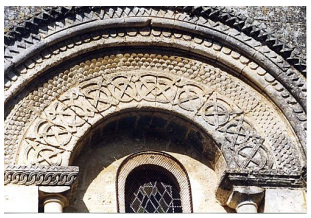
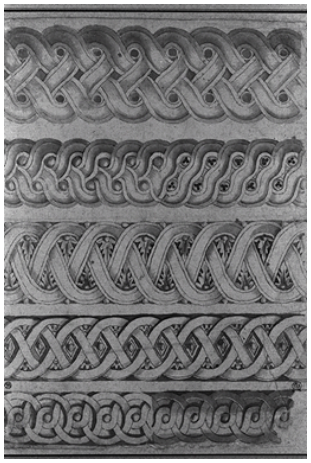




Assignment 2

COMP9021, Session 2, 2014

1 Aims

The main purpose of the assignment is to let you practice the following programming techniques:

- read data from standard input and store them in an array;
- perform operations on arrays;
- execute tests and repetitions;
- make use of the internal representation of data.

2 General presentation

You will design and implement a program that will

- check whether some numbers, stored in a file whose contents is to be redirected to standard input, represent a particular coding of a *frieze*, and
-
  - either display the period of the pattern of the frieze and the transformations that keep it invariant, based on a result that classifies friezes into 7 groups of symmetries,
  - or output some Latex code, to be redirected to a file, from which a pictorial representation of the frieze can be produced.

The representation of a frieze is based on a coding with numbers in the range `0` ... `15`, each such number n being associated with a particular point p such that

- if the rightmost digit of the representation of n in base 2 is equal to `1` then p is to be connected to its northern neighbour: `|`

- if the second rightmost digit of the representation of n in base 2 is equal to 1 then p is to be connected to its north-eastern neighbour: /
- if the third rightmost digit of the representation of n in base 2 is equal to 1 then p is to be connected to its eastern neighbour: –
- if the fourth rightmost digit of the representation of n in base 2 is equal to 1 then p is to be connected to its south-eastern neighbour: \

3 Examples

3.1 First example

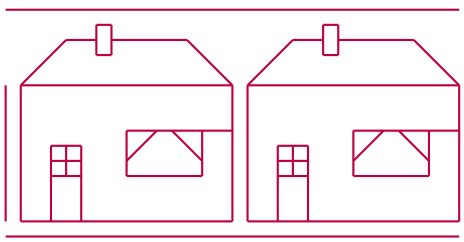
Given a file named `frieze_1.txt` whose contents is

```
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 0
0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0
0 0 0 0 4 4 1 5 4 4 4 4 8 0 0 0 0 0 0 4 4 1 5 4 4 4 4 8 0 0 0
0 0 0 2 0 0 5 1 0 0 0 0 0 8 0 0 0 0 2 0 0 5 1 0 0 0 0 0 8 0 0
0 0 2 0 0 0 0 0 0 0 0 0 0 0 8 0 0 2 0 0 0 0 0 0 0 0 0 0 0 8 0
0 6 4 4 4 4 4 4 4 4 4 4 4 4 4 4 0 6 4 4 4 4 4 4 4 4 4 4 4 4 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 0 0 0 0 0 0 0 4 4 4 12 4 4 4 4 1 1 0 0 0 0 0 0 0 4 4 4 12 4 4 4 4 1
1 1 0 4 4 0 0 0 1 2 0 0 8 1 0 1 1 0 4 4 0 0 0 1 2 0 0 8 1 0 1
1 1 0 5 5 1 0 0 3 0 0 0 0 1 0 1 1 0 5 5 1 0 0 3 0 0 0 0 1 0 1
1 1 0 5 5 1 0 0 5 4 4 4 4 1 0 1 1 0 5 5 1 0 0 5 4 4 4 4 1 0 1
1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1
1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1
1 5 4 5 4 5 4 4 4 4 4 4 4 4 4 1 5 4 5 4 5 4 4 4 4 4 4 4 4 4 1
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 0
```

your program when run as `a.out <frieze_1.txt` should output

```
Pattern is a frieze of period 15 that is invariant under translation only.
```

and when run as `a.out print <frieze_1.txt` should produce some output which if redirected to a file, say `frieze_1.tex`[1], can be given as argument to `pdflatex` to produce a file named `frieze_1.pdf` that views as follows.



