Object-Oriented Programming (List 4)

Due: April 14th 2010

1. Write a function `double dmod( double x, double y )` that (assuming $x \geq 0, y > 0$) computes $x - n.y$ where n is the biggest natural number, s.t. $x - n.y \geq 0$. (It is the same as `%`, but on `double`.)

2. You have seen how to implement the Taylor sequence for the function e^x in the lecture, and successsfully written a program for computing $\sin(x)$ in the previous exercise list. Now implement the Taylor sequence for $\cos(x)$. It is defined by

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \cdots.$$

   In contrast to last week, you now have to write it as function. Write a function `double cosine( double x )`, that computes the cosine of a number. Use the function for printing a table of cosines from 0 to 90 degrees.

3. Write a function

```
unsigned int selectmin( const std::vector< unsigned int > & v )
```

   that selects the minimum element from a vector. (Without assuming that the vector is sorted.)

4. Write a function

```
void appendinorder( std::vector< unsigned int > & v, unsigned int x );
```

   that, assuming that v is a sorted vector, inserts x on the proper place in the vector.

5. Write function

```
std::vector< unsigned int > sort( const std::vector< unsigned int > & v )
```

that sorts a vector, using the previous function `appendinorder`.

6. Modify the previous functions, so that `appendinorder` can sort a vector of `double` .