NIT-J CSED

Computer Programming Lab

Lab 4: Functions

Q1. Write a C program to input radius of circle from user and find diameter, circumference and area of the given circle using function.

Q2. WAP to find the sum of the following series from 1 to n using user-defined function.
1!/1 + 2!/2 + 3!/3 + ….. + n!/n. For example, 1!/1+2!/2+3!/3+4!/4+5!/5 = 1+1+2+6+24=34.

[**Hint**: use a separate function to compute the factorial of the corresponding number.]

Q2. WAP that accepts 'n' numbers from keyboard and count how many of those numbers are even numbers using user-defined function. The main() function reads all 'n' numbers one by one and at the same time sending them *by reference* to a called function named '*int count even()*'. The job of the '*count even*' function is to check the number received by it as even or odd. For every even number received, it should increase a count variable and finally send the value of the count variable to the main() function where it gets printed.

Q4. WAP to check whether a number is *prime, Armstrong, perfect number or not* using functions. Kindly check the previous assignments (Assignments I to VI) to find the definitions of prime, Armstrong and perfect number.
The function prototype is as follows:

```
/* Function declarations */
        int Prime(int num);
        int Armstrong(int num);
        int Perfect(int num);
```

Example: Input

```
        Input any number: 11
        Output
        11 is prime number
        11 is not a Armstrong number
        11 is not a perfect number
```

Assignment

1. WAP using user-defined function to find all strong numbers between 1 to n. Use a separate function to compute the factorial of the corresponding number.
   [**Hint**: A number in which the sum of factorial of individual digits is equal to the number is called strong number.
   For example, 145 is a strong number because 145= 1! + 4! + 5!
   Example: Input
   Input lower limit: 1
   Input upper limit: 1000
   Output

Strong numbers between 1 and 1000: 1, 2, 145]

2. Write a function to compute the distance between two points and use it to develop another function that will compute the area of the triangle whose vertices are A(x1, y1), B(x2, y2), and C(x3, y3). Use these functions to develop a function which returns a value 1 if the point (x, y) lines inside the triangle ABC, otherwise a value 0.
3. Write a menu driven program which has following options:
    1. Find the p^{th} power of a number *n* using recursive function
    2. Find the binary equivalent of a number *n* and count all 1's using recursive function.
    3. Find the reverse of the number using recursive function.
    4. Exit

Once a menu item is selected the appropriate action should be taken and once this action is finished, the menu should reappear. Unless the user selects the Exit option the program should continue to work.