Laboratory 20

CS-122

Spring 2022

Laboratory 12 Consists of: Four Recursion Problems to Solve

1. Recursive Sum of Numbers
2. Recursive Power Method
3. Make the Count Characters in String recursive algorithm case insensitive
4. Display numbers Ascending

Sum of Numbers

- Design a function that accepts an integer argument and returns the sum of all the integers from 1 up to the number passed as an argument.
- For example, if 50 is passed as an argument, the function will return the sum of 1, 2, 3, 4, . . . 50.
- Use recursion to calculate the sum.
- Call your program: *YourName*_SumOfNum.py
- If you are doing Lab20 synchronously, ask the Instructor to check out your SumOfNum program to receive full credit.
- If you are doing Lab20 asynchronously, then submit *YourName*_SumOfNum.py to Canvas.
- After you have solved this problem, then do the second one.

Recursive Sum Of Numbers
Lab20-1

```
#include <iostream>
using namespace std;
int sumofnums(int);
int main()
{
    int   number;
    cout <<"Enter the number you want to sum up to: ";
    cin >> number;
    cout << "The sum of numbers is: " <<  sumofnums(number) << endl;
    return 0;
}
int  sumofnums(int n)
{
     ????????
     ????????
     ????????
     ????????

}
```

Recursive Power Method
Part 2

- Design a function that uses recursion to raise a number to a power.
- The function should accept two arguments:
  - The number to be raised
  - And the exponent.
- Assume that the exponent is a non-negative integer.
- Call your program: *YourName*_RaiseToPower.py.
- If you are doing Lab20 synchronously, ask the Instructor to check out your RaiseToPower program to receive full credit.
- If you are doing Lab20 asynchronously, then submit *YourName*_ RaiseToPower.py to Canvas.

Recursive Power Method
Lab 20-2

```
#include <iostream>
using namespace std;
int raiseToPower(int, int);
int main()
{
    int   number, n;
    cout << "Enter number to raise to power ";
    cin >>  number;
    cout << "\nEnter the power to raise number to: " ;
    cin >> n;
    cout << "Result is: " << raiseToPower(number,n) << endl;
    return 0;
}
int raiseToPower(int x,int y)
{
    XXXXXXXXXXX
    XXXXXXXXXXX
    XXXXXXXXXXX
    XXXXXXXXXXX
}
```

Count Characters in a String Independent of Case Part 3

- In the Count Characters in a string recursive function, you have seen how you can use recursion to count the number of characters.
- However, a “c” and a “C” are considered to be two separate characters.
- Rewrite countchar.py so that it counts the desired character regardless of whether it is upper case or lower case.
- Call your program: *YourName*_CountChars.cpp.
- If you are doing Lab20 synchronously, ask the Instructor to check out your CountChars program to receive full credit.
- If you are doing Lab20 asynchronously, then submit *YourName*_ CountChars.cpp to Canvas.

Count Characters in String: Lab 20-3

```
#include <iostream>
#include <string>
using namespace std;
int numChars(char, string, int);
int main()
{
  string str = "abcdddef";
  cout << "The letter d appears "
       << numChars('d', str, 0) << " times.\n";
  return 0;
}
int numChars(char search, string str, int subscript)
{
  if (subscript >= str.length())
        return 0;
  else if (str[subscript] == search)
        return 1 + numChars(search, str, subscript+1);
  else
        return numChars(search, str, subscript+1);
}
```

Display Numbers Ascending
Part 4

- If time permits, try Part 4, and see if you can get it working.
- Design a function that asks you to display all the positive integers, starting with 1, up to the number entered at the keyboard.
- The function should use recursion to display all the numbers up to and including the number typed in.
- Call your program: *YourName*_DisplayAscending.cpp.
- If you are doing Lab20 synchronously, ask the Instructor to check out your LargestInList program to receive full credit.
- If you are doing Lab20 asynchronously, then submit *YourName*_ LargestInList.cpp to Canvas.

```
#include <iostream>
using namespace std;
// Function prototype
void print_nums(int);
int main()
{
  int num;
  cout << “How many numbers to display up to?\n";
  cin >> num;
  cout << "The ascending numbers are \n ";
  print_nums(num);
  return 0;
}

void print_nums(int n)
{
          ??????????
          ??????????
          ??????????
}
```

Display Integers, starting at 1 up to the integer entered at the keyboard, using Recursion.