[1] by actually running `a.out print <frieze_1.txt >frieze_1.tex`

3.2 Second example

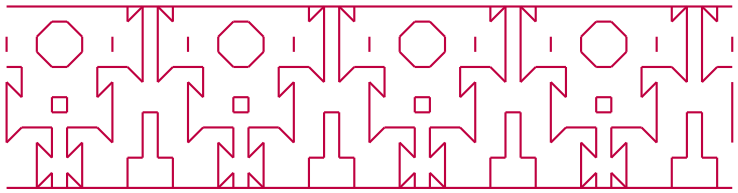
Given a file named `frieze_2.txt` whose contents is

```
4 4 4 4 4 4 4 4 4 4 12 4 4 4 4 4 4 4 4 4 4 4 12 4 4 4 4 4 4 4 4 4 4 4 12 4 4 4 4 4 4 4 4 4 4 4 12 4 0
0 0 0 4 8 0 0 0 3 1 1  1 0 0 0 4 8 0 0 0 3 1 1  1 0 0 0 4 8 0 0 0 3 1 1  1 0 0 0 4 8 0 0 0 3 1 1  1 0
0 0 2 0 0 0 0 0 0 1 1  0 0 0 2 0 0 0 0 0 0 1 1  0 0 0 2 0 0 0 0 0 0 1 1  0 0 0 2 0 0 0 0 0 0 1 1  0 0
1 0 9 0 0 1 0 1 0 1 1  0 1 0 9 0 0 1 0 1 0 1 1  0 1 0 9 0 0 1 0 1 0 1 1  0 1 0 9 0 0 1 0 1 0 1 1  0 1
4 0 0 4 2 0 4 4 8 1 1  4 4 0 0 4 2 0 4 4 8 1 1  4 4 0 0 4 2 0 4 4 8 1 1  4 4 0 0 4 2 0 4 4 8 1 1  4 0
8 1 0 0 0 0 1 0 0 1 3  0 8 1 0 0 0 0 1 0 0 1 3  0 8 1 0 0 0 0 1 0 0 1 3  0 8 1 0 0 0 0 1 0 0 1 3  0 0
1 1 0 4 0 0 3 1 0 0 0  0 1 1 0 4 0 0 3 1 0 0 0  0 1 1 0 4 0 0 3 1 0 0 0  0 1 1 0 4 0 0 3 1 0 0 0  0 1
1 0 0 5 1 0 0 1 0 4 0  0 1 0 0 5 1 0 0 1 0 4 0  0 1 0 0 5 1 0 0 1 0 4 0  0 1 0 0 5 1 0 0 1 0 4 0  0 1
1 4 4 0 4 4 8 1 0 1 1  0 1 4 4 0 4 4 8 1 0 1 1  0 1 4 4 0 4 4 8 1 0 1 1  0 1 4 4 0 4 4 8 1 0 1 1  0 1
3 0 8 1 1 0 0 1 0 1 1  0 3 0 8 1 1 0 0 1 0 1 1  0 3 0 8 1 1 0 0 1 0 1 1  0 3 0 8 1 1 0 0 1 0 1 1  0 1
0 0 1 1 3 1 0 0 4 1 5  0 0 0 1 1 3 1 0 0 4 1 5  0 0 0 1 1 3 1 0 0 4 1 5  0 0 0 1 1 3 1 0 0 4 1 5  0 0
0 0 1 0 8 1 0 0 1 0 0  1 0 0 1 0 8 1 0 0 1 0 0  1 0 0 1 0 8 1 0 0 1 0 0  1 0 0 1 0 8 1 0 0 1 0 0  1 0
4 4 7 5 5 5 4 4 5 4 4  5 4 4 7 5 5 5 4 4 5 4 4  5 4 4 7 5 5 5 4 4 5 4 4  5 4 4 7 5 5 5 4 4 5 4 4  5 0
```

your program when run as `a.out <frieze_2.txt` should output

```
Pattern is a frieze of period 12 that is invariant under translation
        and vertical reflection only.
```

and when run as `a.out print <frieze_2.txt` should produce some output which if redirected to a file, say `frieze_2.tex`[2], can be given as argument to `pdflatex` to produce a file named `frieze_2.pdf` that views as follows.



