**C# SET 2**

**NAME: Mahasri M**

1. A farmer is asking you to tell him how many legs can be counted among all his animals. The

farmer breeds three species:

chickens = 2 legs

cows = 4 legs

pigs = 4 legs

The farmer has counted his animals and he gives you a subtotal for each species. You have to

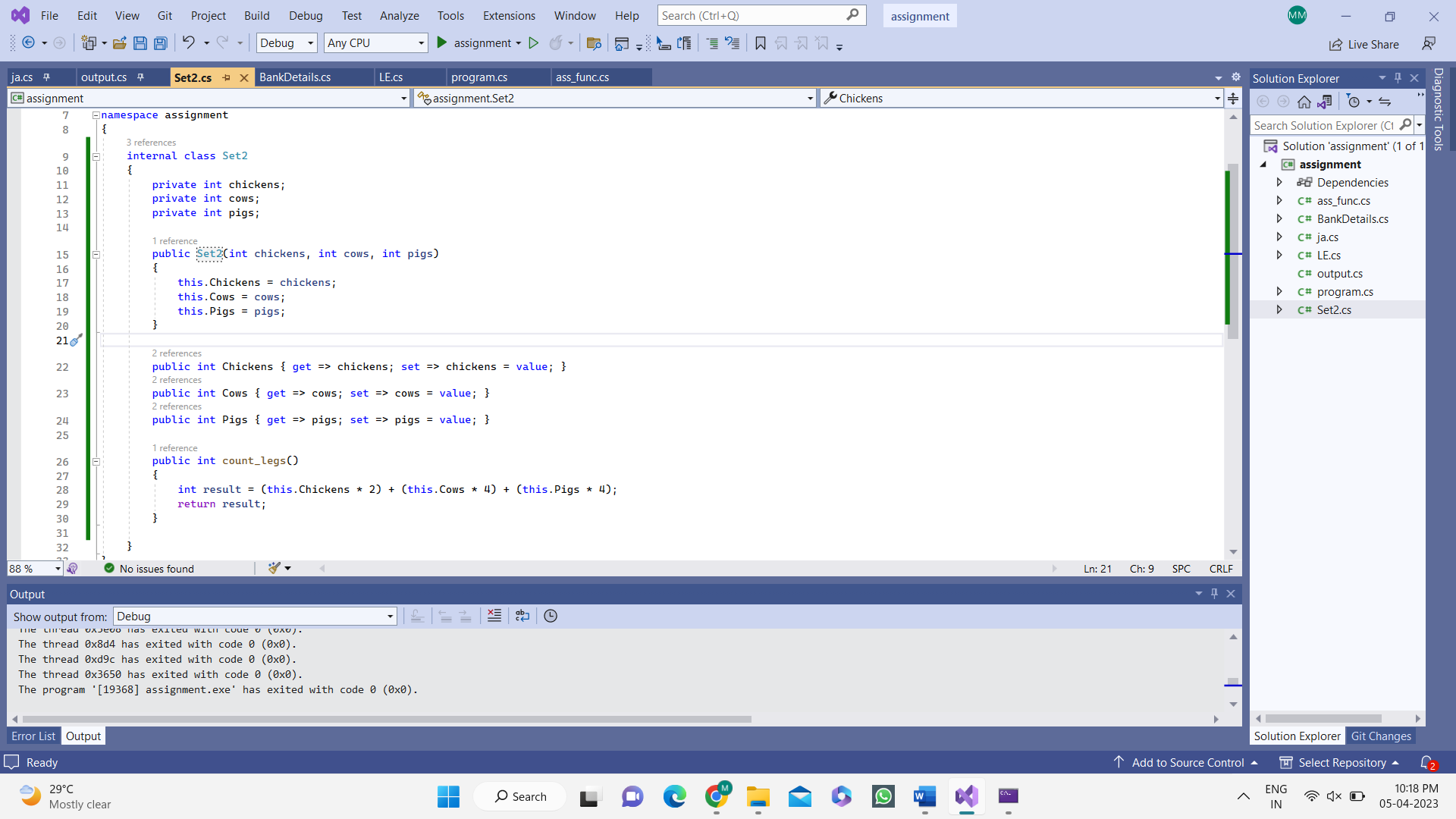
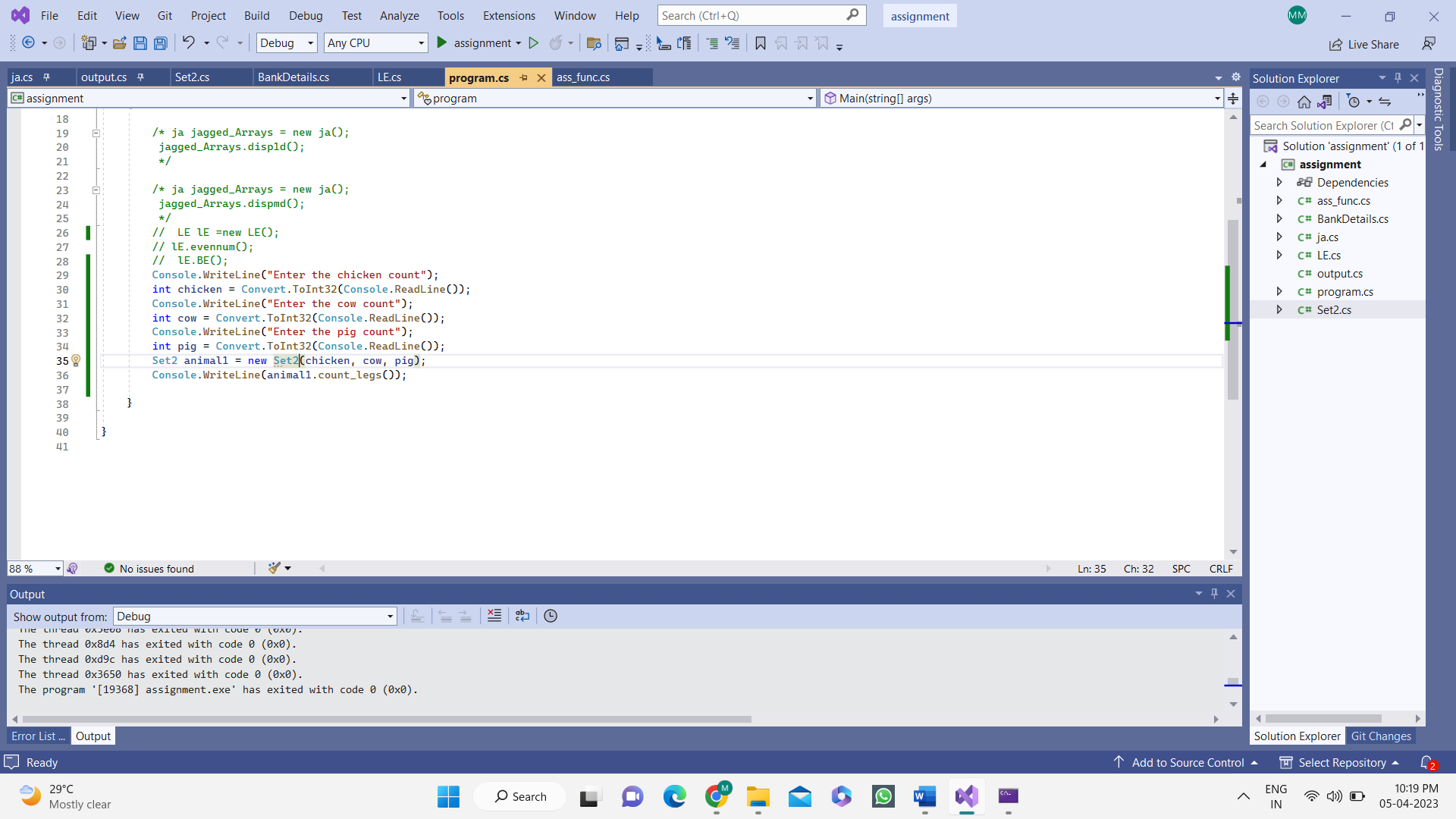
implement a function that returns the total number of legs of all the animals.

