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

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

Due: December 3, 2002

Problem 1

1. Write a program that computes $\sin(x)$ according to the formula bellow, and prints out the result.

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

   The computation should stop after the first k elements have been summed, where k is an input argument. x is in radians.

2. Write a program that computes $\cos(x)$ according to the formula bellow, and prints out the result.

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

   The computation should stop when the absolute value of the next element in the series is smaller than an input argument ϵ. Notice that $0^0 = 1$.

Problem 2

The two formula bellow express π

$$\pi = 4 \times (1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots) \quad (1)$$

$$\pi = 2 \times (1 + \frac{1}{3} + \frac{1}{3} \cdot \frac{2}{5} + \frac{1}{3} \cdot \frac{2}{5} \cdot \frac{3}{7} + \frac{1}{3} \cdot \frac{2}{5} \cdot \frac{3}{7} \cdot \frac{4}{9} + \ldots) \quad (2)$$

Write a program that computes π twice according to the two formulas above. The computation should stop in both cases when the absolute value of the next element in the series is smaller than some ϵ. The program should print out the number of elements summed for each computation. For $\epsilon = 0.001$, the number of elements should be 10 for the computation of the first formula and 501 for the second.

Problem 3

Modify the dice-rolling program so that it computes:

1. the floating-point average μ of all the rolls of the pair of dice

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

2. the standard deviation s, which can be expressed as

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

If you put the line

```
#include <math.h>
```

at the top of the file, you'll be able to call sqrt().

3. the histogram using asterisks. For each roll print out as many asterisks as the number of times the roll has been sampled, normalized to fit lines of 80-characters long. For example, the output might look like this:

```
 2:  29 *********
 3:  52 *****************
 4:  82 ***************************
 5: 113 *************************************
 6: 145 ************************************************
 7: 167 *******************************************************
 8: 130 *******************************************
 9: 122 ****************************************
10:  78 **************************
11:  57 *******************
12:  25 ********
```

Problem 4

1. Write a program that converts decimal numbers to base-27 numbers of 6 digits in length. The digits used to represent the base-27 numbers are the space ' ' and the English letters 'a' through 'z'. The space represents 0, 'a' represents 1, 'b' represents 2,..., and 'z' represents 26. Formally,

$$l_5 l_4 l_3 l_2 l_1 l_0 = l_5 \times 27^5 + l_4 \times 27^4 + l_3 \times 27^3 + l_2 \times 27^2 + l_1 \times 27^1 + l_0 \times 27^0$$

For example:

$$\begin{aligned}
\texttt{a} &= 1 = 1 \\
\texttt{aaaaaa} &= 1 \times 27^5 + 1 \times 27^4 + 1 \times 27^3 + 1 \times 27^2 + 1 \times 27^1 + 1 \times 27^0 = 14900788 \\
\texttt{abcdef} &= 1 \times 27^5 + 2 \times 27^4 + 3 \times 27^3 + 4 \times 27^2 + 5 \times 27^1 + 6 \times 27^0 = 15473895 \\
\texttt{hello} &= 4359030
\end{aligned}$$

Each place value to the left is equal to 27 times the place value to the right, which implies that each place value to the right is equal to the place value to the left divided by 27.

The program should scan in an decimal integer number and print out the equivalent base-27 number in English letters.

Hint use the following expansion:

$$l_5 l_4 l_3 l_2 l_1 l_0 = l_0 + 27 \times (l_1 + 27 \times (l_2 + 27 \times (l_3 + 27 \times (l_4 + 27 \times l_5))))$$

2. Modify the program to handle base-27 numbers of 7 digits in length, and use it to find out the largest "english" number that its decimal equivalent can be held by an **unsigned int**.