

1 Description

You will write a very simple graphics application. Instead of pixels and colours this program will manipulate ASCII characters and display them on the console. Examples of this style are called *ascii-art*. The current state of the picture is called the canvas. This name is used as a painting metaphor: when an artist paints on a canvas each brush stroke places paint on top of the exists picture, overwriting whatever was there before. The canvas in your program will be a rectangular grid of characters, and each shape that is drawn will overwrite whatever was in those cells previously.

Your program will accept commands from the keyboard in a specified format. In this assignment we have listed the commands and given you a description of what they should do. You must also read the example transcripts to see the format they are entered in, and the format of the responses. Together these two descriptions give you a specification of the input your program must accept, and the output it must produce. You will need to write the code that performs the operations each command creates. Your program should be structured as a main I/O loop that inputs a command, decides what to do with it and performs the correct operation. The canvas should be output to screen after each drawing operation, as shown in the examples.

The coordinate system for the commands will place position $(0, 0)$ in the top-left of the canvas (i.e. it will be the first character output to the console). The bottom-right coordinate depends on the size of the canvas in use, i.e. if the current canvas has width w and height h then the bottom-right position is $(w - 1, h - 1)$. Details of creating and accessing 2D arrays will be given in one of the lectures on indirection in LP4.

We now describe the different commands that your program must accept. Examples of how to use them follow later.

canvas w h starts a new picture. The width and height of the picture are supplied as arguments w and h which must be integer values.

pen c changes the character that is used to fill shapes. The new character is supplied as the argument c .

rectangle l t r b draws a rectangle on the canvas, it requires four integer arguments. The left l and right r columns indicate the horizontal range that should be filled with the pen character. The top t and bottom b indicate the vertical range. Remember the coordinate system means that $t < b$.

circle x y r draws a circle on the canvas. The x and y parameters are integers that give the center of the circle. The radius is specified by a float r . A character is filled if the distance from the position is less than or equal to the radius.

quit does not take any arguments, it finishes the program.

triangle1 x y w h dir draws a triangle on the canvas. The first four arguments are integers that specify the (x, y) position of the bottom-left corner of the triangle. The w parameter is the numbers of columns in width, and h is the number of columns in height. The *dir* parameter is a string with either the value “left” or the value “right”. The final parameter defines the direction of the slope. The triangles produced by this command are axis aligned — there is a flat horizontal base and one flat vertical side. The third side is a diagonal that slopes in one of the two ways indicated by the final parameter. Examples are shown later. Each position inside the triangle is only filled if the centroid (explained later) is within the boundary.

triangle2 x₁ y₁ w x₂ y₂ also draws a triangle, but with only one axis aligned side. Each argument is integer, one point in the triangle is at position (x_1, y_1) . There is a flat horizontal edge to the right of this point of length w so that the second point is at $(x_1 + w - 1, y_1)$. The third point is at (x_2, y_2) . Each position inside the triangle is only filled if the centroid (explained later) is within the boundary.

There are a number of problems that you must solve that will be common to all of the drawing operations: how will you represent the state of the canvas, how will you modify the canvas according to the pen ...etc. To implement the operations that are needed for the higher grades you must handle the centroids of each cell in the canvas. The centroids are 0.5 units higher than the integer coordinates of the cell. So for example, the cell at position (3,4) has a centroid at (3.5, 4.5). When implementing the *triangle1* and *triangle2* operations you need to decide if the centroid of each cell is inside the boundary of the shape. See figure 1 for an example of which cells should be filled for the triangle delimited by the red lines. The circle operation does not use centroids, you simply calculate the distance between the center coordinate and each cell coordinate and use the distance and radius to decide if the cell should be filled with the pen character.

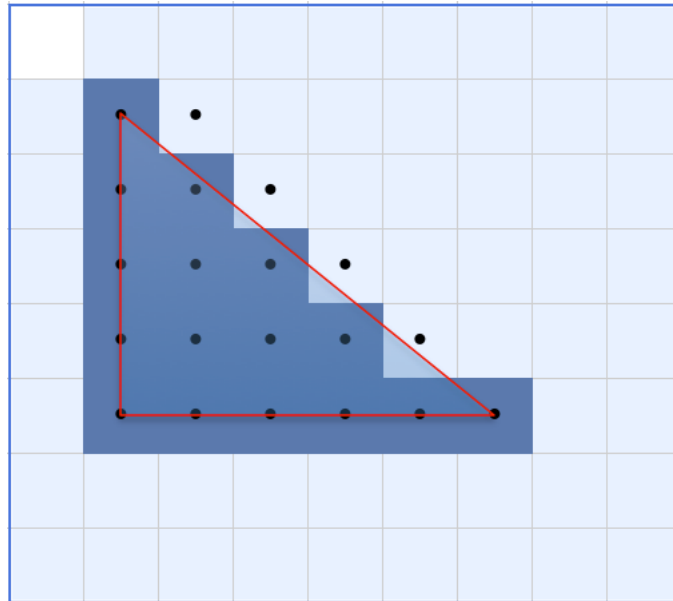


Figure 1: Triangle example

Unlike your first coursework the input and output specification is strict. Do not print unnecessary messages. Accept inputs in the format shown. Output data in the format shown.

2 Examples

A simple example of using the program to demonstrate the I/O loop.

```
Cmd> canvas 5 5
Cmd> pen X
Cmd> rectangle 0 1 2 3
```

```
XXX
XXX
XXX
```

```
Cmd> pen +
Cmd> rectangle 2 1 3 2
```

```
XX++
XX++
XXX
```

```
Cmd> quit
```

Your prompt must be exactly as shown, the commands must be accepted with exactly this spelling and case, the arguments must be separated by spaces.

The triangle cases are a little more complex. Here are some examples of these drawing commands.

```
Cmd> canvas 10 10
Cmd> pen X
Cmd> triangle1 2 9 8 5 left
```

```

X
XX
XXXX
XXXXXX
XXXXXXXX
Cmd> pen j
Cmd> triangle 2 9 8 5 right
```

```

X      j
XX     jj
XXXXjjjj
XXjjjjjj
jjjjjjjj
Cmd> pen .
Cmd> triangle2 2 2 6 5 4
```

```

.....
...
.
X      j
XX     jj
XXXXjjjj
XXjjjjjj
jjjjjjjj
```

3 Submission

You must email your source code to cw.bth.dv1445@gmail.com. The subject of your message *must* be your 10-digit person number (two year digits, month, date and the final four digits). We will use this to automatically file your message, if you do not set this properly your submission will not count. You must attach your source code to the email message, use your person-number with `.cpp` on the end as the file name.

We will check the submissions and you will receive a reply on Monday 15th April, either telling you that your submission was valid (and that it compiles ok), or that you must resubmit to fix a problem.

4 Marking Scale

These criteria are cumulative; to get a grade four you must fulfil the points listed under grade four and those listed under grade three. To get a grade five you must fulfil all of the points.

Grade 3

- A program that accepts commands and performs actions as described.
- A program that can correctly allocate storage for an arbitrary sized canvas.
- A program that correctly outputs the canvas state after a drawing command.
- A program that uses the input and output formats specified.
- A program that continues until the quit command is used.
- A program that implements the rectangle command as specified.
- A program that implements the circle command as specified.

Grade 4

- A program that implements the triangle1 command and calculates the correct set of positions to draw as specified.

Grade 5

- A program that implements the triangle2 command and calculates the correct set of positions to draw as specified.