

Test worksheet 2

by Josh Willis

Submission date: 03-Dec-2025 11:40AM (UTC+0000)

Submission ID: 269430956

File name: 58111_Josh_Willis_Test_worksheet_2_703116_670422312.docx (100.13K)

Word count: 844

Character count: 4288

Name: JOSHUA WILLIS

Student Number: 2405267

WORKSHEET 2 – TESTING (Tiles)

1. Original code:

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

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <math.h>

void main()
{
    float length = -3;
    float width = -3;
    float tileLength, tileWidth;
    int tiles;

    while (length < 0)
    {
        printf("Enter the length of your surface: ");
        scanf("%f", &length);
        if (length < 0)
        {
            printf("This length is invalid, please try again\n");
        }
        else
        {
            printf("Thank you\n");
        }
    }

    while (width < 0)
    {
        printf("Enter the width of your surface: ");
        scanf("%f", &width);
        if (width < 0)
        {
            printf("This width is invalid, please try again\n");
        }
        else
        {
            printf("Thank you\n");
        }
    }
}
```

```
    }  
  
    tileLength = length / 20;  
    tileWidth = width / 20;  
  
    tileLength = ceil(tileLength); tileWidth = ceil(tileWidth);  
  
    tiles = tileLength * tileWidth;  
  
    printf("%d", tiles);  
}
```

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:

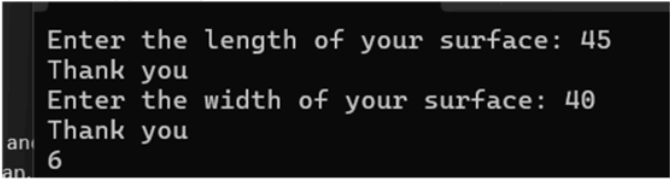
- i. Enter dimensions: Width 40 Length 45
- ii. Enter dimensions: Width 100000 Length 200000
- iii. Enter dimensions: Width -16 Length -20

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)

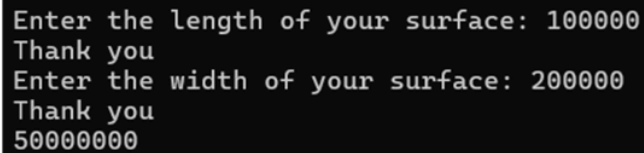
Test i: Input - Enter dimensions: width 40 length 45
Screenshot(s) of output: 
Result: Did the programme behave as expected ie syntactically and logically correct? YES This input came out correct and as expected for the result of the program

Continue the rest of the tests below. (Copy and paste a box for each one)

Test:

Input - Enter dimensions: Width 100000 Length 200000

Screenshot(s) of output:



```
Enter the length of your surface: 100000
Thank you
Enter the width of your surface: 200000
Thank you
50000000
```

Result: Did the programme behave as expected ie syntactically and logically correct?

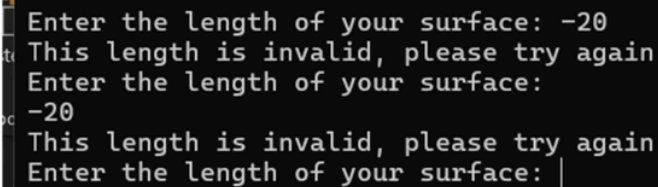
YES

Entering this test for the code came out all logically correct and behaved as planned

Test:

Input – enter dimensions: width -16 length -20

Screenshot(s) of output:



```
Enter the length of your surface: -20
This length is invalid, please try again
Enter the length of your surface:
-20
This length is invalid, please try again
Enter the length of your surface: |
```

Result: Did the programme behave as expected ie syntactically and logically correct?

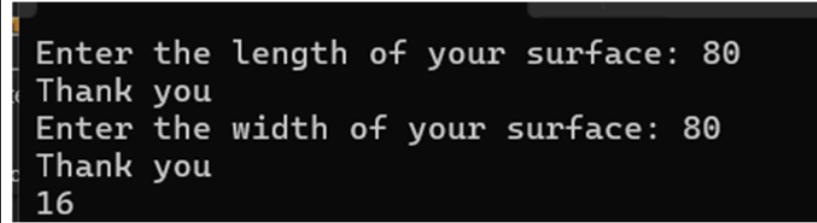
YES

The behaviour was correct to decline the negative number and continue to ask the user to retry and put a positive number in

Test:

Input – Enter dimensions: Width 80 Length 80. The result should be 16 tiles

Screenshot(s) of output:

A screenshot of a terminal window with a black background and white text. The text shows the program's interaction: it prompts for length, receives 80, says 'Thank you', prompts for width, receives 80, says 'Thank you' again, and finally outputs the result '16'.

