

```
/* Coding Problems (25 points) */
```

```
/*
```

Please write code for the two problems below. Write the code for both problems in functions below the main function and call both from the main function. The functions should accept data from main, calculate and return the result to main. Getting data from the user and printing the results should be done in main. When completed, the main function should get input data from the user, call the functions, receive the results of the function calls, and print their results to screen.

1. An integer n is divisible by 9 if the sum of its digits is divisible by 9. Develop a function to display each digit, starting with the rightmost digit. Your program should also determine whether or not the number is divisible by 9. Write a function that calculates the sum of the digits in an int, checks to see if the sum is evenly divisible by 9 and returns 1 for yes and 0 for no. Test it on the following numbers:

```
n = 154368
n = 621594
n = 123456
```

Hint: Use the % and 10 operator to get the least significant digit and use / and 10 to remove the least significant digit. Sample output:

```
Run 1:
IS IT DIVISIBLE BY 9
Enter an integer: 154368
154368 is divisible by 9
```

```
Run 2:
IS IT DIVISIBLE BY 9
Enter an integer: 123456
123456 is NOT divisible by 9
```

2. The value for PI can be determined by the series equation:

$$PI = 4(1/1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 \dots)$$

Write a function using a loop to approximate the value of PI using the formula given including terms up through 1/99, 1/999 and 1/9999. As the number of iterations increase, the estimate gets closer to the value of PI. The function should accept the number of iterations and return the estimate of PI.

Hint: To flip the sign from + to -, declare a variable and multiply it by -1 inside the loop. Sample output:

```
Run 1:
ESTIMATE PI
Enter number of iterations: 99
Estimated PI is 3.121595
```

```
Run 2:
ESTIMATE PI
```

```
Enter number of iterations: 9999
Estimated PI is 3.141393
```

```
*/

// Preprocessor directives
#include <stdio.h>
#include <math.h>

// Function prototypes
int IsDivisibleByNine(long int UserEnteredNumber);
double GetEstimatedValueOfPI(long int NumberOfIterations);

// Main function
int main()
{
    //First Problem
    long int UserEnteredNumber = 0;
    printf("IS IT DIVISIBLE BY NINE? \n");
    printf("Enter an integer:");
    scanf("%ld", &UserEnteredNumber);
    if (IsDivisibleByNine(UserEnteredNumber) == 1)
    {
        printf("%ld is divisible by nine. \n", UserEnteredNumber);
    }
    else
    {
        printf("%ld is not divisible by nine. \n", UserEnteredNumber);
    }

    //Second Problem
    long int NumberOfIterations = 0;
    double EstimatedPIValue = 0.0;
    printf("ESTIMATE PI \n");
    printf("\n Enter number of iterations: ");
    scanf("%ld", &NumberOfIterations);
    EstimatedPIValue = GetEstimatedValueOfPI(NumberOfIterations);
    printf("\n Estimated PI is: %lf \n", EstimatedPIValue);

    return 0;
}

// Function for Coding Problem 1
int IsDivisibleByNine(long int UserEnteredNumber)
{
    int IsNumDivisibleByNine = 0;
    int SumOfDigits = 0;
    int LeastSiginitificantDigit = 0;

    while (UserEnteredNumber > 0)
    {
        LeastSiginitificantDigit = UserEnteredNumber % 10;
        printf("The current least significant digit is %d. \n",
LeastSiginitificantDigit);
        SumOfDigits += LeastSiginitificantDigit;
        UserEnteredNumber /= 10;
    }
}
```

```

    if ((SumOfDigits > 0) && (SumOfDigits % 9 == 0))
    {
        IsNumDivisibleByNine = 1;
    }

    return IsNumDivisibleByNine;
}

// Function for Coding Problem 2
double GetEstimatedValueOfPI(long int NumberOfIterations)
{
    double ValueOfPI = 0.0;
    int NegateTheValue = -1;
    double IndividualTerm = 0.0;

    for (long int iloop = 1; iloop <= NumberOfIterations; iloop++)
    {
        IndividualTerm = 1.0/((2.0*iloop)-1.0);

        if(iloop%2 == 0)
        {
            IndividualTerm *= NegateTheValue;
        }

        ValueOfPI += IndividualTerm;
    }

    ValueOfPI *= 4.0;

    return ValueOfPI;
}

```