Assignment 3 - Software I, Summer 2003 (0368-2157-20)

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

Due: Aug. 27, 2003

Before starting to answer the questions, please read very carefully the "Submission Guidelines"[1]. Make sure your program detects invalid input data, and prints out appropriate error messages. Do not add "friendly" messages to your program, as it is tested automatically by another program.

Ex 3.1 exact_point

Write a program that computes a final intersection point as in exercise 2.3, but this time the computation should be carried out in exact precision. The coordinates of the input points are integers, and the coordinates of the final point and any intermediate point are rational numbers, expressed as a fraction $\frac{p}{q}$, where p and q are arbitrary-large integers, and $q \neq 0$.

You need to implement the three data structures below and a few operations on these data structures:

Big_int - represents an arbitrary-large integer

Rat - represents a rational number

Point - represents a Cartesian point

The definition of these data structures, along with the declarations of the functions that operate on them, should be placed in three separate include files, namely `Big_int.h`, `Rat.h`, and `Point.h`. For your convenience, I have prepared these files for you. You may download them from
`http://www.cs.tau.ac.il/~efif/courses/software1/code/exact_point/{Big_int.h,Rat.h,Point.h}`.

While the prototypes of the functions in these files are optional, you must use the data structures listed in these files and replicated below:

Digit

```
typedef struct digit {
  unsigned char d;         /* A single digit in decimal format */
  struct digit * next;     /* A pointer to the next digit in the list */
} Digit;
```

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

An element in a linked list of digits. This struct is used by the Big_int structure. The digits in such a list represents an arbitrary large integer in decimal format, where the first digit in the list is the least significant and the last digit is the most significant.

Big_int

```
typedef struct big_int {
  int is_negative;        /* Is the integer negative? */
  int num_digits;         /* The number of digits in the list of digits */
  Digit * digits;         /* A linked list of digits */
} Big_int;
```

A struct that represents an arbitrary large integer. The printing format of a `Big_int` is a sequence of the decimal digits.

Rat

```
typedef struct rat {
  Big_int num;            /* The numerator */
  Big_int den;            /* The denominator */
} Rat;
```

A struct that represents a rational number. The printing format of a `Rat` is the numerator `Big_int`, followed by the '/' character, followed by the denominator `Big_int`.

Point

```
typedef struct point {
  Rat x;                  /* The x coordinate of the point */
  Rat y;                  /* The y coordinate of the point */
} Point;
```

A struct that represents a Cartesian point. The printing format of a `Point` is the x-coordinate `Rat`, followed by a space, followed by the y-coordinate `Rat`.

The comments in the file contain keywords that can be interpreted by the Doxygen documentation system. Visit the page `http://www.cs.tau.ac.il/~efif/courses/software1/code/exact_point/html` to see the documentation produced by doxygen.

Files Names

As usual place the files for the assignment under `~/software1/assign3`. This time, you need to put the implementation of the operations on the different data structures in separate files, i.e., `Big_int.c`, `Rat.c`, and `Point.c`. Place the `main()` function in yet another file `exact_point.c`. You need to provide an appropriate makefile file, that can be used to build the executable `exact_point`. Note that names are case sensitive (i.e. `Big_int.c` is different than `big_int.c`).

For example, input:

```
2 2
6 4
6 1
2 5
```

output:

```
96/24 72/24
```

For your convenience add a function to normalize a rational, and apply it to any rational before it is printed out. This is not mandatory though.

Good Luck!

More Information on the Submission

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/assign3 ~/software1/assign3/*
```