[2]by actually running `a.out print <frieze_2.txt >frieze_2.tex`

3.3 Third example

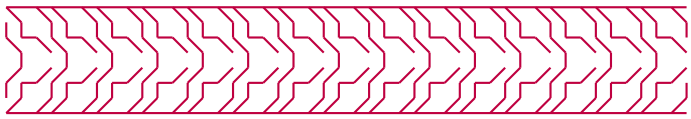
Given a file named `frieze_3.txt` whose contents is

```
12 4 12   12 4 12   12 4 12   12 4 12   12 4 12   12 4 12   12 4   12 12   4 12 12   4 12 12   4 12 12   4 12 12   4 12 12   4 12 12   4 12 12   4 12 0
0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0
9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   9 5 8   1
0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0 0 0   0
0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0 1 0   0
2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   2 4 2   0
1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1 1 0   1
6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   6 4 6   0
```

your program when run as `a.out <frieze_3.txt` should output

```
Pattern is a frieze of period 3 that is invariant under translation
        and horizontal reflection only.
```

and when run as `a.out print <frieze_3.txt` should produce some output which if redirected to a file, say `frieze_3.tex`[3], can be given as argument to `pdflatex` to produce a file named `frieze_3.pdf` that views as follows.



[3] by actually running `a.out print <frieze_3.txt >frieze_3.tex`

3.4 Fourth example

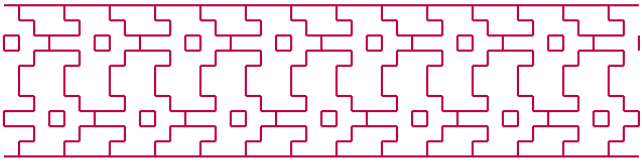
Given a file named `frieze_4.txt` whose contents is

```
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 0
0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0
4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 0
5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 1
0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0

0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0

0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0 5 0 0
4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 4 4 1 4 0 5 0
5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 5 4 0 5 1 4 1
0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0 4 1 0
4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 4 5 4 0
```

your program when run as `a.out <frieze_4.txt` should output

```
Pattern is a frieze of period 6 that is invariant under translation
        and glided horizontal reflection only.
```

and when run as `a.out print <frieze_4.txt` should produce some output which if redirected to a file, say `frieze_4.tex`[4], can be given as argument to `pdflatex` to produce a file named `frieze_4.pdf` that views as follows.



[4] by actually running `a.out print <frieze_4.txt >frieze_4.tex`

3.5 Fifth example

Given a file named `frieze_5.txt` whose contents is

```
4  4 4 4 4 4 12 4   4  4 4 4 4 4 12 4   4  4 4 4 4 4 12 4   4  4 4 4 4 4 12 4   4  4 4 4 4 4 12 4   4  4 4 4 4 4 12 4   0
0  4 1 0 0 2 0  8   0  4 1 0 0 2 0  8   0  4 1 0 0 2 0  8   0  4 1 0 0 2 0  8   0  4 1 0 0 2 0  8   0  4 1 0 0 2 0  8   0
10 0 0 0 2 0 0  0   10 0 0 0 2 0 0  0   10 0 0 0 2 0 0  0   10 0 0 0 2 0 0  0   10 0 0 0 2 0 0  0   10 0 0 0 2 0 0  0   0
0  8 0 2 0 0 4  2   0  8 0 2 0 0 4  2   0  8 0 2 0 0 4  2   0  8 0 2 0 0 4  2   0  8 0 2 0 0 4  2   0  8 0 2 0 0 4  2   0
4  4 6 4 4 4 5  4   4  4 6 4 4 4 5  4   4  4 6 4 4 4 5  4   4  4 6 4 4 4 5  4   4  4 6 4 4 4 5  4   4  4 6 4 4 4 5  4   0
```

