|  |  |  |
| --- | --- | --- |
| **Lab 09** | |  |
| **Topic** | * Recursion * Recursion Application |
| **Objective** | * The basic purpose of this lab is to implement ADT of Recursion and test its applications. |

**Instructions:**

* Indent your code.
* Comment your code.
* Use meaningful variable names.
* Plan your code carefully on a piece of paper before you implement it.
* Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp

# void main() is not allowed. Use int main()

* **You have to work in multiple files. i.e separate .h and .cpp files**

# You are not allowed to use system("pause")

* **You are not allowed to use any built-in functions**

# You are required to follow the naming conventions as follow:

* + **Variables:** firstName; (no underscores allowed)
  + **Function:** getName(); (no underscores allowed)
  + **ClassName:** BankAccount (no underscores allowed)

# Students are required to complete the following tasks in lab timings.

**Task 1**

**Write a C++ program to calculate the power of a number using recursion.**

**Instructions:**

* Your task is to write a C++ function **recursivePower(int base, int exp)** that calculates the power of a number using recursion.
* The function should take two values **‘base’** and ‘**exponent’** as parameters, and then recursively calculates and returns the result.
* You also need to check conditions where exponent should be a positive number.
* Test your implementation by calculating the power of a number.

Example:

Input: **Base = 2, Exponent = 3**

OUTPUT:- **8**

**int recursivePower(int base, int exp)**

**Task 2**

**Write a C++ program to print the fibonacci series from 1 to 𝑛 using recursion.**

A Fibonacci series is the series where first two numbers are 0 and 1, while all other numbers are sum of the previous 2 numbers. Such as, **0 1 1 2 3 5 8 13.**

**Instructions:**

* Your task is to write a C++ function **recursiveFibonacci(int n)** that displays the fibonacci numbers upto ‘N’ numbers using recursion.
* The function should take the value of 𝑛 as an input, and then recursively calculate and **display** the series upto 𝑛 numbers.
* Test your implementation by displaying the Fibonacci series.

Example:

Input: **7**

Display fibonacci numbers upto ‘n’ number of times

OUTPUT:- **0 1 1 2 3 5 8**

**void recursiveFibonacci(int n)**

**Task 3**

**Implement binary search using recursion to find an element in a sorted array.**

**Instructions:**

* Write a C++ function **recursiveBinarySearch(int arr[], int start, int end, int key)** that takes a sorted integer **array**, starting and ending index **‘start’**, **‘end’** and the element from array **‘key’** that you need to find.
* The function should return the position or index for the required element from the given sorted array.
* Test your implementation by applying binary search using recursion.

**int recursiveBinarySearch(int arr[], int start, int end, int key)**

**Task 4**

**Implement the recursive function to check if an integer array is a palindrome or not.**

**Instructions:**

* Write a C++ function **recursiveCheckPalindrome(int arr[], int start, int end)** that takes an integer **array,** its starting and ending index **‘start’** and **‘end’** as parameters using recursion.
* Test your implementation by applying palindrome check using recursion.

**Example:**

Input:  **2 5 3 6 3 5 2 OR 4 6 3 3 6 4**

Output: Palindrome

Input: **2 5 3 6 5 2 OR 4 4 9 5 8**

Output: Not a Palindrome

**bool recursiveCheckPalindrome(int arr[], int start, int end)**

**Now create menu based program to perform the following operations**

1. Press 1 to recursively calculate power of a number. **int recursivePower(base, exp)**
2. Press 2 to recursively display Fibonacci series upto n number of times. **void recursiveFibonacci(n)**
3. Press 3 to recursively find an element using binary search. **int recursiveBinarySearch(arr[], start, end, key)**
4. Press 4 to check palindrome recursively. **bool recursiveCheckPalindrome(arr[], start, end)**
5. Press 0 to exit.