Assignment 6

```c
// 1. WAP in C to count the digits of a given number using
recursion.

#include <stdio.h>
int count = 0;

int countDigits(int n){
    if(n!=0){
        n = n/10;
        count = count + 1;
        countDigits(n);
    }
    else{
        return count;
    }
}

int main(){
    int number;
    printf("Enter a number : ");
    scanf("%d", &number);

    printf("Number of digits : %d", countDigits(number));
    return 0;
}
```

Output :-

Enter a number : 125
Number of digits : 3

```c
// 2. WAP in C to print the array elements using
recursion.

#include <stdio.h>
int index = 0;

int arrayValueThrower(int array[]){
    printf("%d\n", array[index]);
    index += 1;
    if(index<5){
        arrayValueThrower(array);
    }
    else{
        return 0;
    }
}

int main(){
    int arr[5] = {1,2,3,4,5};
    arrayValueThrower(arr);
    return 0;
}
```

Output :-

1
2
3
4
5

```c
// 3. WAP in C to print the fibonacci series using recursion.

#include <stdio.h>
int new = 0;

int printTerm(int a, int b, int index){
    new = a+b;
    printf(" %d", new);
    a=b;
    b=new;
    if(index>1){
        index -= 1;
        printTerm(a, b, index);
    }
    else{
        return 0;
    }

}

int main(){
    int terms;

    printf("No. of terms : ");
    scanf("%d", &terms);

    if(terms>=3){
        printf("0 1");
        printTerm(0, 1, terms-2);
    }

    return 0;
}
```

Output :-

No. of terms : 10
0 1 1 2 3 5 8 13 21 34

```c
/*
WAP to print the following pattern using indirect recursion.

            * * * * * * * * * *
            # # # # # # # # # #

Print the row of star and the row of the hash using two separate functions.
Establish indirect recursion between these two separate functions.

*/

#include <stdio.h>

void fun1(int n);
void fun2(int n);

void fun2(int n)
{
    if(n==10){printf("\n");}
    if (n ==0){return;}
    else if(n<=10)
    {printf("# ");fun1(n-1);}
    else{printf("* ");fun1(n - 1);}
}

void fun1(int n)
{

    if(n==10){printf("\n");}
    if (n ==0){return;}
    else if(n<=10){printf("# ");fun2(n-1);}
    else{printf("* ");fun2(n - 1);}
}

int main()
{
    fun1(20);
    return 0;
}
```

Output :-

```
* * * * * * * * * *
# # # # # # # # # #
```