your program when run as `a.out <frieze_5.txt` should output

```
Pattern is a frieze of period 8 that is invariant under translation
        and rotation only.
```

and when run as `a.out print <frieze_5.txt` should produce some output which if redirected to a file, say `frieze_5.tex`[5], can be given as argument to `pdflatex` to produce a file named `frieze_5.pdf` that views as follows.

[5] by actually running `a.out print <frieze_5.txt >frieze_5.tex`

3.6 Sixth example

Given a file named `frieze_6.txt` whose contents is

```
4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      0
4 0 4 4 4 0 4 4 4 0      4 4 4 0 4 4 4 0 4 4      4 0 4 4 4 0 4 4 4 0      4 4 4 0 4 4 4 0 4 4      0
0 1 1 4 0 1 1 4 0 1      1 4 0 1 1 4 0 1 1 4      0 1 1 4 0 1 1 4 0 1      1 4 0 1 1 4 0 1 1 4      0
1 5 1 1 1 5 1 1 1 5      1 1 1 5 1 1 1 5 1 1      1 5 1 1 1 5 1 1 1 5      1 1 1 5 1 1 1 5 1 1      1
5 4 4 1 5 4 4 1 5 4      4 1 5 4 4 1 5 4 4 1      5 4 4 1 5 4 4 1 5 4      4 1 5 4 4 1 5 4 4 1      1


4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      0
```

your program when run as `a.out <frieze_6.txt` should output

```
Pattern is a frieze of period 4 that is invariant under translation,
        glided horizontal and vertical reflections, and rotation only.
```

and when run as `a.out print <frieze_6.txt` should produce some output which if redirected to a file, say `frieze_6.tex`[6], can be given as argument to `pdflatex` to produce a file named `frieze_6.pdf` that views as follows.

[6] by actually running `a.out print <frieze_6.txt >frieze_6.tex`

3.7 Seventh example

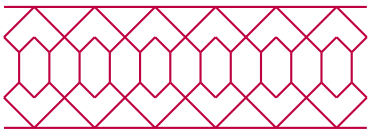
Given a file named `frieze_7.txt` whose contents is

```
4 4 12 4 4 4 12 4 4 4 12 4 4 4 12 4 4 4 12 4 4 4 12 4 0
      0 2 0 8 0 2 0 8 0 2 0 8 0 2 0 8 0 2 0 8 0 2 0 8 0
10 0 8 0 10 0 8 0 10 0 8 0 10 0 8 0 10 0 8 0 10 0 8 0 0
      0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0
      0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
      0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0
10 0 2 0 10 0 2 0 10 0 2 0 10 0 2 0 10 0 2 0 10 0 2 0 0
      0 8 0 2 0 8 0 2 0 8 0 2 0 8 0 2 0 8 0 2 0 8 0 2 0
      4 4 6 4 4 4 6 4 4 4 6 4 4 4 6 4 4 4 6 4 4 4 6 4 0
```

your program when run as `a.out <frieze_7.txt` should output

```
Pattern is a frieze of period 4 that is invariant under translation,
        horizontal and vertical reflections, and rotation only.
```

and when run as `a.out print <frieze_7.txt` should produce some output which if redirected to a file, say `frieze_7.tex`[7], can be given as argument to `pdflatex` to produce a file named `frieze_7.pdf` that views as follows.



