

Worksheet Two - Assignment FINAL

by Jake Richards

Submission date: 03-Dec-2025 12:00PM (UTC+0000)

Submission ID: 269434451

File name: 64894_Jake_Richards_Worksheet_Two_-_Assignment_FINAL_703116_1169223945.docx (41.25K)

Word count: 768

Character count: 3907

Name: _____

Student Number: _____

WORKSHEET 2 – TESTING (Tiles)

1. Original code:

[Copy and paste your original code from Moodle into this box. Do not change it.]

```
float length;
float width;

printf("How long is the room? ");
scanf("%f", &length);

printf("How wide is the room? ");
scanf("%f", &width);

while (length <= 0)
{
    printf("Please enter a length greater than 0cm: ");
    scanf("%f", &length);
}
while (width <= 0)
{
    printf("Please enter a width greater than 0cm: ");
    scanf("%f", &width);
}
float totalTileLength = length / 20;
float totalTileWidth = width / 20;

int area = ceil(totalTileLength) * ceil(totalTileWidth);
printf("\n\nYour total tiles for the room is: %d", area);
```

2. Code Testing

For each test below (i – vii), run your code and enter the values provided. Provide a screenshot of the of what happened when the programme ran. Did the programme behave correctly or did it do something unexpected, and if so, why?

Input:

- i. Enter dimensions: Width 40 Length 45
- ii. Enter dimensions: Width 100000 Length 200000
- iii. Enter dimensions: Width -16 Length -20

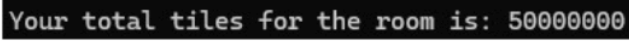
Logic:

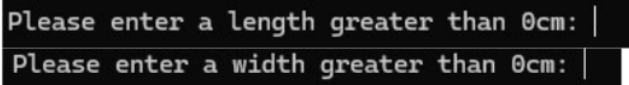
- iv. Enter dimensions: Width 80 Length 80. The result should be 16 tiles
- v. Enter dimensions: Width 40 Length 45. The result should be 6 tiles
- vi. Enter dimensions: Width 85 Length 90. The result should be 25 tiles.
- vii. Enter dimensions: Width 15 Length 15. The result should be 1 tile.

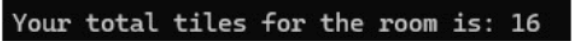
Note: If you decide to extend your program to allow a tile to be cut such that 2 pieces can be used from the same tile, your results may differ from the above. In this case, explain your logic as part of your answer)

Fill in a box for each test above. You should present 7 separate boxes, 1 for each test. (Note: the first box is provided as a template)

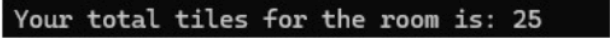
Test i: Input - Enter dimensions: width 40 length 45
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES Outputted the correct amount of full tiles that would be necessary for the room of the defined size.

Test ii: Input - Enter dimensions: width 100000 length 200000
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES Outputted the correct amount of full tiles that would be necessary for the room of the defined size.

Test iii: Input - Enter dimensions: width -16 length -20
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES Requires user to input correct length first before outputting to enter a width greater than 0cm.

Test iv: Input - Enter dimensions: Width 80 Length 80. The result should be 16 tiles
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES Outputted the correct amount of full tiles that would be necessary for the room of the defined size.

Test v: Input - Enter dimensions: Width 40 Length 45. The result should be 6 tiles
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES Outputted the correct amount of full tiles that would be necessary for the room of the defined size.

Test vi: Input - Enter dimensions: Width 85 Length 90. The result should be 25 tiles.
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES Outputted the correct amount of full tiles that would be necessary for the room of the defined size.

Test vii: Input - Enter dimensions: Width 15 Length 15. The result should be 1 tile.
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES Outputted the correct amount of full tiles that would be necessary for the room of the defined size.

3. Consider the results of the above tests and revisit your code in order to solve any problems discovered at the testing phase. (Note: you may not be able to solve all of them, but solve as many as you can.)

Paste your revised code here:

[Copy and paste your revised code here – note, do not use a screenshot]

```
void main()
{
    float length;
    float width;

    printf("How long is the room? ");
    scanf("%f", &length);

    printf("How wide is the room? ");
    scanf("%f", &width);

    while (length <= 0)
    {
        printf("Please enter a length greater than 0cm: ");
        scanf("%f", &length);
    }
    while (width <= 0)
    {
        printf("Please enter a width greater than 0cm: ");
        scanf("%f", &width);
    }
    float totalTileLength = length / 20;
    float totalTileWidth = width / 20;

    int area = ceil(totalTileLength) * ceil(totalTileWidth);
    printf("\n\nYour total tiles for the room is: %d", area);
}
```

Worksheet Two - Assignment FINAL

GRADEMARK REPORT

FINAL GRADE

7/10

GENERAL COMMENTS

WORKSHEET 2

Coding- Good, your algorithm is suitable. You could also consider that where there is less than half a tile in the calculation, the other half could be utilised (as the pattern on this particular tile can be used in any direction)and this would save a tile. Consider the validation.You could also consider validating for an upper limit 3.5/5

Testing - As commented on the last worksheet, please show more of a context in your screenshot, I cannot see the full picture with your screenshots. Test ii isn't really logically correct as you would be extremely unlikely to tile a wall that size, particularly with 20cm tiles.

3.5/5

Overall- 7/10

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5