CS-110: Lab 11

Recursion

https://github.com/mmujtaba25/cs-110

Muhammad Mujtaba

CMD ID: 540040

mmujtaba.bese25seecs@seecs.edu.pk

Class: BESE 16B

Batch: 2k25

Task 1 [CLO 2]:

CODE:

```cpp
#include <iostream>

int sumArray(const int arr[], int size)
{
    // return if first element reached
    if (size == 0)
        return 0;

    // add array from last element to first
    return arr[size - 1] + sumArray(arr, size - 1);
}

int countOccurrences(const int arr[], int size, int element)
{
    // return if first element reached
    if (size == 0)
        return 0;

    int occurrence = (arr[size - 1] == element) ? 1 : 0;
    return occurrence + countOccurrences(arr, size - 1, element);
}

int main()
{
    constexpr size_t SIZE = 10;
    int arr[SIZE]{0};

    std::cout << "Enter " << SIZE << " integers:\n";

    for (size_t i = 0; i < SIZE; ++i)
    {
        std::cin >> arr[i];
    }

    int element;
    std::cout << "Enter an integer to count its occurrences: ";
    std::cin >> element;

    std::cout << "Sum of array elements: " << sumArray(arr, SIZE) << "\n";
    std::cout << "Occurrences of " << element << ": " << countOccurrences(arr, SIZE, element)
<< "\n";

    return 0;
}
```

OUTPUT:

```
obscure@Obscures-MacBook-Air output % ./task1
Enter 10 integers:
1 2 3 4 4 5 6 7 8 9
Enter an integer to count its occurrences: 4
Sum of array elements: 49
Occurrences of 4: 2
obscure@Obscures-MacBook-Air output %
```

Task 2 [CLO 2]:

CODE:

```cpp
#include <iostream>

int sumDigits(int number)
{
    // return if no digits left
    if (number == 0)
        return 0;

    // add last digit; truncate last digit using int casting; call recursively
    return (number % 10) + sumDigits(number / 10);
}

int main()
{
    int number;
    std::cout << "Enter a non-negative integer: ";
    std::cin >> number;

    std::cout << "Sum of digits: " << sumDigits(number) << "\n";

    return 0;
}
```

OUTPUT:

```
obscure@Obscures-MacBook-Air output % ./task2
Enter a non-negative integer: 1111
Sum of digits: 4
obscure@Obscures-MacBook-Air output % ./task2
Enter a non-negative integer: 1001
Sum of digits: 2
obscure@Obscures-MacBook-Air output % ./task2
Enter a non-negative integer: 1234
Sum of digits: 10
obscure@Obscures-MacBook-Air output % ./task2
Enter a non-negative integer: 5050
Sum of digits: 10
obscure@Obscures-MacBook-Air output % ./task2
Enter a non-negative integer: 5555
Sum of digits: 20
obscure@Obscures-MacBook-Air output %
```

Task 3 [CLO 3]:

CODE:

```cpp
#include <iostream>
#include <string>

bool checkPalindrome(const std::string &str, int left, int right)
{
    // check if first and last is same
    // repeat for next pair

    // passed middle
    if (left >= right)
        return true;

    // at any point if not same, return false
    if (str[left] != str[right])
        return false;

    // move towards the middle
    return checkPalindrome(str, left + 1, right - 1);
}

int main()
{
    std::string input;
    std::cout << "Enter a string: ";
    std::getline(std::cin, input);

    if (checkPalindrome(input, 0, input.length() - 1))
        std::cout << "The string is a palindrome.\n";
    else
        std::cout << "The string is not a palindrome.\n";

    return 0;
}
```

OUTPUT:

```
obscure@Obscures-MacBook-Air output % ./task3
Enter a string: radar
The string is a palindrome.
obscure@Obscures-MacBook-Air output % ./task3
Enter a string: test
The string is not a palindrome.
obscure@Obscures-MacBook-Air output % ./task3
Enter a string: teset
The string is a palindrome.
obscure@Obscures-MacBook-Air output %
```

Task 4 [CLO 3]:

CODE:

```cpp
#include <iostream>

int uniquePaths(int m, int n)
{
    // base case: if we reach the last row or last column, there's only one path
    if (m == 1 || n == 1)
        return 1;

    // recursive case: sum of paths from the cell to the right and the cell below
    return uniquePaths(m - 1, n) + uniquePaths(m, n - 1);
}

int main()
{
    int m, n;
    std::cout << "Enter grid dimensions (m n): ";
    std::cin >> m >> n;

    int paths = uniquePaths(m, n);
    std::cout << "Number of unique paths: " << paths << "\n";

    return 0;
}
```

OUTPUT:

```
obscure@Obscures-MacBook-Air CS-110 Fundamentals Of Computer Programming % cd Lab\ 11/output
obscure@Obscures-MacBook-Air output % ./task4
Enter grid dimensions (m n): 2 3
Number of unique paths: 3
obscure@Obscures-MacBook-Air output % ./task4
Enter grid dimensions (m n): 3 3
Number of unique paths: 6
obscure@Obscures-MacBook-Air output % ./task4
Enter grid dimensions (m n): 9 9
Number of unique paths: 12870
obscure@Obscures-MacBook-Air output %
```