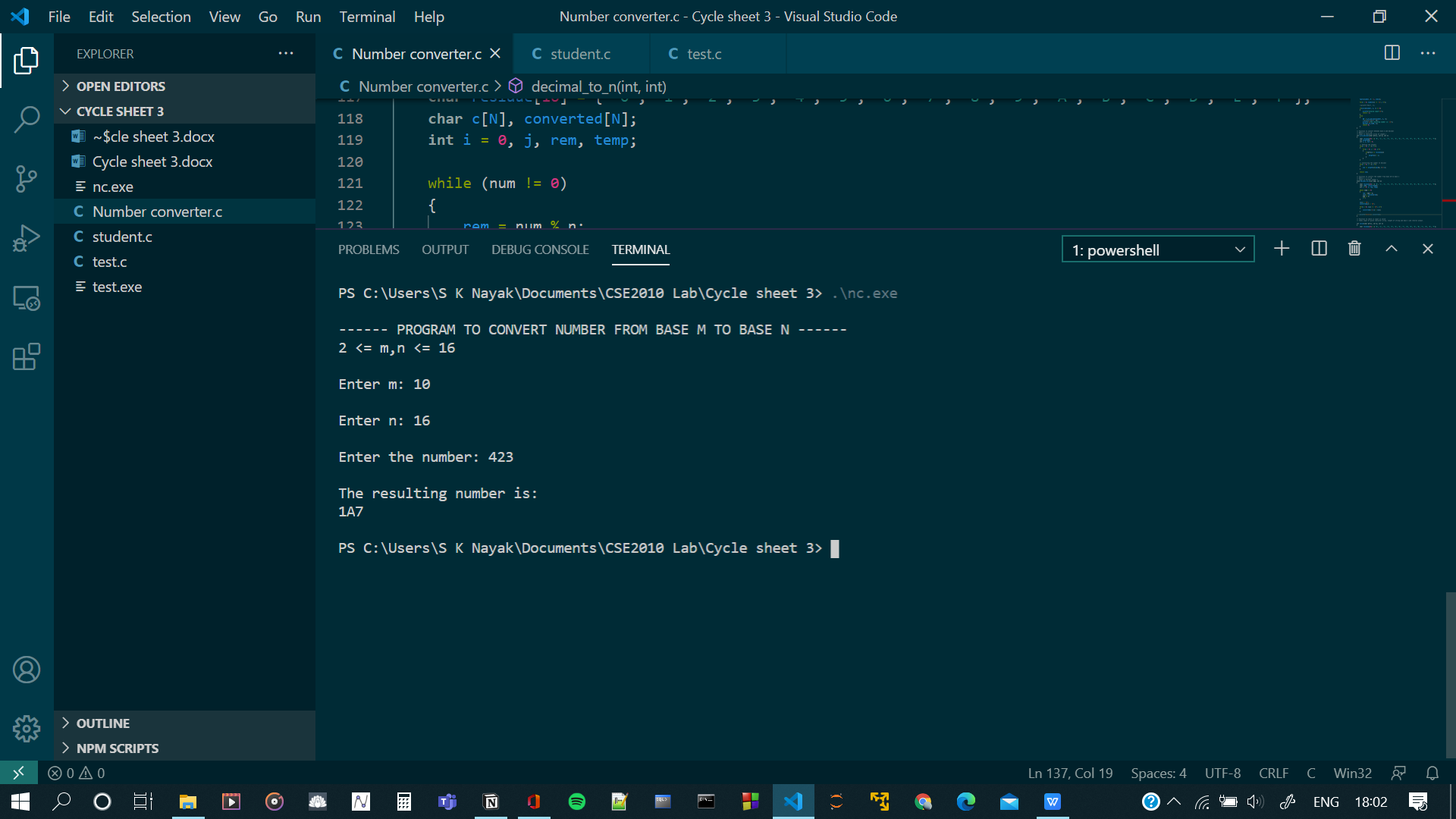
1. Decimal to Binary



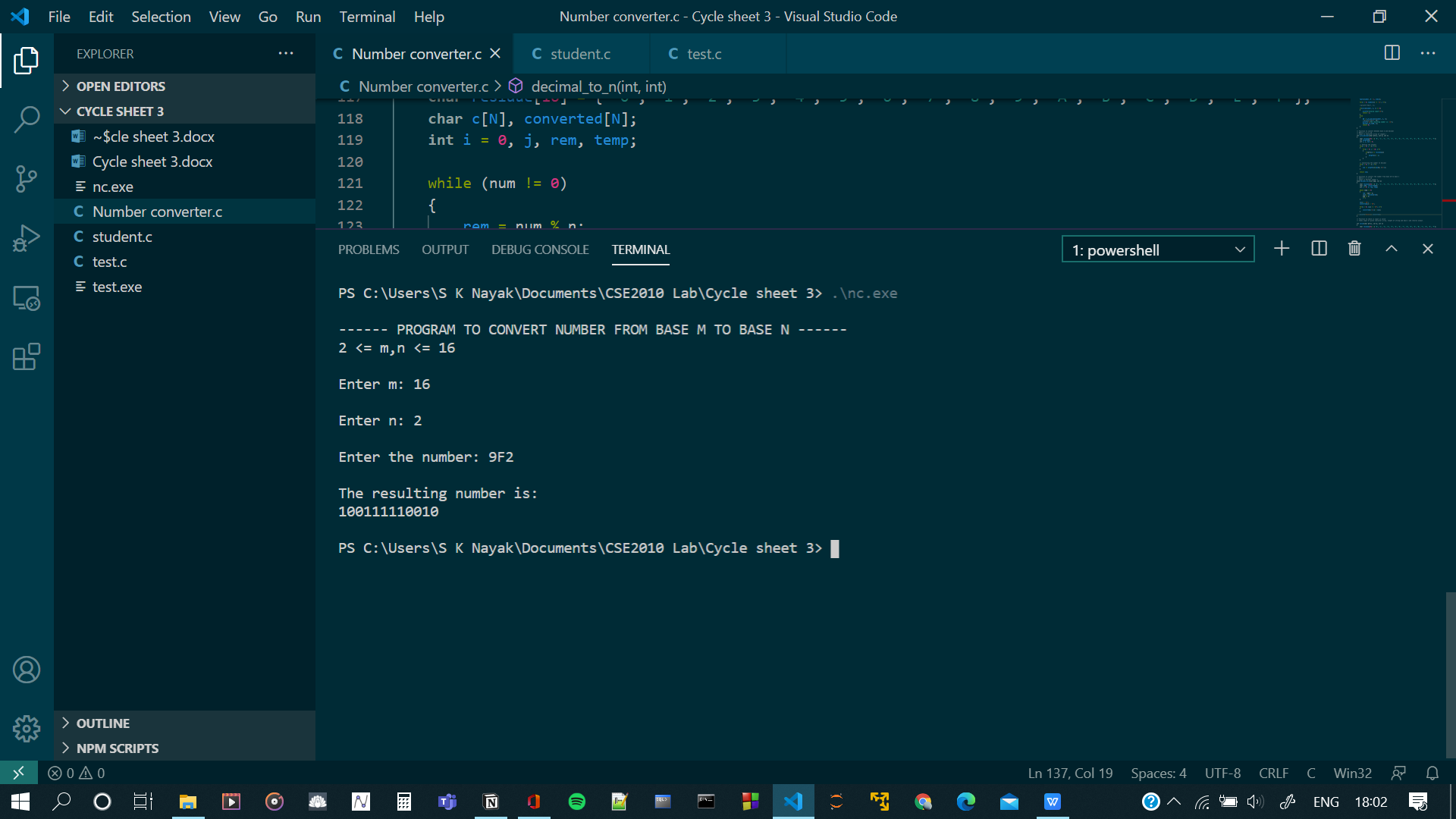