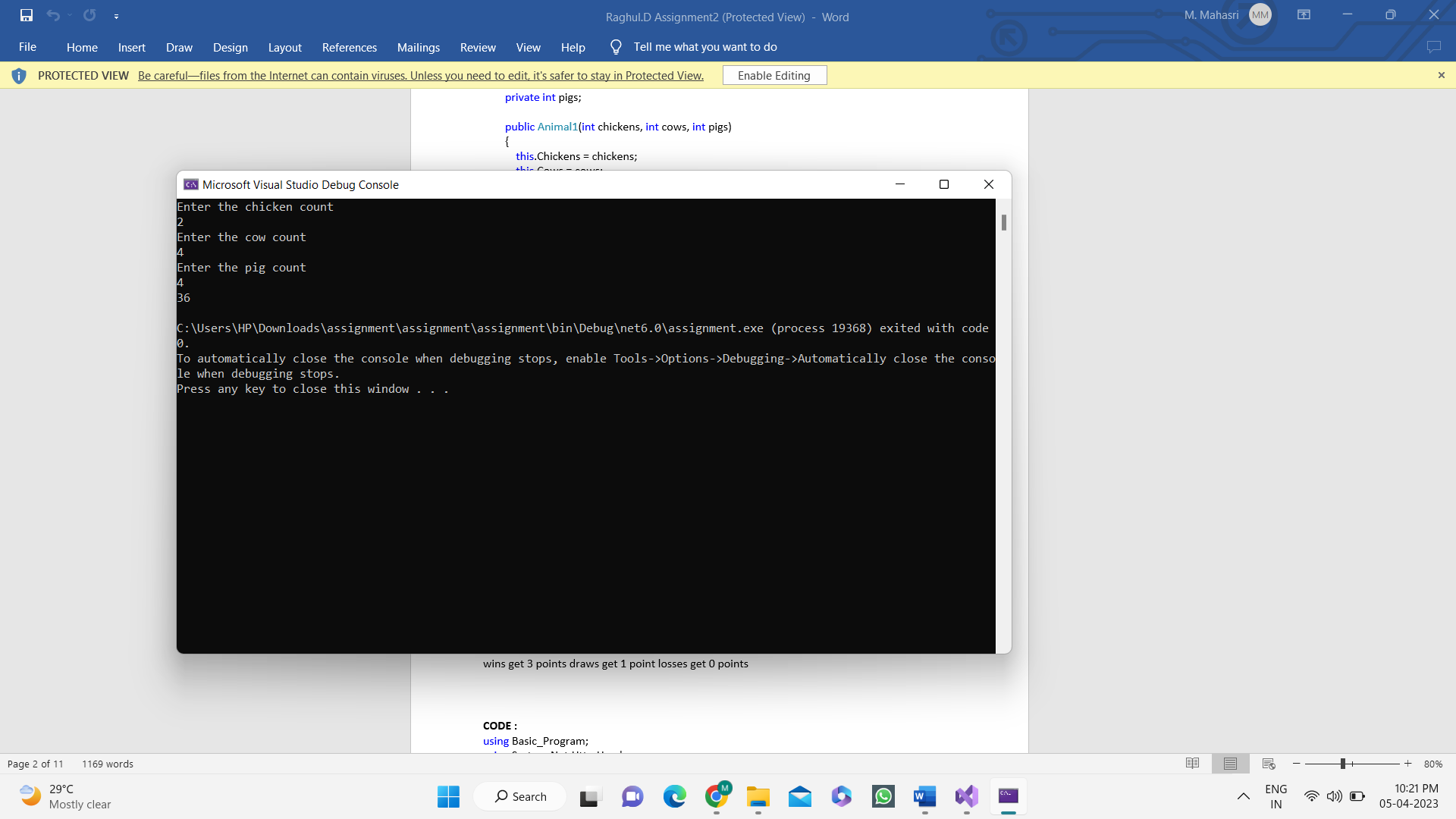
Example Input Output

animals(2, 3, 5) ➞ 36

animals(1, 2, 3) ➞ 22

animals(5, 2, 8) ➞ 50

**CODE:**

**OUTPUT:**

2. Create a function that takes the number of wins, draws and losses and calculates the number of

points a football team has obtained so far.

wins get 3 points draws get 1 point losses get 0 points

Examples

FootballPoints(3, 4, 2) ➞ 13

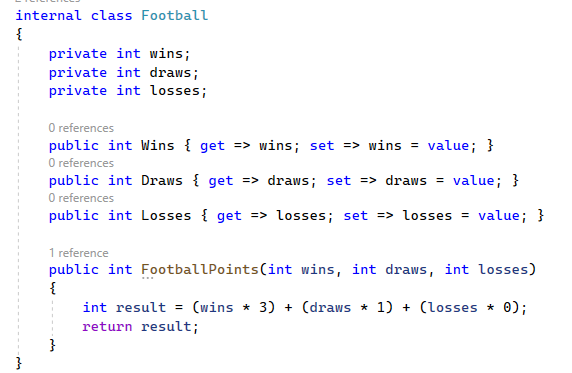
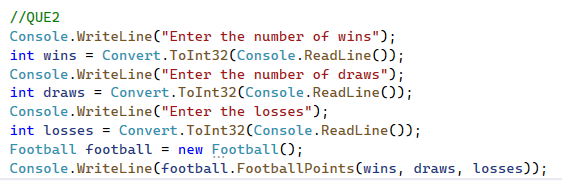
FootballPoints(5, 0, 2) ➞ 15

FootballPoints(0, 0, 1) ➞ 0

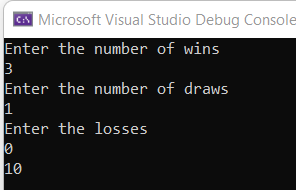
Notes

Inputs will be numbers greater than or equal to 0.

