

Section ONE

Do Tracing for the following

1) Sum of Digits

```
int sumOfDigits(int n) {
    // Base case
    if (n == 0) {
        return 0;
    } else {
        // Recursive case
        return n % 10 + sumOfDigits(n / 10);
    }
}

int main() {
    int num = 12345;
    printf("Sum of digits of %d is %d\n", num, sumOfDigits(num));
    return 0;
}
```

2) Factorial Function

```
#include <stdio.h>

int factorial(int n) {
    // Base case
    if (n == 0 || n == 1) {
        return 1;
    } else {
        // Recursive case
        return n * factorial(n - 1);
    }
}

int main() {
    int n = 5;
    printf("Factorial of %d is %d\n", n, factorial(n));
    return 0;
}
```

3) Palindrome checking for a given string

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>

bool isPalindrome(char str[], int start, int end) {
```

```

// Base case
if (start >= end) {
    return true;
} else {
    // Recursive case
    return (str[start] == str[end]) && isPalindrome(str, start + 1, end - 1);
}
}

int main() {
    char word[] = "level";
    if (isPalindrome(word, 0, strlen(word) - 1)) {
        printf("%s is a palindrome.\n", word);
    } else {
        printf("%s is not a palindrome.\n", word);
    }
    return 0;
}

```

4) Towers of Hanoi

```
#include <stdio.h>
```

```

void towerOfHanoi(int n, char source, char auxiliary, char destination) {
    // Base case
    if (n == 1) {
        printf("Move disk 1 from %c to %c\n", source, destination);
        return;
    }

    // Recursive case
    towerOfHanoi(n - 1, source, destination, auxiliary);
    printf("Move disk %d from %c to %c\n", n, source, destination);
    towerOfHanoi(n - 1, auxiliary, source, destination);
}

int main() {
    int numDisks = 3;
    towerOfHanoi(numDisks, 'A', 'B', 'C');
    return 0;
}

```

5) GCD

```
#include <stdio.h>
```

```

// Function to find GCD using recursion
int gcdRecursive(int a, int b) {
    // Base case

```

```

    if (b == 0) {
        return a;
    } else {
        // Recursive case
        return gcdRecursive(b, a % b);
    }
}

int main() {
    int num1 = 48, num2 = 18;

    printf("GCD of %d and %d is %d\n", num1, num2, gcdRecursive(num1, num2));

    return 0;
}

```

6) Function to find GCD iteratively

```

#include <stdio.h>

//
int gcdIterative(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

int main() {
    int num1 = 48, num2 = 18;

    printf("GCD of %d and %d is %d\n", num1, num2, gcdIterative(num1, num2));

    return 0;
}

```

Section two

- 7) Write a recursive routine to print array elements
- 8) Write a recursive routine to print array elements in reverse
- 9) Write a recursive routine for multiplication of two integers
- 10) Write a recursive routine for division of two integers
- 11) Write a function a) iterative b) recursion for Fibonacci numbers
- 12) Write a program to convert infix to postfix/ prefix
- 13) Write a program to evaluate postfix/ prefix expression
- 14) Write a program for stack operations
- 15) Write a program for queue operations

