A4 to Ax paper size lab

Christer Zhou and **Tobias Skarhed**

chrzho-7@student-ltu.se, tobska-7@student.ltu.se

Solution

Our architecture was based on making the `main` function as simple as possible, and extract all other specific functionality to other functions. `getInput` would handle everything that had to do with input; checking size of string, valid characters etc. `changeSize` would only handle the resizing of a paper. We had a struct called `paper` to bundle the width, height and size together.

```
typedef struct{
    double width;
    double height;
    unsigned int size;
}paper;
```

For the structs width and height we used `double`s to get more precise calculations.

Checking input

We checked for:

- Strings no longer than 4 (As 1000 is maximum allowed)
- That each character had an ASCII value between 48 to 57. Numeric characters 0-9.
    - Since `-` is not allowed, we cover negative numbers too.
- Values that do not exceed 1000.
- Trailing zeroes.

Improvements

- Compensate for user error
    - Assume what the user tries to put into the program and tell the user that an assumption has been made, while returning the assumed value
- Do not exit program if input is invalid
    - Keep asking for new input until correct

Problems

The formula does not agree with the defined ISO standard for Ax papers. We had some issue with formatting the numbers in `printf` when working with small sizes. That was solved by using `%g` to get scientific notation.

```
printf("A paper of size A%i is %g by %g mm\n",
    new_paper.size,
    new_paper.width,
    new_paper.height
);
```

Testing

Test cases:

- `A2323 -> Too big number`
- `A01 -> Trailing zeroes not allowed`
- `A!23 -> Only numeric characters allowed`
  - Covers `-`, spaces and pretty much every other character that is not a number.
- `A12222 -> Maximum length allowed is 4`