

ENGRECE 12: Assignment-5

February 6, 2004

Due Wednesday 2/18/2004 12:00pm

1 Salary Database [20 points]

The goal of this programming exercise is to practice the use of list operations and the functional programming elements *map()*, *filter()*, and *reduce()* available in Python. The objective is to write a Python program, *sdb.py*, for a simple salary database for employees of a company. For privacy reasons, the data base will not contain the names of any employees, but only a list of salaries in dollar amounts. Each employee is simply identified by the index number in the list of salaries.

Our sample database looks as follows:

Employee 1	\$ 37500
Employee 2	\$ 40000
Employee 3	\$ 26000
Employee 4	\$ 32500
Employee 5	\$ 45000
Employee 6	\$ 92000
Employee 7	\$ 32500
Employee 8	\$ 14000
Employee 9	\$ 89000
Employee 10	\$105000
Employee 11	\$ 32500
Employee 12	\$ 26500

Your program should provide a menu with the following options:

```
Salary Database - Main Menu
1. Add a new salary entry
2. Print the list of salaries
3. Print the highest salary
4. Print the lowest salary
5. Print the average salary
6. Print a list of salaries within a range
7. Raise all salaries by a percentage
8. Exit
```

For menu item 1, let the user enter a new dollar amount to be appended to the end of the list.

For menu item 2, print all salaries in the database in four columns as follows:

\$ 37500	\$ 40000	\$ 26000	\$ 32500
\$ 45000	\$ 96000	\$ 32500	\$ 14000
\$ 89000	\$105000	\$ 32500	\$ 26500

For menu item 3 and 4, use the Python *reduce()* function to compute the appropriate value and print it to the screen.

For menu item 5, use the Python *reduce()* and *len()* functions to compute the average and print it to the screen.

For menu item 6, let the user enter an upper and lower boundary. Then, use the Python *filter()* function and print the resulting list in the same format as for menu item 2.

For menu item 7, let the user enter a percentage rate, e.g. 5%. Then, use the Python *map()* function to compute the raised salaries and store it in the database.

Implement the above salary database in a program called *sdb.py*.

Script: To demonstrate that your program works, create a typescript with the following steps:

- enter the salaries given in the sample database above
- print the salaries
- print the highest, lowest, and average salary amounts
- print the salaries between \$30000 and \$50000
- print the salaries below \$30000
- print any six-figure salaries
- raise the salaries by 5.5%
- print the salaries
- print any six-figure salaries after the raise

2 Exercise-4.8 Towers Of Hanoi [20 points]

HINT:

Use recursion to implement this exercise. A brief pseudo code of the recursive function *move()* is given below. This is just the pseudo code of the recursive function and it does not contain the main code. Your main code will prompt for user input and call this recursive function.

```
move(number of disks, from peg, to peg, temporary peg)
    if the 'Number of disks' equal to 1 then print 'From' peg and 'To' peg and return
    Decrement number of disks by 1
    move(Number of disks, From peg, Temporary peg, To peg)
    move(1, From peg, To peg, Temporary peg)
    move(Number of disks, Temporary peg, To peg, From peg)
```

Script: To demonstrate that your program works, create a typescript that shows the Towers-of-Hanoi solution for 4 disks.

EXTRA CREDITS [10 POINTS]:

For extra 10 credits, provide a graphical representation (similar to the figure in the book). The program should print the three stacks after each move. The drawing can be done by using special character symbols like

-, |, *

3 What to turn in

Use the command

```
% python ~/ece12/tools/submit.py
```

to turn in your homework. Your files should be a level above the hw3 directory. You should have the following files:

- sdb.py
- sdb.script
- 4.8.py
- 4.8.script