Department of Informatics
Database Technology Group
University of Zurich — Prof. Dr. Anton Dignös

Informatics II
Exercise 2

Mar 1, 2021
Labs date: Week 3

Goal:

- Run recursive functions and trace the recursion process by hand.
- Declare, write and call basic recursive functions in C.
- Declare, write and call mutual and multiple recursive functions in C.
- Write and compare iterative and recursive functions in C.

Recursion

Task 1. Basic Recursion Write the recursive function `int exponent(int x, int pow)` in C that computes the exponent of the base. An input/output example is illustrated below (input is typeset in bold):

```
Please enter the base: 2
Please enter the power: 3
The result is: 8
```

Task 2. Multiple Recursion In a sequence $a(n) = a_1, a_2, ..., a_n$ with $n \geq 3$ integers, the first two integers $a_1 = 1$ and $a_2 = 2$ are fixed. For $i \geq 3$

$$a_i = \begin{cases} a_{i-1}/3 + a_{i-2} & \text{if } a_{i-1} \text{ is divisible by 3} \\ a_{i-1} + a_{i-2} & \text{Otherwise} \end{cases}$$

1. Determine a_3 and a_4, sketch the trace process of your recursive function and calculate how many times your recursive function is called.
2. Write the recursive function **int sequence(int n)** in C that computes and prints all the integers of $a(n)$.

Task 3. Iteration and Recursion Implement the C program that prints the index of the first upper letter of the given string using iteration and recursion, respectively. If there isn't any upper letter, return -1.

1. Implement an **iterative** function `int iterativeFirstUpper(char str[])`.
2. Implement a **recursive** function `int recursiveFirstUpper(char str[], int pos)` where `pos` is the position of a character.

Hint: For a character `ch`, then `ch` is an uppercase character if `(ch >= ‘A’ and ch <= ‘Z’)`.

Task 4. Pascal Triangle Pascal triangle is a useful tool that calculates the coefficients in the expansion of the polynomial $(a+b)^n$. Each element in the pascal triangle is associated with a coordinate (i, j) that is row i and index j of the row i. Both i and j start from 0. In row i, there are $i + 1$ elements. For example, in row 2, there are 3 elements, and 2 is associated with $(2, 1)$. Among the $i + 1$ elements of row i, the first and last elements are both 1, any other element at (i, j) are the sum of the element at $(i − 1, j − 1)$ and the element at $(i − 1, j)$ in the previous row $i − 1$. For example, in row 2, 2 at $(2, 1)$ is the sum of 1 at $(1, 1)$ at row 1 and 1 at $(1, 0)$ at row 1.

```
row 0: 1
row 1: 1 1
row 2: 1 2 1
row 3: 1 3 3 1
row 4: 1 4 6 4 1
row 5: 1 5 10 10 5 1
...
```

1. Write the recursive function `int pascal(int i, int j)` in C that return the value at the position (i, j) in the pascal triangle.
2. Write a function `void printPascal(int n)` that prints the first n rows of the pascal triangle.
3. (Optional) The pascal triangle can be printed in another way, as seen in the below example:

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5   10   10   5   1
...
```

In the above printing format, there should be 3 whitespaces between any two elements in the same row. Meanwhile, at the beginning of nth row, there should be 0 whitespaces, and at the beginning of the $n − 1$th row, there should be 2 whitespaces and so on so forth.

Write a recursive function `int recursive_find_indent(int current_row, int total_rows)` to find the proper whitespaces at the beginning of `current_row`. Then write a function `void format_print_pascal(int n)` that prints the first n rows of the pascal triangle.