Assignment 1 - Software I, Spring 2003 (0368-2157-12)

`http://www.cs.tau.ac.il/~efif/courses/software1`

Due: Mars 11, 2003

Before starting to answer the questions, please read very carefully the "Submission Guidelines"[1]. Make sure your programs detect invalid input data, and print out appropriate error messages. Do not add "friendly" messages to your programs, as they are tested automatically by other programs.

Ex 1.1 d2u

Write a program that reads text in `DOS` format from the standard input until it reaches the end-of-file, and transforms it into `UNIX` format, which is written to standard output.

In a `DOS` text format, at the end of each line there are two characters `'\r'` and `'\n'` in this order, while in `UNIX`, at the end of each line, there is only a single character `'\n'`.

Ex 1.2 u2d

This program reads text in `UNIX` format from the standard input until it reaches the end-of-file, and transforms it into `DOS` format.

To test your programs `d2u`, `u2d`, you can check if they work like the `UNIX` equivalent programs `dos2unix` and `unix2dos`. Take a text file (for example a copy of your program source) and do:

```
dos2unix < dos-file > unix-file
unix2dos < unix-file > dos-file
./d2u < dos-file > my-unix-file
./u2d < unix-file > my-dos-file
diff unix-file my-unix-file
diff dos-file my-dos-file
```

If the results are not identical, your program is erroneous.

Ex 1.3 cos

Write a program that reads a real number x that represents an angle in radians followed by a positive small real number ϵ. It computes $\cos(x)$ according to the formula below, and prints out the result.

$$\cos(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}$$

[1] `http://www.cs.tau.ac.il/~efif/courses/software1`

The computation should stop when the absolute value of the last element summed is smaller than or equal to ϵ. Notice that $0^0 = 1$.

Use `double` type to represent any real number in the program, and print out the result with 6 digits of precision for the fraction part. Example:

```
cos
3.1415927 0.01
-0.999900
```

Ex 1.4 gcd

Write a program that reads two integers a and b, and prints out their *greatest common divisor* $\gcd(a,b)$.

The $\gcd(a,b)$ of two positive integers a and b, is the largest divisor common to a and b. For example, $\gcd(3,5) = 1$, $\gcd(12,60) = 12$, and $\gcd(12,90) = 6$. The greatest common divisor can also be defined for three or more positive integers as the largest divisor shared by all of them.

The Euclidean algorithm, also called Euclid's algorithm, finds the *greatest common divisor*. It relies on the following observation: If $a > b > 0$, then $\gcd(a,b) = \gcd(a \mod b, b)$, where $a \mod b = a - \left\lfloor \frac{a}{b} \right\rfloor b$.

The modulo of 2 integers can be computed in `C` using the `%` operator; that is, $a \mod b =$ `a % b`.

Ex 1.5 sequence

Write a program that reads a sequence of positive integers and prints out the length of the longest strictly increasing sub-sequence of consecutive numbers. The input is a sequence of integers given in one line, separated by spaces. Example:

```
sequence
23 23 2 56 456 500 8 90 81 500 34 100 93 23445 1
4
```

Good Luck!

More Information on the Submission

Files Name

The files for the exercises should be located under `~/software1/assign1`, and their names should match the name of the exercise. For example, the `C` source file and corresponding executable for exercise 1.1, namely d2u, should be `~/software1/assign1/d2u.c` and `~/software1/assign1/d2u` respectively. Note that names are case sensitive (i.e. `ex1.C` is different than `ex1.c`).

Giving Permission to the Files

Before submitting the solution set, please give permission to the files by executing the following command:

```
chmod 705 ~ ~/software1 ~/software1/assign1 ~/software1/assign1/*
```