[7] by actually running `a.out print <frieze_7.txt >frieze_7.tex`

4 Detailed description

4.1 Input

The input is expected to consist of $height + 1$ lines of $length + 1$ numbers in $\{0, ..., 15\}$, where $length$ is at least equal to 4 and at most equal to 50 and $height$ is at least equal to 2 and at most equal to 16, with possibly lines consisting of spaces only that will be ignored and with possibly spaces anywhere on the lines with digits. The x^{th} digit n of the y^{th} line, with $0 \leq x \leq length$ and $0 \leq y \leq height$,

- is to be associated with a point situated $x \star 0.2$ cm to the right and $y\star$ 0.2 cm below an origin,
- is to be connected to the point 0.2 cm above if the rightmost digit of n is 1,
- is to be connected to the point 0.2 cm above and 0.2 cm to the right if the second rightmost digit of n is 1,
- is to be connected to the point 0.2 cm to the right if the third rightmost digit of n is 1, and
- is to be connected to the point 0.2 cm to the right and 0.2 cm below if the fourth rightmost digit of n is 1.

To qualify as a frieze, the input is further constrained to fit in a rectangle of length $length \star 0.2$ cm and of height $heigth \star 0.2$ cm, with horizontal lines of length $length$ at the top and at the bottom, identical vertical borders at both ends, no crossing segments connecting pairs of neighbours inside the rectangle, and a pattern of integral period at least equal to 2 that is fully repeated at least twice in the horizontal dimension.

4.2 Output

If not run as `a.out` or as `a.out print` (followed by `<`*`filename`* where *`filename`* is the name of a file that stores the input) then the program should print out a single line that reads

```
I expect no command line argument or "print" as unique command line argument.
```

and immediately exit. Otherwise, if the input is incorrect in that it does not contain only numbers in $\{0, \ldots, 15\}$ besides spaces, or in that it contains either two few or too many lines of numbers, or in that some line of numbers contains too many or two few numbers, or in that two of its lines of numbers do not contain the same number of numbers, then the program should print out a single line that reads

```
Incorrect input.
```

and immediately exit. If the previous conditions hold but the further conditions spelled out above for the input to qualify as a frieze do not hold, then the program should print out a single line that reads

```
Input does not represent a frieze.
```

and immediately exit.

4.2.1 When a.out is run with no command-line argument

If the input is correct and represents a frieze and the program is run as `a.out` (followed by `<`*`filename`* where *`filename`* is the name of a file that stores the input) then the program should output one or two lines that read

```
Pattern is a frieze of period N that is invariant under translation only.
```

or

```
Pattern is a frieze of period N that is invariant under translation
        and vertical reflection only.
```

or

```
Pattern is a frieze of period N that is invariant under translation
        and horizontal reflection only.
```

or

```
Pattern is a frieze of period N that is invariant under translation
        and glided horizontal reflection only.
```

or

```
Pattern is a frieze of period N that is invariant under translation
        and rotation only.
```

or

```
Pattern is a frieze of period N that is invariant under translation,
        glided horizontal and vertical reflections, and rotation only.
```

or

```
Pattern is a frieze of period N that is invariant under translation,
        horizontal and vertical reflections, and rotation only.
```

with `N` an appropriate integer at least equal to 2.

These 7 possible outputs are based on a mathematical result on the classification of friezes that lists all possible complete lists of symmetries that leave a frieze invariant under an isometry (that is, a transformation that does not alter the distance between any two points). These possible lists involve 5 symmetries.

- Translation by *period*; of course, any frieze is invariant under this symmetry.
- Vertical reflection about some vertical line; that line does not necessarily delimit the pattern nor does it necessarily go through its middle (these conditions are actually equivalent).

- Horizontal reflection about the line that goes through the middle of the frieze.
- Glided horizontal reflection, that is, horizontal reflection about the line that goes through the middle of the frieze and translation by half the period of the resulting lower half of the frieze.
- Rotation around some point situated on the horizontal line that goes through the middle of the frieze; this is equivalent to horizontal refection combined with vertical reflection.

Pay attention to the expected format, including spaces. Note that your program should output no blank line. For a given test, the output of your program will be compared with the expected output; your program will pass the test if and only if both outputs are absolutely identical, character for character, including spaces. For the provided examples, the expected outputs are available in files that end in `_output.txt`. To check that the output of your program on those examples is correct, you can redirect it to a file and compare the contents of that file with the contents of the appropriate `_output.txt` file using the `diff` command. If `diff` silently exits then your program passes the test; otherwise it fails it. For instance, run

```
a.out <frieze_1.txt >frieze_1_my_output.txt
```

and then

```
diff frieze_1_my_output.txt frieze_1_output.txt
```

to check whether your program succeeds on the first provided example.