```
Enter the length of your surface: 80
Thank you
Enter the width of your surface: 80
Thank you
16
```

Result: Did the programme behave as expected ie syntactically and logically correct?

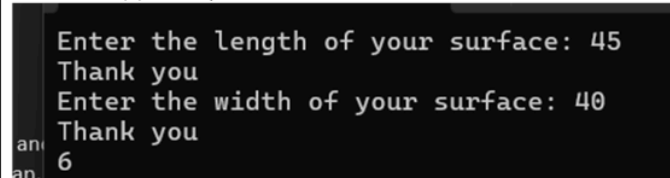
YES

All the calculations for this test are correct and the code produced a correct number of tiles

Test:

Input – Enter dimensions: Width 40 Length 45. The result should be 6 tiles

Screenshot(s) of output:

A screenshot of a terminal window with a black background and white text. The text shows the program's interaction: it prompts for length, receives 45, says 'Thank you', prompts for width, receives 40, says 'Thank you' again, and finally outputs the result '6'.

```
Enter the length of your surface: 45
Thank you
Enter the width of your surface: 40
Thank you
6
```

Result: Did the programme behave as expected ie syntactically and logically correct?

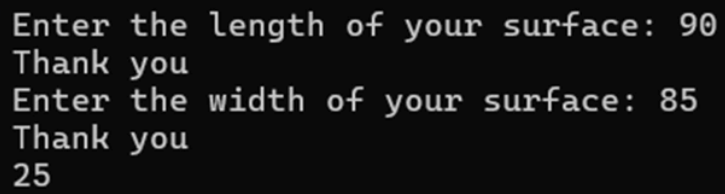
YES

This produced the correct result of tiles and the logic is all correct

Test:

Input – Enter dimensions: Width 85 Length 90. The result should be 25 tiles.

Screenshot(s) of output:



```
Enter the length of your surface: 90
Thank you
Enter the width of your surface: 85
Thank you
25
```

Result: Did the programme behave as expected ie syntactically and logically correct?

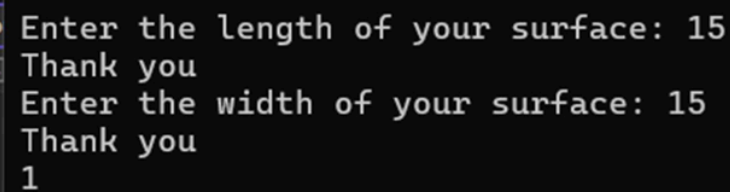
YES

All the processes and the calculation worked as the should to end with the correct resulting number of tiles for the give surface area input

Test:

Input – Enter dimensions: Width 15 Length 15. The result should be 1 tile.

Screenshot(s) of output:



```
Enter the length of your surface: 15
Thank you
Enter the width of your surface: 15
Thank you
1
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

The code decided that 1 tile was needed and this is correct for the little surface area which is needing to be filled with the tile, all logic and syntax worked perfectly

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]

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <math.h>

void main()
{
    float length = -3;
    float width = -3;
    float tileLength, tileWidth;
    int tiles;

    while (length < 0)
    {
        printf("Enter the length of your surface: ");
        scanf("%f", &length);
        if (length < 0)
        {
            printf("This length is invalid, please try again\n");
        }
    }
    while (width < 0)
    {
        printf("Enter the width of your surface: ");
        scanf("%f", &width);
        if (width < 0)
        {
            printf("This width is invalid, please try again\n");
        }
    }
    tileLength = length / 20;
    tileWidth = width / 20;

    tileLength = ceil(tileLength); tileWidth = ceil(tileWidth);

    tiles = tileLength * tileWidth;

    printf("You will need %d tiles", tiles);
}
```

Test worksheet 2

GRADEMARK REPORT

FINAL GRADE

8/10

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 (Test ii shows this) 3.5/5

Testing - Test ii is not really logically correct - it would be very unlikely that you'd tile such a massive wall with 20cm tiles. Otherwise a good discussion with suitable screenshots

4.5/5

Overall - 8/10

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6
