Department of Computer Science & Engineering

Problem Solving with C Laboratory-UE19CS152

Week-5

Objective: Students will learn to write user-defined functions and recursion.

1) Write a function for factorial using recursion and use it to find C(n, r).

Input:

5 2

Output:

ncr is: 10

2) Read input from the user and write the functions for the following:

i) Armstrong number

ii) Perfect number

Input:

153

Output:

The 153 is an Armstrong number.

The 153 is not a Perfect number.

Input:

28

Output:

The 28 is not an Armstrong number.

The 28 is a Perfect number.

steps to compile and execute:

gcc filename.c -lm //linking to math library

./a.out

3)Write a function to check whether a given number is prime and use that to find the next prime number, greater than a given number.

Input:

Enter a number

4

Output:

Next prime number=5

Input:

Enter a number

113

Output:

Next prime number=127

4) Write a user-defined functions to convert Decimal number to equivalent Binary and Octal number.

Output:

Enter 1: decimaltobinary

Enter 2: decimaltooctal

Enter 3: any key to exit

Enter your choice

1

Enter a Decimal Number: 45

Decimal to Binary Conversion

Equivalent Binary Number is: 101101

Enter 1: decimaltobinary

Enter 2: decimaltooctal

Enter 3: any key to exit

Enter your choice

2

Enter a Decimal Number: 45

Decimal to Octal Conversion

Equivalent Octal Number is: 55

steps to compile and execute:

gcc filename.c -lm //linking to math library

./a.out

5)Write a function to count the digits and find the sum of digits of a given number using recursion.

Input

12345

Output:

The number of digits in the number is : 5

The Sum of digits of 12345 = 15

Practice programs:

1) Write a function to output the following using recursion.

```
 1
 2  3
 4  5  6
 7  8  9  10
11  12  13  14  15
```

2) Write a function to find the Hailstone Sequence of a given number upto 1.

Input a positive integer n:

i) If n is **1** then the sequence ends.

ii) If n is **even** then the next element = n/2

iii) If n is **odd** then the next element= (3 * n) + 1

Repeat step (ii) to (iii) until step (i)