

Assignment 1 - Software I, Summer 2003 (0368-2157-20)

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

Due: Jul. 30, 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. There are 6 exercises in this assignment.

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.

¹<http://www.cs.tau.ac.il/~efif/courses/software1>

Ex 1.3 pi1

Write a program that computes π according to the formula below.

$$\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots\right)$$

The computation should stop when the absolute value of the next element in the series is smaller than some positive ϵ read as input. The program should print out the computed π followed by the number of elements summed.

Use `double` type to represent any real number in the program. Verify that the input is valid, and print out the result with 6 digits of precision for the fraction part. For example:

```
pi1
0.001
3.139593 500
```

Ex 1.4 pi2

Write a program that computes π according to the alternative formula below.

$$\pi = 2 \times \left(1 + \frac{1}{3} + \frac{1}{3} \cdot \frac{2}{5} + \frac{1}{3} \cdot \frac{2}{5} \cdot \frac{3}{7} + \frac{1}{3} \cdot \frac{2}{5} \cdot \frac{3}{7} \cdot \frac{4}{9} + \dots\right)$$

Compute the $(i + 1)$ -th element in the sequence based on the i -th for efficiency, and use the same instructions as in the previous exercise. For example:

```
pi2
0.001
3.139470 9
```

Ex 1.5 average

Write a program that calculates the floating point average of n integer numbers, and computes the smallest and largest numbers among the n input numbers. The program should read a non-negative integer number, say n , followed by n integers. It should print out the average, the smallest, and the largest numbers before it ends.

Use `double` type to represent the average, and print it out with 2 digits of precision for the fraction part. For example:

```
average
2 1 2
1.50 1 2
```

Ex 1.6 ring

Write a program that prints out a “disturbed” ring made of the asterisk character (*). The program reads the ring radius as input, and considers each line height and each character width as 1 unit length. It must print exactly 1 asterisk at the top and bottom lines for each pole, and 2 asterisks at each line in between, one on each side. The asterisk must be printed at the integer grid point of distance from the center not smaller than the radius. The radius must be in the closed range [0,39].

Examples:

```
ring
```

```
1
*
* *
*
```

```
ring
```

```
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/*
```