CS19003: Programming and Data Structure Lab

Lab 5: Functions and Recursion

Sandip Chakraborty

Announcement: Lab 4 Evaluation

L -- R	Anirban Das - anirbanfuture@gmail.com
S	Bishakh Ghosh - ghoshbishakh@gmail.com
T -- Y	Prasenjit Karmakar - Prasenjitkarmakar52282@gmail.com
A	Subhendu Khatuya - subha.cse143@gmail.com
B -- K	Anurag Satpathy - anurag.satpathy@gmail.com

Submission Requirements

- On top of your code, enter your name and roll numbers within comments.

```
/***********************************
 * Name: Your Name
 * Roll: Your Roll
 **********************************/
```

Programming Practices

- Do not copy the code from others. Any form of plagiarism will lead to zero marks.
- Take help of the Internet to solve the problem.
- If you get a compilation error, try to read the error message and find out the reason behind the error.
- If you do not get the expected output, try to debug your code with extra `printf` statements.

Zero Tolerance to Plagiarism

- **You are supposed to write the entire code yourself**
- Do not copy the code / a block of code from any other sources (Internet, or from your friends)
- We'll take a zero-tolerance policy against plagiarism – if a significant portion of the code is similar between two students, both will be given zero.
- We'll not entertain any alibi like "both of us have copied from the same Internet source", "I had shared my Moodle password to my fried", etc.

Submission Instructions

- Once you are done with a code, save the code in your local machine as A5_1.c and A5_2.c, corresponds to the two problems Problem 1 and Problem 2.

- Upload A5_1.c and A5_2.c in Moodle course page, corresponding to Assignment 5 -- https://moodlecse.iitkgp.ac.in/

- **The deadline for this assignment is EOD (11:59 pm) on 24/05/2022**

Problem - 1 – General Description

- The infinite series expansion of sin(x) is represented as follows (x is in radian, a floating-point number):

$$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - \ldots..$$

Problem - 1 – General Description

- The infinite series expansion of sin(x) is represented as follows (x is in radian, a floating-point number):

$$\sin(x) = \underset{\underset{\text{1st term}}{\uparrow}}{x} - \underset{\underset{\text{2nd term}}{\uparrow}}{x^3/3!} + \underset{\underset{\text{3rd term}}{\uparrow}}{x^5/5!} - \underset{\underset{\text{4th term}}{\uparrow}}{x^7/7!} + x^9/9! - \ldots..$$

Problem 1: Part 1

- Write a C function `readData()` which does the following:
  - Take an integer n as the input from the user
  - The program then keeps on reading floating-point numbers from the user until the input value is within $-1 < x < +1$; this value is assigned to a floating-point variable x
  - Then the program keeps on reading another set of floating-point numbers from the user until the input value is within $-0.5 < a < 0.5$; this value is assigned to a floating-point variable a

Problem 1: Part 2

- Now write the following two functions:
  - `findSinX(x, n)`: This function takes x and n as the input, and compute sin(x) with the value of x up to n terms. Print the value of sin(x)
  - `findSinA(a)`: This function takes a as the input, and compute sin(a) up to k terms such that the result is accurate up to 10^{-4} approximation. This indicates that the value of sin(a) remains same up to the 4 decimal position for k and (k+1) terms. Print the value of sin(a) and k (the number of terms) to achieve that value.

Important notes for Problem 1:

- You need to define the function prototypes accordingly. The three functions will be called from the main function.
- You cannot use <math.h> header
- You cannot write separate functions for power and factorial computation. Do it within the main function itself.
- Assume that x **≠ 0 and a ≠ 0**

Problem 1: Marks Distribution

- Implementation of `readData(...)`: 2
- Implementation of `findSinX(x,  n)`: 2
- Implementation of `findSinA(a)`: 2
- Implementation of `main()` and calling the other three functions suitably with necessary arguments: 1

Problem 2

- Write a recursive C function int `countSubstr(char str[], ...)` to count the number of substrings of a given C string, where the substrings start and end with the same character and have length of more than one (single character substrings are not counted). You can include additional arguments in the function as per your requirement. **The function returns the count value and print the all-possible substrings.**
- As an example, if the input is `csedept`, then the output would be one (only one substring ede that starts and ends with the same character e and having length more than one).

Problem 2 cont.

- The function will take a null-terminated string as the input. So, you should not pass the length of the string. Assume that the maximum length of the string is 20, and the minimum length is 2. You do not need to check these bounds explicitly; just consider that the user follows these bounds while providing the input.
- Write a `main()` function, which takes the input from the user, and call the `countSubstr(…)` function with suitable arguments. Then the main function captures the return value from this function and prints the count value.

Problem 2: Hint

- Design a recursive function to extract all possible substrings of a given string.
  - **Base condition**: The substrings with a single character
  - **Recursion**: You can generate three other substrings from a given substring – by removing the first character, by removing the last character, by removing both the first and the last characters
- For each string of length > 1, check whether the first and the last characters are the same. Think of how you'll increment the count value for this case.

Problem 2: Marks distribution

- Implementation of `countSubstr(…)` function: 6
  - Recursively find out all possible substrings of length > 1: 4
  - Check if a substring starts and ends with the same character, then print it and increment the count: 2
  - Deduct 3 marks if the length of the string is passed as an argument

- Main function: 2
  - Take the input and call the recursive function: 1
  - Print the output (count value): 1