University of Zurich^UZH

Department of Informatics
Database Technology Group
Prof. Dr. M. Böhlen

Data Structures and Algorithms
Assignment 1

Feb 22, 2016

Note: Task 1 and Task 2 have to be completed in the first lab with the guidance of the tutors.

Introduction to C [10 points]

Task 1. Given a string of characters, write a program in C that uses the function **void reverse(char word[], char reverseString[])** to *reverse* a string. Your program should first print a prompt asking the user to type a string and then print the reverse of the input string. An input/output example is illustrated below (input is typeset in bold):

Type a string: **Hello**
Reverse string: olleH

Task 2. Consider an array `A[0...n-1]` with `n` integer values and an integer x. Write a program in C that prints proper prompt messages, reads the elements of A, reads x and then prints the number of copies of x in A. An input/output example is illustrated below (input is typeset in bold):

Elements of A separated by spaces (non-number to stop): **1 2 3 2 2 5 end**
Type x: **3**
Copies = 1

Task 3. Consider an array `A[0...n`$_A$`-1]` with `n`$_A$ integers, and an array `B[0...n`$_B$`-1]` with `n`$_B$ integers. Array `A` does not include duplicated elements. Array `B` may contain duplicated elements. Arrays `A` and `B` are both sorted in increasing order. Write a program in C that reads arrays `A` and `B` and prints all pairs of the form `(v,t)`, where `v` corresponds to a value in array `A` and `t` to the number of times it appears in array `B`. [10 points]

Sorting (n^2) [20 points]

Task 4. Before a football game starts, team A and team B shall take a group picture. One team is in the front row and the other team in the back row. To get a nice picture the photographer requires a person in the front row to be shorter than the corresponding person in the back row. Given arrays A and B, both of size n, whose elements are the heights of the team members, check if it is possible to make a nice picture.

Write a program in C that uses the function **int team_photo(int A[], int B[], int size)** to print "TRUE" if it is possible to arrange two teams as described, and print "FALSE" otherwise. Hint: Implement and use a function **void bubblesort(int a[], int size)** to solve the task. [20 points]

Recursion [30 points]

Task 5. Given integers b and e, write a C program that uses the function **int power(int base, int exponent)** to calculate b^e recursively. [4 points]

Task 6. Given two integers a and b, write a C program that uses the function **int gcd(int a, int b)** to calculate the *greatest common divisor (GCD)* of a and b, recursively. For example, gcd(5,7) should return 1, gcd(8,12) should return 4, and gcd(203,203) should return 203.

Hint: The **gcd** can be implemented using Euclid's algorithm, which works as follows:

- If the two numbers are identical, gcd(a,a) = a
- If one number is larger, subtract the smaller number from the larger number, and then compute the GCD of the difference and the smaller number.

For instance, to determine gcd(12,8), compute $12-8=4$ and then solve gcd(4,8). Since 8 is larger than 4, this can be solved by computing gcd(4,4), which is 4. [10 points]

Task 7. The Levy C curve in Figure 1 is a fractal whose initial pattern is a straight line as illustrated in Fig. 1(a). In the first iteration, an isosceles triangle with angles of 45^o, 90^o and 45^o is built using this line as its hypotenuse. The original line is then replaced by the other two sides of this triangle. In each iteration, each straight line is replaced by the other two sides of a right-angled isosceles triangle built on it. After n iterations the curve consists of 2^n line segments, each of which is smaller than the original line by a factor of $2^{\frac{n}{2}}$.

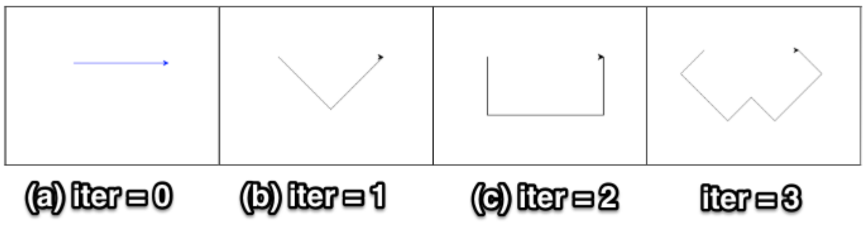

Figure 1: Levy C Curve

Drawing the Levy C curve can be done by drawing each segment separately. Each segment is specified by an (r, ϕ) pair corresponding to the polar coordinates of its ending point (x_1) with respect to its starting one (x_0), which coincides with the ending point of its neighbouring segment on the left. For example, the form of the curve after the second iteration, as shown in Fig. 1(b), can be drawn following these steps: "draw a segment of length $\sqrt{0.5}$ with angle -45, draw a segment of length $\sqrt{0.5}$ with angle 45".

(a) Create the C function **drawLevyCurve(int iter, double r, int phi)**, which uses a recursive scheme to determine the polar coordinates to draw Levy C curve after the i-th iteration. As input it receives the number of remaining iterations **(iter)** and the polar coordinates **(r, phi)** of the ending point of the segment to which they should be applied. [8 points]

   Example: The invocation **drawLevyCurve(1,1,0)** describes the process of applying one iteration **(iter=1)** on the horizontal **(phi=0)** 1cm long **(r=1cm)** line in Fig. 1(a). Its output would be (0.707107,-45), (0.707107,45). Accordingly, an invocation **drawLevyCurve(2,1,0)** should print the pairs needed for the form of the curve presented in Fig. 1(c).

(b) Write a program in C that uses the function drawLevyCurve(int iter, double r, int phi) to draw the curve using OpenGL. [8 points]

   Note: A skeleton of the solution of Task 7(b) is provided.

Submission

Please submit a folder *a<exercise number>_<family name>_<matriculation number>.zip* where `family name` and `matriculation number` correspond to your personal data. The folder should include the C files you created for each of the tasks. Each C file should be named *task<task number>.c* and it should include your personal data in the form of a comment on the top.

The zipped folder should be uploaded in the page of the course in OLAT in the `‘‘Drop box’’` corresponding to Assignment 1 latest on **Sunday, March 6th at 23:59**.

Important Notes

- Include the library `math.h` (#include <math.h>). It will allow you to use the functions `pow` and `sqrt` which are useful in some of the tasks.