|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected output | Output | Comment |
| 10  10  3  2  7  7  #  -1 | A single rectangle with a top left vertex of coordinates (3,2) and bottom right (7,7). The origin starts at (1,1) rather than (0,0) in the screenshots. |  | A single rectangle. |
| 10  10  0  6  4  9  @  -1 | A single rectangle with a top left vertex of (0,6) and a bottom right vertex of (4,9) |  | Drawing a new rectangle without deleting the previous file that was generated the previous one. |
| 10  10  3  2  7  7  #  0  6  4  9  @  -1 | The previous two rectangles printed to the same screen, with the second printed over the first. |  | Drawing both rectangles at the same time. |
| 50  50  3  2  7  7  #  0  6  4  9  @  30  30  40  45  ?  -1 | A large, 50 by 50 area with three rectangles in, the ones from the previous input as well as a large one in the bottom right corner |  | Testing a larger a varied area with another rectangle on, along with a new symbol |
| 10  10  3  2  10  10  #  -1 | “Rectangle 1 is invalid” | Plus “Rectangle 1 is invalid” | The exception is only thrown when the out of bounds index is reached so the first seven inputs are valid and can be printed to a file. After this an exception is thrown and the rectangle stops printing as it goes beyond the bounds of the matrix. |
| 10  10  0  10  7  7  #  -1 | With the first coordinate being invalid this time an exception should be thrown immediately without anything being drawn to the matrix | Plus the message “Rectangle 1 is invalid” | This works as said in the expected output |
| 10  10  10  0  7  7  #  -1 |
| 10  10  -1 | A blank grid with no errors |  | When no rectangles are given the matrix will print but it’s completely blank by default, the size of the drawing area is read in but there’s nothing in there. |
| 10  10  H  2  7  7  #  -1 | “Rectangle 1 is invalid” and a rectangles.txt with just a blank matrix printed to it. | “Rectangle 1 is invalid” | Checking that the validity check works for each value in the rectangle |
| 10  10  3  H  7  7  #  -1 |
| 10  10  3  2  H  7  #  -1 |
| 10  10  3  2  7  H  #  -1 |
| 10  10  3  2  7  7  漢  -1 | A single rectangle using 漢  As its drawing character |  | Just testing more unusual characters. |
| 10  0  3  2  7  H  #  -1 | “Your matrix has an invalid dimension”  No rectangles.txt produced | “Your matrix has an invalid dimension”  No rectangles.txt produced | Checking that exceptions involving the height and width are properly handled. |
| 0  10  3  2  7  H  #  -1 |
| H10  10  3  2  7  H  #  -1 | “Rectangle 1 is invalid”  And a blank rectangles.txt with a width of 3 and a height of 10 |  | The height is read as 10 then the width is read as 3 as the program begins reading in values as soon as it reaches a line with an integer as the first character. |
| 10  10  -3  2  7  7  #  -1 | “Rectangle 1 is invalid” and a blank output. | “Rectangle 1 is invalid” | “Rectangle 1 is invalid” is only thrown in the second instance of this test, with the 2 replaced with a -2. This happens because the conditions of the for loop are met immediately in the other cases, causing a blank rectangle to be drawn without any exception to be thrown. |
| 10  10  3  -2  7  7  #  -1 |
| 10  10  3  2  -7  7  #  -1 |
| 10  10  3  2  7  -7  #  -1 |
| 10  10  3  2  7  7  #  3  -2  8  8  %  3  2  7  7  $  -1 | A single rectangle made from # characters. The middle rectangle should not be read but if it is there will be % symbols present, if the final rectangle is read it will overwrite the first one.  “Rectangle 2 is invalid” | “Rectangle 2 is invalid” | Checking that the drawing loop breaks as soon as an invalid rectangle is detected. |
| 10  10  3  2  7  7  # | “Rectangle 2 is invalid” and a single rectangle drawn. | “Rectangle 2 is invalid” | If there is no negative character after the rectangle the program will continue to check for inputs, if the line is blank it will try and build a rectangle from this, which can’t be done, causing an exception to be thrown. Terminating the program. |
| 10  10  3  2  7  7  #  -2 | No problems, single rectangle drawn |  | Checking that negative numbers other than -1 work as expected in terminating the program. |