DESIGN AND ANALYSIS OF ALGORITHMS FOR POLYNOMIAL PROBLEMS

ASSIGNMENT :-01

1) **Write a program to Print Fibonacci Series using recursion.**

```
#include <stdio.h>

int main()
{
   int i, n, t1 = 0, t2 = 1, nextTerm;

   printf("Enter the number of terms: ");
   scanf("%d", &n);

   printf("Fibonacci Series: ");

   for (i = 1; i <= n; ++i)
 {
      printf("%d, ", t1);
      nextTerm = t1 + t2;
      t1 = t2;
      t2 = nextTerm;
   }

   return 0;
}
```

2) **Write a program to check if the given no is Armstrong or not using a recursive function.**

```
#include <stdio.h>
int main()
 {
   int num, originalNum, remainder, result = 0;
   printf("Enter a three-digit integer: ");
   scanf("%d", &num);
   originalNum = num;

   while (originalNum != 0)
{
            remainder = originalNum % 10;

      result += remainder * remainder * remainder;
      originalNum /= 10;
   }
if (result == num)
     printf("%d is an Armstrong number.", num);
   else
     printf("%d is not an Armstrong number.", num);

   return 0;
}
```

3) Write a program to find the GCD of two numbers using recursive factorization.

```
#include <stdio.h>
int hcf(int n1, int n2);
int main()
 {
   int n1, n2;
   printf("Enter two positive integers: ");
   scanf("%d %d", &n1, &n2);
   printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));
   return 0;
}

int hcf(int n1, int n2)
{
   if (n2 != 0)
      return hcf(n2, n1 % n2);
   else
      return n1;
}
```

4) Write a program to get the largest element of an array.

```
#include <stdio.h>
int main() {
  int n;
  double arr[100];
  printf("Enter the number of elements (1 to 100): ");
  scanf("%d", &n);

  for (int i = 0; i < n; ++i)
{
    printf("Enter number%d: ", i + 1);
    scanf("%lf", &arr[i]);
  }
```

```
   for (int i = 1; i < n; ++i)
{
    if (arr[0] < arr[i])
{
      arr[0] = arr[i];
    }
  }
  printf("Largest element = %.2lf", arr[0]);
  return 0;
}
```

5) Write a program to find the Factorial of a number using recursion.

```
#include<stdio.h>
long int multiplyNumbers(int n);
int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n, multiplyNumbers(n));
    return 0;
}

long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}
```

6) Write a program for to copy one string to another using recursion

```
#include <stdio.h>

void copyString(char *str1, char *str2)
{
  *str2 = *str1;
  if (*str1 == '\0')
{
    return;
  }
  copyString(str1 + 1, str2 + 1);
}

int main()
{
  char str1[] = "Hello, World!";
  char str2[50];

  copyString(str1, str2);

  printf("Copied String: %s", str2);

  return 0;
}
```

7) **Write a program to print the reverse of a string using recursion**

```
    #include <stdio.h>

void reverseString(char str[])
 {
   if (str[0] == '\0')
 {
      return;
   }
   reverseString(&str[1]);
   printf("%c", str[0]);
}

int main()
{
   char str[] = "Hello, World!";
   reverseString(str);
   return 0;
}
```

8) **Write a program to generate all the prime numbers using recursion**

```
#include <stdio.h>

int isPrime(int num, int i)
{
   if (i == 1)
{
      return 1;
   }
else
{
        if (num % i == 0)
{
            return 0;
        }
else
{
            return isPrime(num, i - 1);
        }
    }
}

void generatePrimes(int n)
{
    if (n > 1)
{
        if (isPrime(n, n / 2) == 1
 {
            printf("%d ", n);
        }
        generatePrimes(n - 1);
    }
}

int main()
{
    int limit;
    printf("Enter the limit to generate prime numbers: ");
    scanf("%d", &limit);

    printf("Prime numbers up to %d are: ", limit);
    generatePrimes(limit);

    return 0;
```

```
}
```

9) **Write a program to check if a number is a prime number or not using recursion.**

```
#include <stdio.h>

int isPrime(int num, int i)
{
    if (i == 1)
 {
        return 1;
    }
else
{
        if (num % i == 0)
{
            return 0;
        }
else
{
            return isPrime(num, i - 1);
        }
    }
}

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

    if (num == 1)
 {
    printf("%d is not a prime number.", num);
  }
else
{
    int prime = isPrime(num, num / 2);
    if (prime == 1)
 {
      printf("%d is a prime number.", num);
    }
 else
{
      printf("%d is not a prime number.", num);
    }
  }

  return 0;
}
```

10) Write a program for to check whether a given String is Palindrome or not using recursion

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

int isPalindrome(char str[], int start, int end)
{
  if (start >= end)
 {
    return 1;
  }
  if (str[start] != str[end])
 {
    return 0;
    }
    return isPalindrome(str, start + 1, end - 1);
}

int main()
 {
    char str[100];
    printf("Enter a string: ");
    scanf("%s", str);

    if (isPalindrome(str, 0, strlen(str) - 1))
{
        printf("%s is a palindrome.\n", str);
    }
Else
 {
        printf("%s is not a palindrome.\n", str);
    }

    return 0;
}
```