# Reading and Writing files

## Problem description

You will write a program that reads from one file, interprets what it reads and writes to a second fine.

There is an example input file in Moodle: **`infile.dat`**. You will need to place that file onto your computer in the same folder that contains the main.cpp program for this project. If you prefer, you can rename it to **infile.txt**, it will work the same either way. Just make sure that your program opens the correct file.

The program should then do the following things:

1) ***Open*** **`infile.dat`** for ***reading***.
2) ***Open*** **`outfile.dat`** for ***writing***.
3) ***Read*** the information on **`infile.dat`**, one line at a time, until the file is at EOF or an E is read.
4) Interpret the information read and ***write*** the appropriate output on **`outfile.dat.`**
5) ***Close*** both files.

In Moodle there are example programs that open a file and read from it, write to it, or append to it. You should use these to give you an idea of how to do the file I/O portion of this exercise.

Each record (line) of infile.dat will have the following format: **code char value1 value2(optional)**

The code will be from the following set of characters: **D, R, S, T, E.**

The char will be a single character and indicate what character you should use to create your output figure.

The values will be the size of the output figure. For D, S, and T, there will be only one value. For R, there will be two. E will have no additional values on its line.

You will need to have functions that you call that actually do the output. They can either write to the output file or they can return a string that the main program writes. There should be one function for each shape (D, R, S, T). You will need to pass the output character and the dimensions to that function also.

Your four functions will need to be recursive. You can look in the section in Moodle on recursion to see an in-class exercise that shows how to do a rectangle recursively to give you some help with your programming.

The codes and their meanings are as follows:

### End

If the code is an **E**, you are to **end** the program.

### Diamond

If the code is a **D**, you are to create a **diamond** of the char. Diamonds should always have an odd number for their dimension. For example, with the line
**D ! 3**

You should generate a **diamond** that looks like
```
 !
!!!
 !
```

For the line
**D * 5**

You should generate a **diamond** that looks like
```
  *
 ***
*****
 ***
  *
```

### Square

If the code is an **S**, you are to create a **square** of the char. For example, with the line
**S ! 3**

You should generate a **square** that looks like
```
!!!
!!!
!!!
```

For the line
**S @ 5**

You should generate a **square** that looks like
```
@@@@@
@@@@@
@@@@@
@@@@@
@@@@@
```

### Triangle

If the code is a **T**, you are to create a **triangle** of the char. For example, with the line
**T = 3**

You should generate a **triangle** that looks like
**=**
**==**
**===**

For the line
**T ^ 5**

You should generate a **triangle** that looks like
**^**
**^^**
**^^^**
**^^^^**
**^^^^^**

### Rectangle

If the code is an R, you are to create a rectangle of the char. Rectangles, have two values, rows and columns. For example, with the line
**R a 2 4**

You should generate a rectangle that looks like
**aaaa**
**aaaa**

For the line
R & 6 3

You should generate a rectangle that looks like
**&&&&&&**
**&&&&&&**
**&&&&&&**

## Notes to keep in mind:

- Before starting, think about what information you'll need to keep track of during execution.

- The program will need a reading loop that is terminated with end of file or the code E, some way to determine which code was entered, and a function to draw each shape. The function will need to have as parameters the character to use to draw the shape, the dimension(s) of the shape, and the ouput file to write it on.

- You should check for an improper input file. That is, if you get an EOF when trying to read something, you should exit the program with an error message.

## Example Input File

T & 4
S @ 6
T x 5
R * 5 7
D – 7
D + 5
R = 4 3
E

## Evaluation

Before you submit your work, take a look at the evaluation criteria and make sure that your program meets them.