ICS3U/C

Nesting, Div & Mod Programming Exercise

□ Analyze and Design flowchart algorithms for each Problem Definition.
□ Obtain validation of each Design flowchart from your teacher.
□ Implement a working program for each algorithm design.
□ Hand in both the Design flowcharts and a print out of the Implemented programs (using FinePrint and 4-up).
□ ALL programs should be re-executable by the user.
□ All programs need only boundary check user input; type checking not required for this assignment.
□ Enrichment Opportunities need not be reflected in flowcharts; but may be added during testing as maintenance adds.

Digits of a number
(digits.py)

Design an algorithm (and implement) which will ask the user for a positive integer. The program will print the:

- number of digits in the number
- the sum of the digits of the number
- the reverse of the number
- and whether the number is a palindrome

For example if 23067 was input, the program will print that it has 5 digits, the sum of the digits is 18 and that the reverse number is 76032.
For example if 23432 was input, the program will print that it has 5 digits, the sum of the digits is 14 and that the reverse number is 23432; making it a palindrome!

Enrichment Opportunity: *include as output, the digital root o f the number. The digital root of a number is obtained by finding the sum of the digits of the number, repeatedly until the result is between 0 and 9. That result is the digital root. For example the digital root of 79 is 7 because: 7+9 = 16, and 1+6 = 7. Avoiding an infinite loop, a digital root should only try repeatedly to a maximum of 20 attempts.*

Smallest and Largest Numbers
(smallest.py)

Design an algorithm (and implement) which will ask the user to enter a series of integers, stopping when the user enters 0. (Edit the first number so that it is NOT 0: that is, ensure that the user enters at least one number, and does not stop instantly.) When the user enters 0, print the smallest and largest number the user has entered. For example if the user entered

```
        8, 3, -4, 5, 8, 10, 5, 0
```

The program would print that the largest number was 10 and the smallest number was -4.
Enrichment Opportunity: *include as output, the Average of all valid numbers entered by the user.*

Parking Garage (park.py)
Enrichment Opportunity for ICS3C

Design an algorithm (and implement) which will determine the price you pay for parking at the following garage with this rate structure (*assuming all vehicles must vacate by midnight):

$3.00 per 20 minutes
$28.00 daily maximum
$5.00 flat rate for 6pm or later*
$8.50 for staff with a pass

Input to this program will include the total minutes parked, the time you entered the garage and whether you have a pass or not.

For example:

total minutes	time entered	pass	price
153	0900	N	$24.00
10	1435	N	$3.00
600	0700	N	$28.00
480	0800	Y	$8.50
360	1800	N	$5.00