19CSE102 Computer Programming

Practice 1: Golden Plate

# Problem 1031A-1

You have a rectangular plate, and you want to add some gilding to it along the bordering cells so that it forms a golden ring. The plate is a rectangle that we split into w x h cells. Your task is to compute the number of cells gilded.

# Input

Two integers w, h such that 1 ≤ w, h ≤ 1000, the number of rows and columns.

# Output

Print the number of cells gilded.

# Example

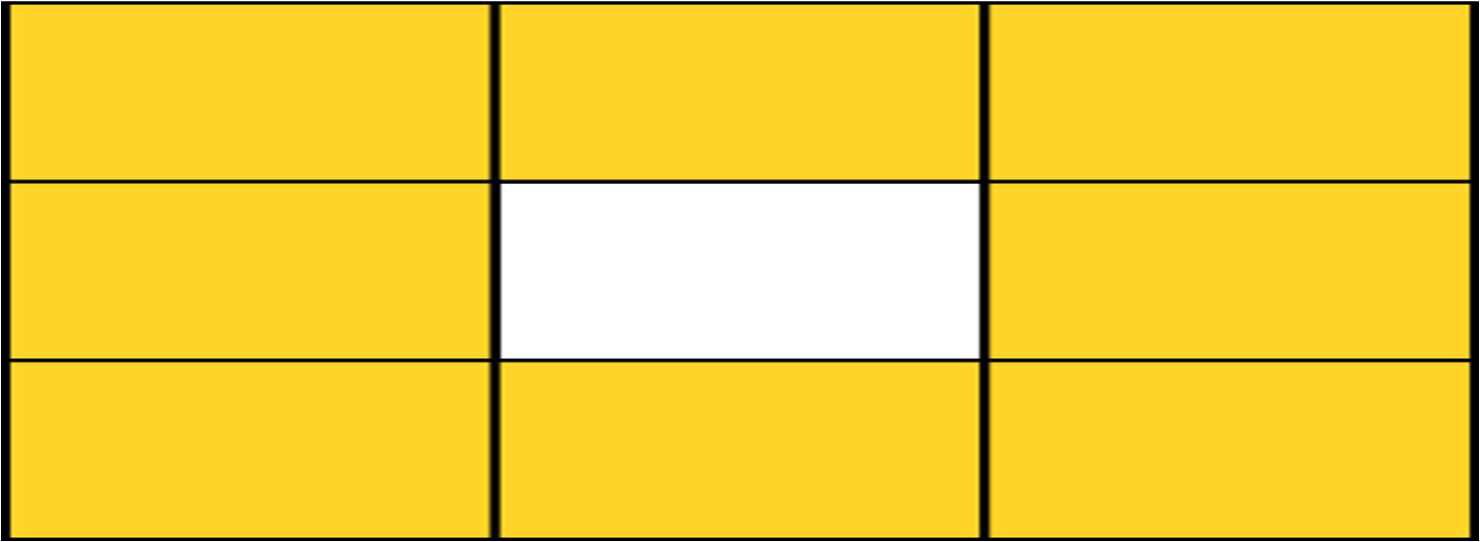
Input 3 3

Output 8

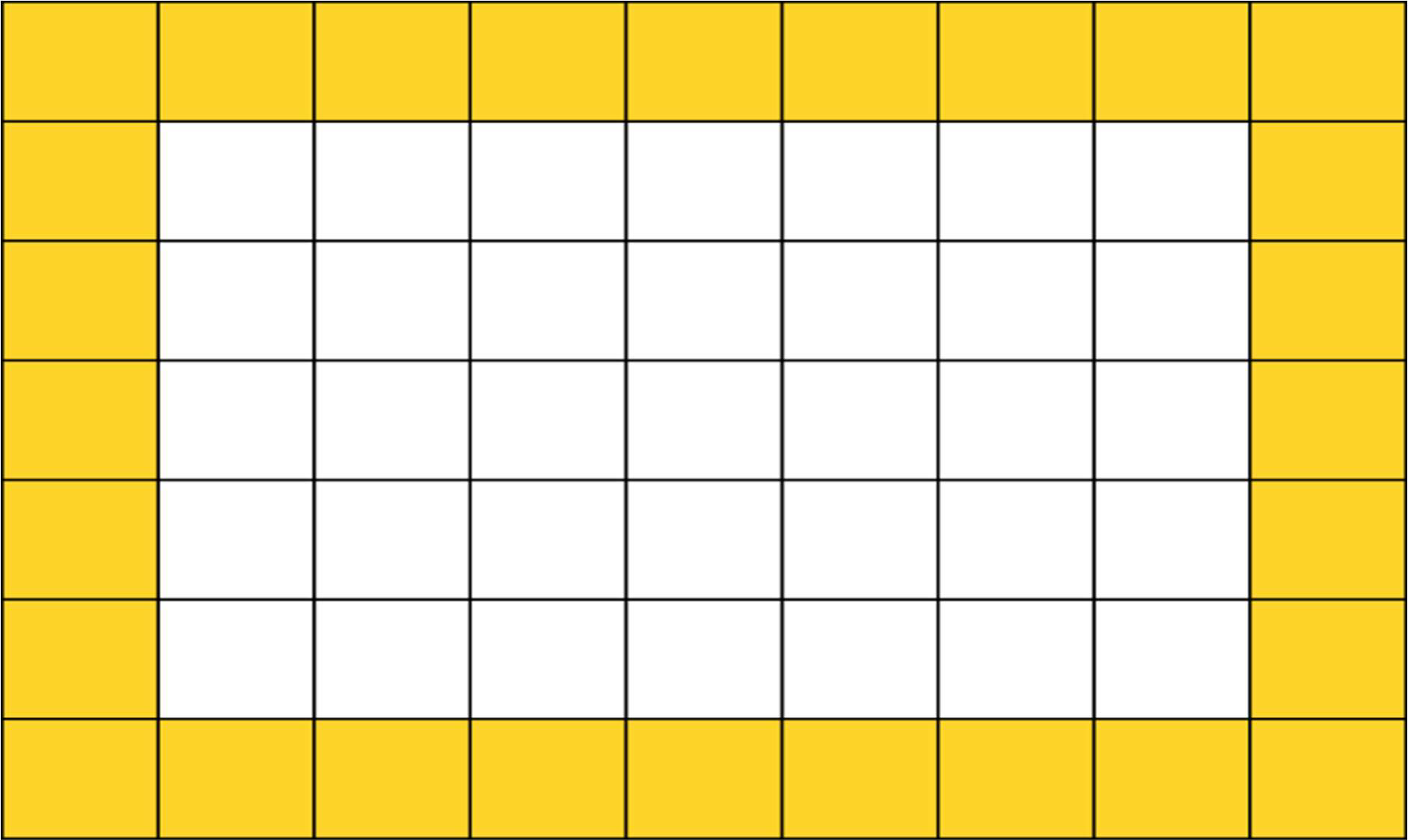
Input 7 9

Output 28

# Note

The examples are described below.

First example: 3 x 3 rectangle - 8 cells gilded



Second example: 7 x 9 rectangle - 28 cells gilded

# Answer the following questions.

1. **Describe your solution approach.**

Add the w and h-2 as 2 is the intersection between the rows and columns and multiply it by 2(formula: perimeter of rectangle). We will get the plated rectangles numbers.

# What would be your implementation considerations (if any)?

# Remember to reduce 2 and use perimeter of rectangle formula.

1. **Provide your implementation.**

**Note**: You need to implement your computation logic in a separate function **gild\_ring**. Do not implement the entire program in main. It will attract negative marks.

#include<stdio.h>

int gild\_ring(int w,int h)

{

return (2\*(w+h));

}

int main()

{

int w,h;

printf("Enter number of rows ");

scanf("%d",&w);

printf("Enter number of columns ");

scanf("%d",&h);

printf("The number of cells to be gilded is %d",gild\_ring(w,h-2));

return 0;

}

# List the test cases.

# None

**Problem 1031A-2**

A modified version of the problem is to draw 2 golden rings in such a way that the first one goes along the edge of the plate and the second one 2 cells way from the edge (i.e., they do not touch each other). You must compute the number of cells gilded.

# Input

Two integers w, h such that 1 ≤ w, h ≤ 1000, the number of rows and columns.

# Output

Print the number of cells gilded.