4.3 When a.out is run with print as command-line argument

If the input is correct and represents a frieze and the program is run as `a.out print` (followed by `<`*`filename`* where *`filename`* is the name of a file that stores the input) then the program should output some lines that redirected to a file, say `frieze.tex`, can be given as an argument to `pdflatex` to produce a file named `frieze.pdf` that depicts the frieze. The provided examples will show you what `frieze.tex` should contain. Segments are drawn in purple with a single *draw* command for each longest segment,

- starting with the vertical segments, from the topmost leftmost one to the bottommost rightmost one with the leftmost ones first,
- followed by the segments that go from north west to south east, from the topmost leftmost one to the bottommost rightmost one with the topmost ones first,
- followed by the segments that go from west to east, from the topmost leftmost one to the bottommost rightmost one with the topmost ones first,
- followed by the segments that go from the south west to the north east, from the topmost leftmost one to the bottommost rightmost one with the topmost ones first.

Pay attention to the expected format, including spaces and blank lines. Lines that start with `%` are comments; there are 4 such lines, that have to be present even when there is no item to be displayed of the kind described by the comment. The output of your program redirected to a file will be compared with the expected output saved in a file (of a different name of course) using the `diff` command. For

your program to pass the associated test, `diff` should silently exit, which requires that the contents of both files be absolutely identical, character for character, including spaces and blank lines. Check your program on the provided examples using the associated `.tex` files. For instance, run

```
a.out print <frieze_1.txt >my_frieze_1.tex
```

and then

```
diff my_frieze_1.tex frieze_1.tex
```

to check whether your program succeeds on the first provided example.

5 Assessment and submission

5.1 Assessment

Up to eight marks will reward correctness of solutions by automatically testing your program on some tests, all different to the provided examples. Read carefully the part on program output to maximise your chances of not failing some tests for stupid reasons.

Up to one mark will reward good formatting of the source code and reasonable complexity of the underlying logic as measured by the level of indentation of statements. For that purpose, the `mycstyle` script will be used, together with your customised `style_sheet.txt`, that you have to submit, and in which **Maximum level of indentation has to be set to 5** for this assignment. If the script identifies problems different to excessive indentation levels then you will score 0 out of 1. If the script identifies excessive indentation levels (which means that at least one line exhibits at least 6 indentation levels), then you will score 0.5 out of 1. If the script identifies no problem then you will score 1 out of 1. If your program attempts too little and contains too little code then the `mycstyle` script won't be used and you will score 0 out of 1.

Up to one mark will reward good comments, good choice of names for identifiers and functions, readability of code, simplicity of statements, compactness of functions. This will be determined manually.

Late assignments will be penalised: the mark for a late submission will be the minimum of the awarded mark and 10 minus the number of full and partial days that have elapsed from the due date.

5.2 Submission

Your program will be stored in a file named `frieze.c`, which has to be developed from the provided template. You can add whatever you want but do not remove anything from the template, and do not modify the `main()` function.

Go through `assignment_checklist.pdf`, provided with the second set of Lecture notes, and make sure that you can tick all boxes (or at the very least, are aware that you should tick them all). Then upload your files (the source code of your program and your customised `style_sheet.txt`) using WebCMS. Assignments can be submitted more than once: the last version is marked. Your assignment is due by September 28, 11:59pm.

5.3 Reminder on plagiarism policy

You are permitted, indeed encouraged, to discuss ways to solve the assignment with other people. Such discussions must be in terms of algorithms, not C code. But you must implement the solution on your own. Submissions are routinely scanned for similarities that occur when students copy and modify other people's work, or work very closely together on a single implementation. Severe penalties apply to a submission that is not the original work of the person submitting it.