1. Decimal to Octal



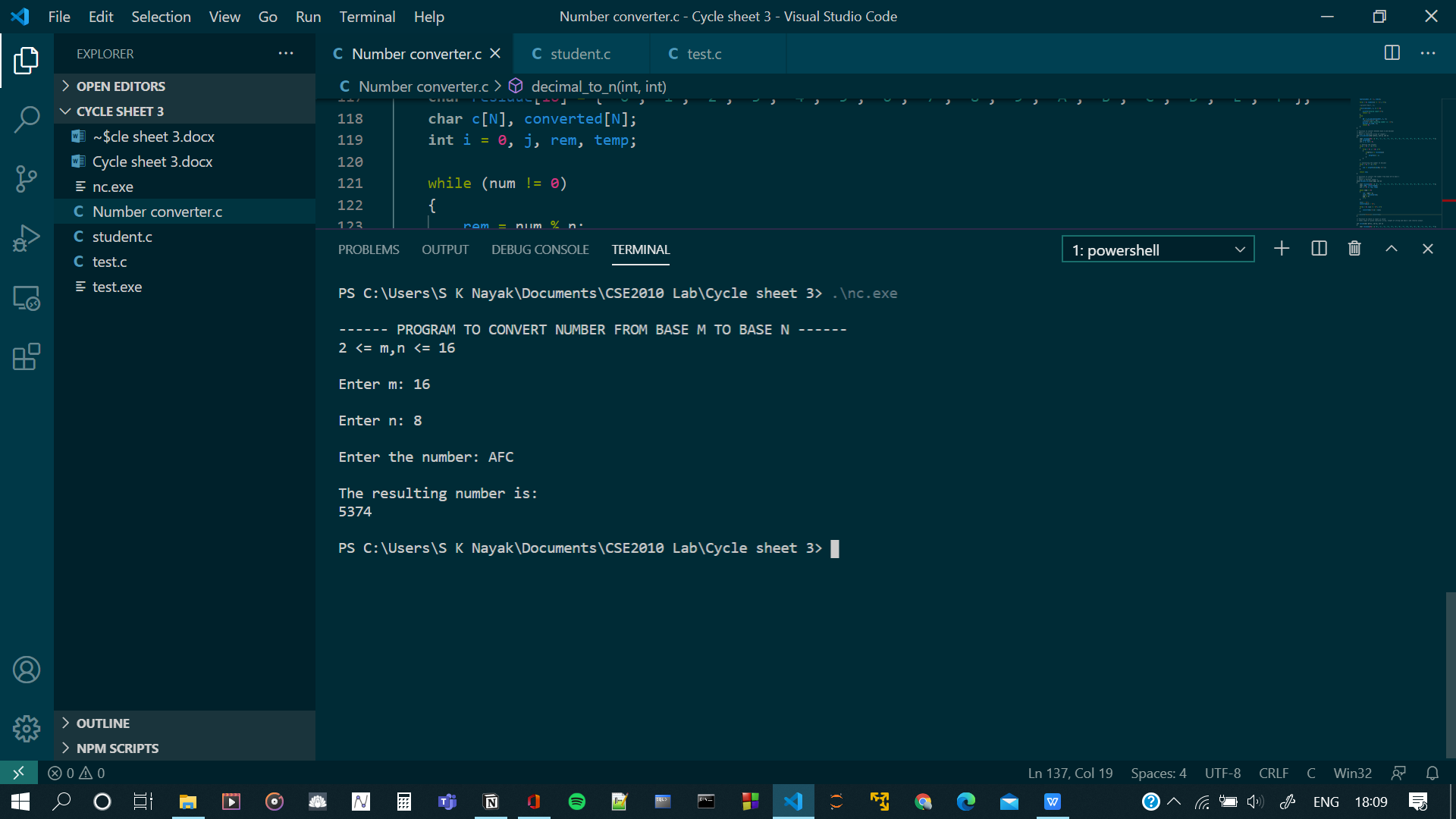
1. Decimal to