Assignment 2 - Computer Programming (0455-3391-01)

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

Due: December 24, 2002

Problem 1

Recall problem (1) of assignment (1) where you were asked to print a simple picture. Regard the simple picture as a pattern, and write a program that prints a picture that consists of the pattern duplicated horizontally and vertically to fill a rectangle of a given width and height. Allow for the pattern to be shifted horizontally and vertically by given two offsets respectively.

The program should read 4 integers before it prints out the picture. The width, the height, the horizontal offset, and the vertical offset. For example, using the pattern (the most left column is all spaces):

```
 --
/  \
\  /
```

providing 32 12 1 2 as input parameters, the final picture should look like:

```
  / \  / \  / \  / \  / \  / \
--   --   --   --   --   --   -
  \ /  \ /  \ /  \ /  \ /  \ /
  / \  / \  / \  / \  / \  / \
--   --   --   --   --   --   -
  \ /  \ /  \ /  \ /  \ /  \ /
  / \  / \  / \  / \  / \  / \
--   --   --   --   --   --   -
  \ /  \ /  \ /  \ /  \ /  \ /
  / \  / \  / \  / \  / \  / \
--   --   --   --   --   --   -
  \ /  \ /  \ /  \ /  \ /  \ /
```

Write a function that prints a line of the pattern, and call it from your main function. It should accept 3 parameters; (1) the index of the line in the pattern (2) the horizontal offset, and (3) the number of characters to print.

```
void printLine(int line, int offset, int num)
```

As before, you will not be graded on artistic quality, although extraordinary work may be displayed on the web site for this course.

There is a link to text file that contains a pattern in the course web page. Try to use it.

Problem 2

1. Write a program that reads an integer number, and "reduces" it to a single digit, by summing the digits until only one remains. For example, if the input is 4523, $4523 \to 14 \to 5$, the output is 5.
2. Write a similar program that reads an integer number in hexadecimal format, and does the same thing with its hexadecimal digits. For example, if the input is 0x8a9b4, $0x8a9b4 \to 0x2a \to 0xc$, the output is 0xc.

Problem 3

Consider a ladder with n levels. You are allowed to advance either one level or two levels each step, and you are asked to reach the top. How many different options do you have starting at the bottom?

The answer is the n-th Fibonacci number. Write a program to compute the n-th Fibonacci number without using arrays.

Problem 4

You have b identical crystal balls. There is a high rise with f floors. One of the floors could be critical but not necessary. When a crystal ball is dropped from the critical floor or from a higher floor the ball is smashed. You need to find the critical floor if exists, or report that it doesn't, with the minimum number of experiments in the worst case. Each drop is counted as one experiment regardless of whether the ball is smashes or not.

If you have only one ball, you must start your experiments from the first floor. In the worst case you will end up performing an experiment every floor just to find out that there is no critical floor. Given one ball the minimum number of experiments required in the worst case is f.

This is a riddle, that we are going to solve with a program, but instead of counting the minimum number of experiments, we will count the maximum number of floors that can be covered, given an allowed number of experiments e.

Let $f(e, b)$ denote the maximum number of floors we can cover with e experiments and b balls.

If $b = 0$ or if $e = 0$, we cannot do much. Thus,

$$f(e, 0) = 0$$
$$f(0, b) = 0$$

Suppose we are allowed e experiments with b balls. Let f' be the maximum number of floors we can cover with $e - 1$ experiments and $b - 1$ balls. Then, the best we can do is perform the first experiment at the floor $f' + 1$. If the ball smashes, we are down to $e - 1$ experiments and $b - 1$ balls, and this is exactly what we need to cover all the floors bellow f'. Otherwise, we can proceed and cover $f(e - 1, b)$ additional floors. This leads to the following recursing formula:

$$f(e, b) = f(e - 1, b) + f(e - 1, b - 1) + 1$$

1. Write a program that reads the number of experiments e and the number of balls b, and prints $f(e, b)$.
2. Write a program that reads the number of floors f and the number of b balls, and prints the minimum number of experiments required. You may do it the hard way. That is, write the function $f(e, b)$ from the previous section, and look for the first $\hat{e} = 1, 2, \ldots, \infty$ that satisfies

$$f \leq f(\hat{e}, b)$$