

Aidan Griffiths - Worksheet 2

by Aidan Griffiths

Submission date: 02-Dec-2025 12:26AM (UTC+0000)

Submission ID: 269287231

File name: 66131_Aidan_Griffiths_Aidan_Griffiths_-_Worksheet_2_703116_130528115.docx (111.09K)

Word count: 819

Character count: 4275

Name: Aidan Griffiths

Student Number: 2501271

WORKSHEET 2 – TESTING (Tiles)

1. Original code:

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

void main()
{
    float length;
    float height;
    int tiles;
    float tilelength;
    float tileheight;

    printf("Welcome!\n");
    printf("Please enter the length: \n");
    scanf("%f", &length);
    printf("Please enter the height: \n");
    scanf("%f", &height);

    if (length > 0 && height > 0)
    {
        tilelength = length / 20;
        tileheight = height / 20;

        tilelength = ceil(tilelength);
        tileheight = ceil(tileheight);

        tiles = (tilelength * tileheight);

        printf("You will need to purchase %d tiles\n", tiles);
    }
    else
    {
        printf("Your wall must be greater than 0cm in height and length...");
    }
}
```


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.

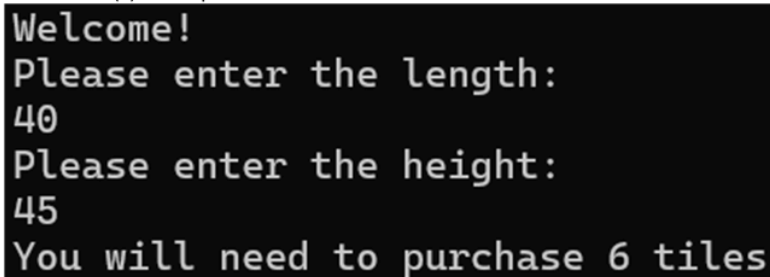
Note: If you decide to extend your program to allow a tile to be cut such that 2 pieces can be used from the same tile, your results may differ from the above. In this case, explain your logic as part of your answer)

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:



```
Welcome!
Please enter the length:
40
Please enter the height:
45
You will need to purchase 6 tiles
```

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

YES

If NO, what went wrong?

If YES, comment on what happened

The program calculated 6 tiles required for a 40x45 wall

Test ii:

Input - **Enter dimensions: Width 100000 Length 200000**

Screenshot(s) of output:

```
Welcome!
Please enter the length:
100000
Please enter the height:
200000
You will need to purchase 50000000 tiles
```

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

YES

If NO, what went wrong?

If YES, comment on what happened

The program calculated 50000000 tiles required for a 100000x200000 wall

Test iii:

input - **Enter dimensions: Width -16 Length -20**

Screenshot(s) of output:

```
Welcome!
Please enter the length:
-16
Please enter the height:
-20
Your wall must be greater than 0cm in height and length...
```

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

YES

If NO, what went wrong?

If YES, comment on what happened

The program did not accept the entered values as a wall cannot be measured in negative values.

Test i:
Logic - Enter dimensions: Width 80 Length 80. The result should be 16 tiles

Screenshot(s) of output:

```
Welcome!
Please enter the length:
80
Please enter the height:
80
You will need to purchase 16 tiles
```

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

YES/NO

If NO, what went wrong?

If YES, comment on what happened

The program calculated 16 required tiles for an 80x80 wall which is a logically correct result

Test ii:
Logic - Enter dimensions: Width 40 Length 45. The result should be 6 tiles

Screenshot(s) of output:

```
Welcome!
Please enter the length:
40
Please enter the height:
45
You will need to purchase 6 tiles
```

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

YES

If NO, what went wrong?

If YES, comment on what happened

The program calculated that 6 tiles are required for a 40x45 wall which is logically correct

Test iii:

Logic - Enter dimensions: Width 85 Length 90. The result should be 25 tiles

Screenshot(s) of output:

```
Welcome!
Please enter the length:
85
Please enter the height:
90
You will need to purchase 25 tiles
```

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

YES

If NO, what went wrong?

If YES, comment on what happened

The program calculated that 25 tiles would be required for a 85x90 wall which is logically correct

Test iv:

Logic - Enter dimensions: Width 15 Length 15. The result should be 1 tile

Screenshot(s) of output:

```
Welcome!
Please enter the length:
15
Please enter the height:
15
You will need to purchase 1 tiles
```

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

YES

If NO, what went wrong?

If YES, comment on what happened

The program calculated that 1 tile would be required for a 15x15 wall which is logically correct

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:

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

void main()
{
    float length;
    float height;
    int tiles;
    float tilelength;
    float tileheight;

    printf("Welcome!\n");
    printf("Please enter the length in cm: \n");
    scanf("%f", &length);
    printf("Please enter the height in cm: \n");
    scanf("%f", &height);

    if (length > 0 && height > 0)
    {
        tilelength = length / 20;
        tileheight = height / 20;

        tilelength = ceil(tilelength);
        tileheight = ceil(tileheight);

        tiles = (tilelength * tileheight);

        printf("You will need to purchase %d tiles\n", tiles);
    }
    else
    {
        printf("Your wall must be greater than 0cm in height and length...");
    }
}
```

Aidan Griffiths - Worksheet 2

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

WORKSHEET 2

Coding- There doesn't seem to be very much progress between the initial code and the final code You could consider including a while loop to give the user another go at entering valid data, as it is they would have to re-run the program to do this. You could also consider an upper limit on wall dimensions. Other than that, the algorithm is 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.

3/5

Testing - Test ii is not really a YES logically, it would be very unlikely that you'd tile such a massive wall with 20cm tiles. Other than that, suitably laidout and discussed (but see by comment about using a while above)

4/5

Overall - /10

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7
