

For all problems in problem set 1:

- 1) Problems should be submitted both electronically (Malia Kilpinen, malia@mit.edu) and on paper. Due date: **9/25/02, 3pm.**
- 2) All submitted problems must contain your name, email address, and which problem you are submitting.
- 3) The following should be written as comments in the code: a short description of what the program is doing, and also document what the most important variables, data structures, constants are and how they are used in the program.
- 4) In addition to 3) each module should have a “header comment” with the following information:

```
-----  
-- Module name: Name of module  
-- Explanation: What is the purpose of this module. Explain it so a person  
--               unfamiliar with the code can understand  
-- Input: input to this module from the calling unit  
-- Output: data that is returned to calling unit (for procedures)  
-- Return value: value that is returned to calling unit (for functions)  
-- Comments: extra information that might be needed for the unit to execute properly  
-----
```

Problem set 1: Language basics

Only part 1a) of this problem set is obligatory. Parts 1b and 1c are optional, but if submitted, they will be graded/corrected. Solutions to optional problems will be taken into consideration when final grades are awarded. All submitted code must be without compilation errors, able to run, and well documented. If not, the code will not be graded.

Readings: chapter 1-3 in Feldman/Koffman

Problem 1a) Sales tax

Problem 1b) Population calculation

Problem 1c) A Calendar

1a) Write a program that calculates the sales tax. Using the terminal, the program should accept as input the following values (all values must be "sanity checked" before accepted as inputs):

- Lower boundary for price range
- Upper boundary for price range
- Increments in table
- Tax rate as a decimal number in the range from 0 to 100%

Example 1) (inputs to the program are printed in bold)

Lower boundary: **10.00**

Upper boundary: **14.00**

Increments: **0.5**

Tax rate: **10.00**

Sales Tax

=====

Tax rate: 10.00

Price without tax	Tax	Price with tax
10.00	1.00	11.00
10.50	1.05	11.55
11.00	1.10	12.10
.	.	.
.	.	.
14.00	1.40	15.40

Example 2)

Lower boundary: **10.00**

Upper boundary: **12.00**

Increments: **0.3**

Tax rate: **20.00**

Sales Tax

=====

Tax rate: 20.00

Price without tax	Tax	Price with tax
10.00	2.00	12.00
10.30	2.06	12.36
10.60	2.12	12.72
.	.	.
.	.	.
11.80	2.36	14.16

Please note: the last "Price without tax" value is 11.80, not 12.00, nor 12.10.

1b) The populations in two countries are expected to grow exponentially. Write a program that as input reads a year, the population for that year for country A and B, and population increase in % for both countries. The program should produce a table with year, population increase this year for each country, and population at the end of the year. This continues until the country with the smallest start population has a larger population than the other country. This of course means the smaller country must have a larger population growth. This needs to be verified when data is entered into the program.

The program should also answer the following question: In what year will the initially smaller country first have a larger population than the initially larger country?

Example (in data written in bold)

Start year: **2002**

Population in country A (in millions): **200**

Population increase in country A (in %): **10.0**

Population in country B (in millions): **100**

Population increase in country B (in %): **50.0**

Year	Increase A	population A	increase B	population B
2002	---	200.00	---	100.00
2003	20.00	220.00	50.00	150.00
2004	22.00	242.00	75.00	225.00
2005	24.20	266.20	112.50	337.50

Country B will in year 2005 have passed country A when it comes to population

1c) Write a program that prints a calendar for a month. The output should look as follows for a specific month:

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Input to the program is weekday number (1-7) for the first day in the month and the number of days in the month. In the above example, the weekday number was 5.