**Mario Problem 1:**

In a file called mario.c in a folder called mario-less, implement a program in C that recreates that pyramid, using hashes (#) for bricks, as in the below:

#



1. Declare a integer type variable h
2. Take input from user for the height of pyramid and store it in h
3. Apply a do while loop to prevent false inputs
4. initialize a variable j to 1
5. initialize a variable m equal to h
6. for user input times:
7. for l start at m, should be greater & = to 0, decrementing it every time by 1:
8. print “ ”
9. end loop
10. for i start at 0, should be smaller than j, incrementing it every time by 1:
11. print “#”
12. end loop
13. print “\n”
14. increment j by 1
15. Decrement m by 1
16. endloop



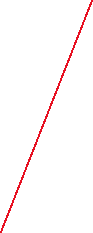
##



###



####



#####

######

#######

########



IMPROVED

1. Declare a integer type variable h
2. Take input from user for the height of pyramid and store it in h
3. Apply a do while loop to prevent false inputs
4. for loop initialize variable i to 1, smaller than user input h, incrementing it:
5. for initialize j to 0, smaller than h-i, incrementing by 1:
6. print “ ”
7. end loop
8. for initialize k, smaller than i, incrementing it:
9. print “#”
10. end loop
11. print “\n”
12. endloop

**Mario Problem 2:**

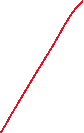
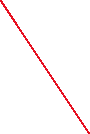
In a file called mario.c in a folder called mario-more, implement a program in C that recreates that pyramid, using hashes (#) for bricks, as in the below:



# #



## ##



### ###



#### ####



1. Initialize a variable h
2. Take input from the user for height of pyramid and store in h
3. Apply a do while loop to prevent false inputs
4. For loop initialize i to 1, smaller than = h, increment i
5. For loop initialize j to 0, smaller than height – i, increment by 1
6. Print “ ” // the space on right side
7. End loop
8. For initialize k, smaller than i, incrementing it:
9. print “#”
10. End loop
11. Print “ ” // the space between pyramids
12. Repeat step 7 to 9
13. Endloop