Hexadecimal



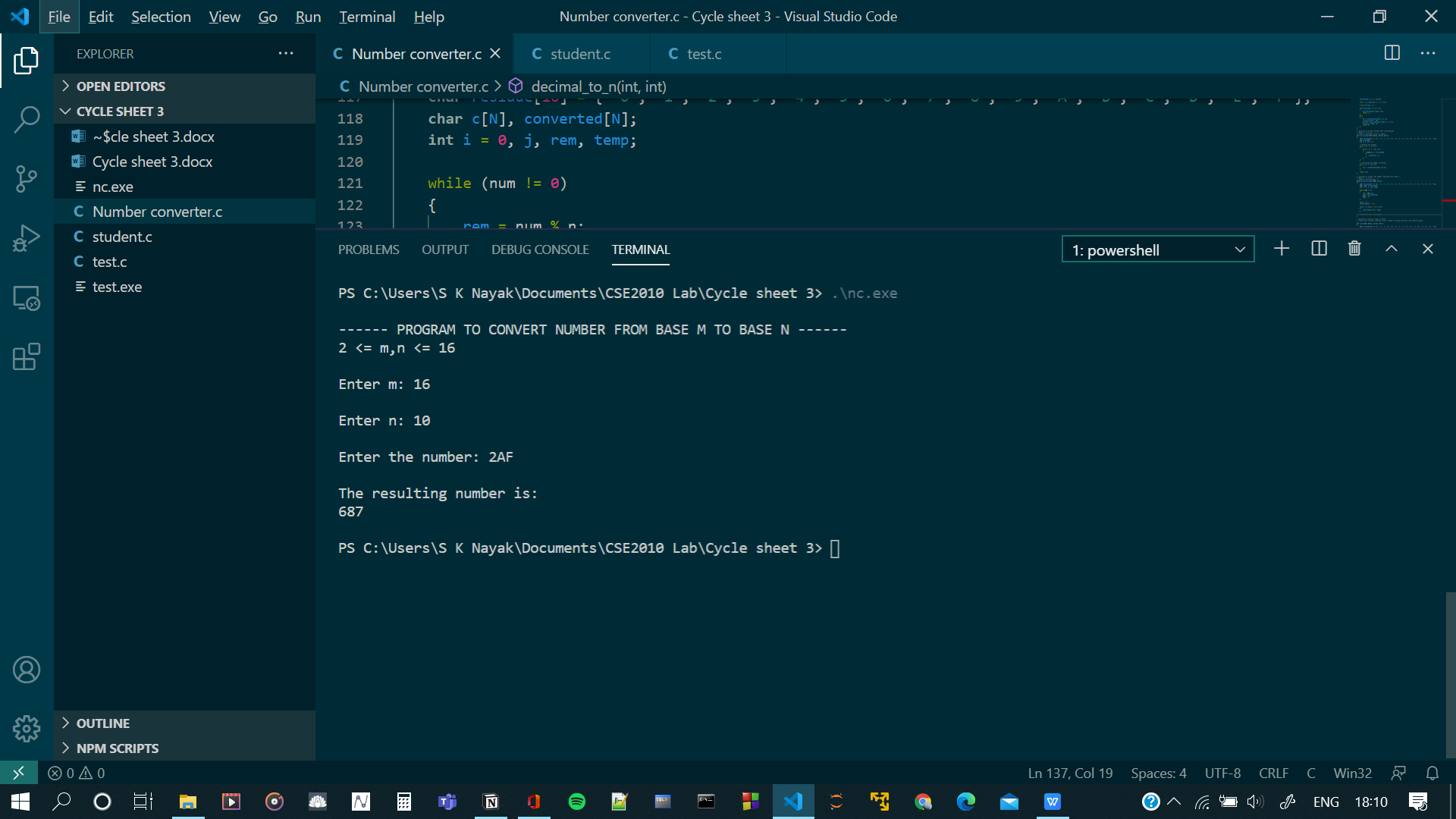
1. Hexadecimal to Binary



1. Hexadecimal to Octal



1. Hexadecimal to Decimal



1. General Conversions (base less than 16)

