Programming Exercise 1.1

Your First Algorithm

Purpose. Learn how algorithms work, by making changes to an existing algorithm. A good way to learn in computer programming is to start by using what someone else already did, then modify it.

Here's a complete algorithm for a family member to calculate the average age of 5 members of their family: their father, mother, brother, sister, and their family's cat:

```
Write on a piece of paper these 5 labels: "dad", "mom", "bro", "sis", "cat".

Ask my dad his age -- write it under the label "dad".

Ask my mom her age -- write it under the label "mom".

Write this new label on the paper: "sum".

Add the numbers under the labels "mom" and "dad" -- write the result under the label "sum".

Ask my brother his age -- write it under the label "bro".

Add the numbers under the labels "sum" and "bro"
  --  write the result under the label "sum",
  scratching out what I wrote there previously.

Ask my sister her age -- write it under the label "sis".

Add the numbers under the labels "sum" and "sis"
  --  write the result under the label "sum",
  scratching out what I wrote there previously.

Ask my mom how old is the cat -- write it under the label "cat".

Add the numbers under the labels "sum" and "cat"
  --  write the result under the label "sum",
  scratching out what I wrote there previously.

Write this new label on the paper: "avg".

Using a calculator, divide the number under the label "sum" by 5
  -- write the result under the label "avg".

Write on a new piece of paper this label: "My family's average age is "
  followed by the value written under the label "avg"
  on the original piece of paper.
```

That's it!

Requirements.

1. Use a word-processor program, like Word or Pages, to create a file named **`MyFirstAlgorithm.docx`**.

2. Modify the supplied example algorithm to calculate the average age of ***six*** family members.

3. Modify the supplied example algorithm to use names other than the ones in the example. Use a real or imaginary family, as you wish.

4. When adding numbers, add only two numbers at a time -- do *NOT* have a step that adds more than 2 numbers together. (Addition is a "binary operation" in computer languages -- a plus sign operates on exactly two numbers.)

5. Be exact and brief in your instructions. Go through your instructions to make sure they work -- that is, *test* the logic of your algorithm. Do *NOT* include any actual answers or results of your calculations. Do *NOT* use looping or branching, because this exercise can be done with sequential processing only.

6. Specify how to round off the result of the division calculation at the end -- that is, say how many digits to include, like "round to the nearest tenth" or "round to the nearest whole number". This is a new thing that's not in the supplied algorithm. No details – just decide how you want the result rounded and say so.

Example. For example, write…

```
Write on a piece of paper...
...
...
...
```