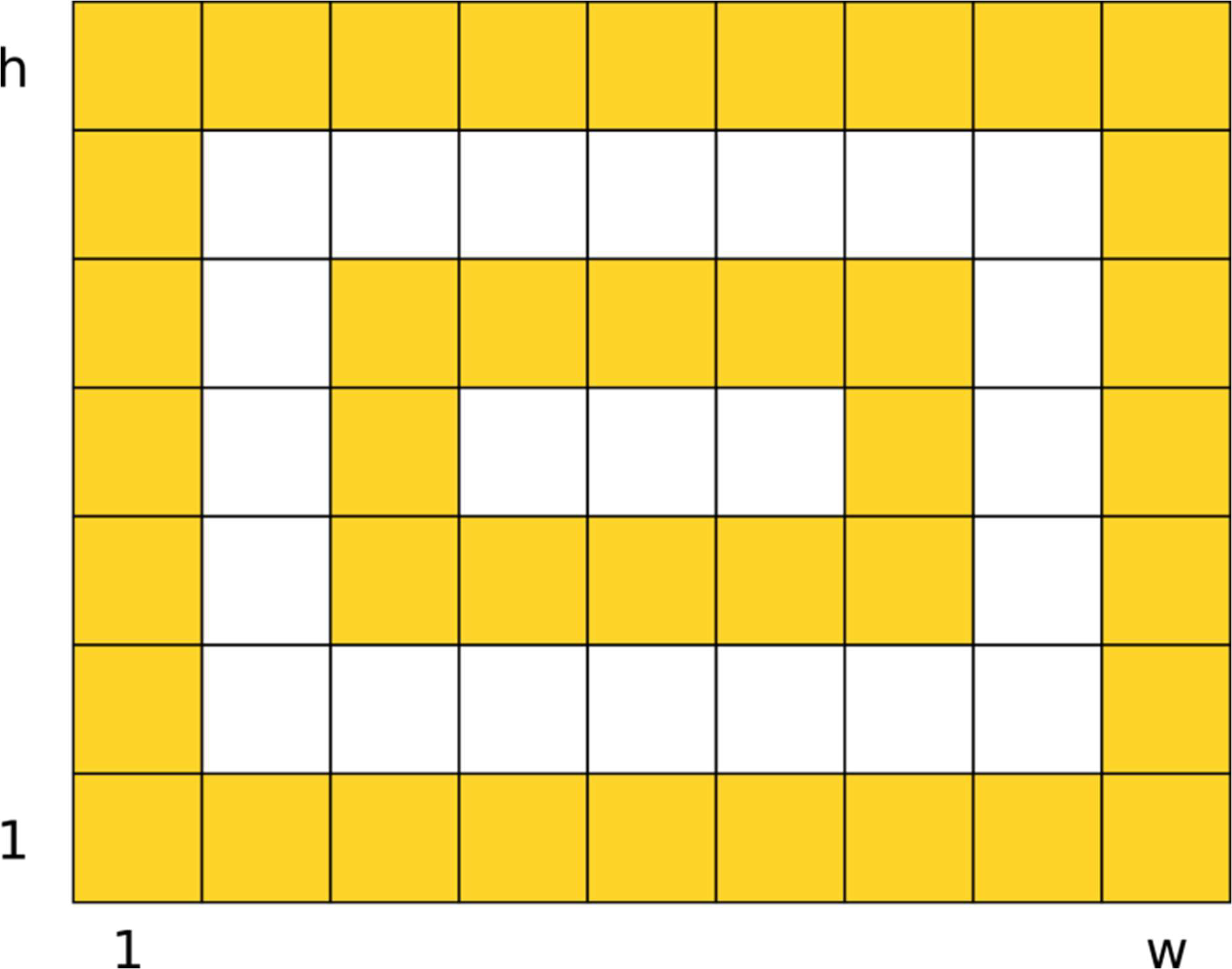
# Example

Input 7 9

Output 40

# Note

The example is described below.



7 x 9 rectangle - 28 + 12 cells gildes

4 rows and 4 columns are removed in this case at once . So I will decrease 4 to the value of w and h to get the value.

# Provide your modfied implementation.

**Note**: You need to reuse the function **gild\_ring** (that you have already defined) for the computation of second ring.

#include<stdio.h>

int gild\_ring(int w,int h)

{

return (2\*(w+h)+2\*((w-4)+(h-4)));

}

int main()

{

int w,h;

printf("Enter number of rows ");

scanf("%d",&w);

printf("Enter number of columns ");

scanf("%d",&h);

printf("The number of cells to be gilded is %d",gild\_ring(w,h-2));

return 0;

}

# Add more test cases to the list.

# None

**Problem 1031A-3**

A further modification of the problem is to draw k golden rings in such a way that the first one goes along the edge of the plate and the second one goes 2 cells way from the edge, the third one goes 2 cells away from the second one, so on and so forth. You must compute the number of cells gilded.

# Input

Three integers w, h, k such that 1 ≤ w, h ≤ 1000, 1 ≤ 𝑘 ≤ ⌊(min(w, h)+1)/4⌋, the number of rows and columns and the number of gilded rings.

# Output

Print the number of cells gilded.

# Example

Input 3 3 1

Output 8

Input 7 9 1

Output 28

Input 7 9 2

Output 40

**Answer the following questions.**

**1. How would you modify your previous solution?**

Used a for loop to run the program k number of times.

# Provide your modified implementation.

**Note**: You need to reuse the function **gild\_ring** for the computation of k rings.

#include<stdio.h>

int gild\_ring(int w,int h,int k)

{

int c=0;

for(int i=0;i<k;i++)

{

c=c+2\*(w+(h-2));

w =w-4;

h =h-4;

}

return c;

}

int main()

{

int w,h,k;

scanf("%d",&w);

scanf("%d",&h);

scanf("%d",&k);

printf("%d",gild\_ring(w,h,k));

return 0;

}

# Improvise your test cases.

# For(i=0;i<k;i++) was used to test whether we are getting k golden rings or not.

**Submission**

* 1. Check out the problem **1031A Golden**

**Plate** in https://codeforces.com/problemset/problem/1031/A

* 1. After checking your solution thoroughly, submit your program and get an **Accepted** message.