Ax paper task - report

# Requirements

* The program should ask user to enter a number paper size
* User can enter anything, but accepted values are 1 to 1000
* User input must be numbers
* The string asking the user what size paper they want must be “Which size do you want to calculate?\nA”
* The output must be formatted as “A paper of size Ax is x by y mm”
* The program must use one or more loops
* The program must follow the flow

# Design

I decided to split everything in to separate functions to improve readability.

I also wrote some verification functions as to check the validity of the input and get the length.

I am aware that there exist libraries to do some of the stuff that my functions do, I did however feel that that it was to inappropriate to use libraries in a skill assessment situation.

Main():

Print “Which size do you want to calculate?\nA”

User inputs What size they want

Checks that user input is number with the Atoi function

If it is a number store the value returned by the function

Else continue

Compute the size of the Ax paper using the Compute function

Go back to the top again

Calculate():

Size = {210, 297} (Size of A4)

If we want an Ax paper where x is smaller than 4

For(i = 0; i < 4-x; ++i)

Swap places of the paper height and width

Double the height of the paper

Else

For ( i = 0; i < x-4 ; ++i)

Swap places of the height and width

Half the width of the paper

Sizex = (int)width if width >=1 else width

Sizey = (int)height if height >=1 else height

Print the output string

// this is a function I wrote to convert strings to numbers

Atoi():

Check that the string is a number;

Check that the string is in the length limits using

Get every characters representative integer value and multiply it

By 10 to the power of the string length – its index;

Return the number

# Testing

I ran my code against some test inputs to see that everything functioned properly. These are some of the inputs I tried.

I used <http://resources.printhandbook.com/pages/paper-size-chart.php> to verify my outputs. They differ at some points by 1 millimeter and Sandeep has said that that is ok.

|  |  |
| --- | --- |
| Test input | Output |
| 4 | A paper of size A4 is 210 by 297 mm |
| ABCD | You must enter a number |
| 1234 | You must enter a value smaller or equal to 1000 |
| 12345 | You must enter a number "shorter" than 10000 |
| 2 | A paper of size A2 is 420 by 594 mm |
| 6 | A paper of size A6 is 105 by 148 mm |
| -1 | -1 |
| 1000 | A paper of size A1000 is 2.56615e-148 by 3.62926e-148 mm |

# Problems and solutions

During testing I struggled with finding an efficient way of verifying that a string is numerical. I did however find a rather nice way of doing it using the ASCII values of the characters. (Link to the ascii table <http://www.asciitable.com/> )

I also struggled a bit with the proper way of printing the values, I first thought that we were to use integers but that would be problematic after A19. The solution was do use long doubles and then format the string with %g

# Conclusions

My takeaway from this is that google has a lot of resources, but they are more limited for c than other programming languages. I can also say that there are some hoops that you must jump through when writing c code that are nonexistent in other languages.

I learnt that c has an easy way to compare text to ascii.