**CODE:**

****

**OUTPUT:**

****

3. Create a function that takes three arguments prob, prize, pay and returns true if prob \* prize &gt; pay;

otherwise return false.

To illustrate:

ProfitableGamble(0.2, 50, 9) should yield true, since the net profit is 1 (0.2 \* 50 - 9), and 1 &gt; 0.

Examples

ProfitableGamble(0.2, 50, 9) ➞ true

ProfitableGamble(0.9, 1, 2) ➞ false

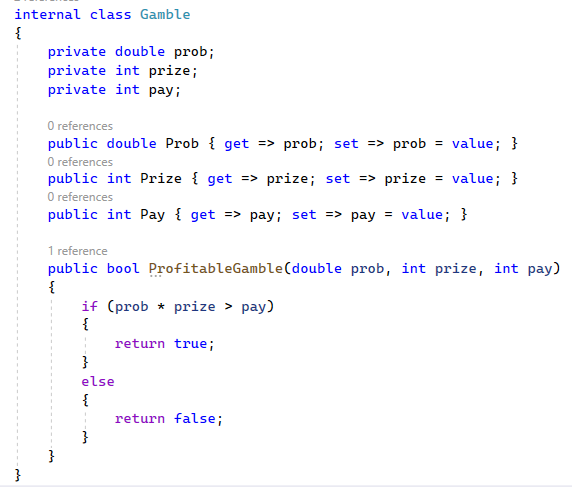
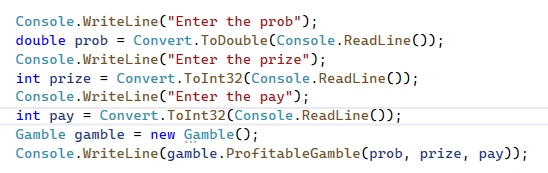
ProfitableGamble(0.9, 3, 2) ➞ true

Notes

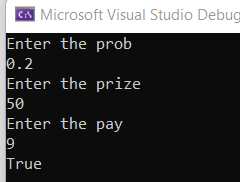
A profitable gamble is a game that yields a positive net profit, where net profit is calculated in the

following manner: net\_outcome = probability\_of\_winning \* prize - cost\_of\_playing.

**CODE:**

****

**OUTPUT:**



4. Here&#39;s an image of four models. Some of the cubes are hidden behind other cubes. Model one

consists of one cube. Model two consists of four cubes, and so on...

Write a function that takes a number n and returns the number of stacked boxes in a model n levels

high, visible and invisible.

Examples

StackBoxes(1) ➞ 1

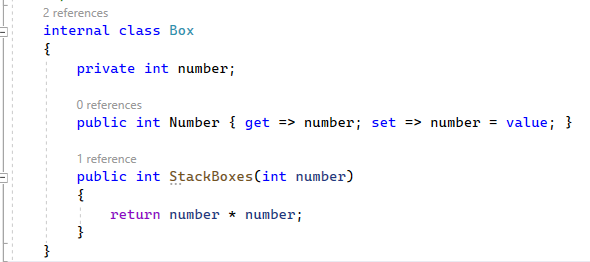
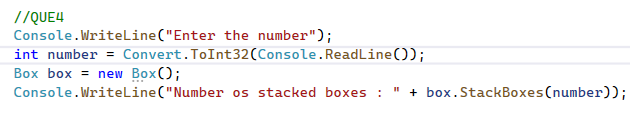
StackBoxes(2) ➞ 4

StackBoxes(0) ➞ 0

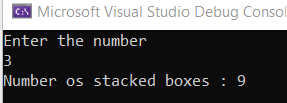
Notes

Step n is a positive integer.

**CODE:**



**OUTPUT:**

****

5. A bartender is writing a simple program to determine whether he should serve drinks to someone.

He only serves drinks to people 18 and older and when he&#39;s not on break.

Given the person&#39;s age, and whether break time is in session, create a function which returns whether

he should serve drinks.

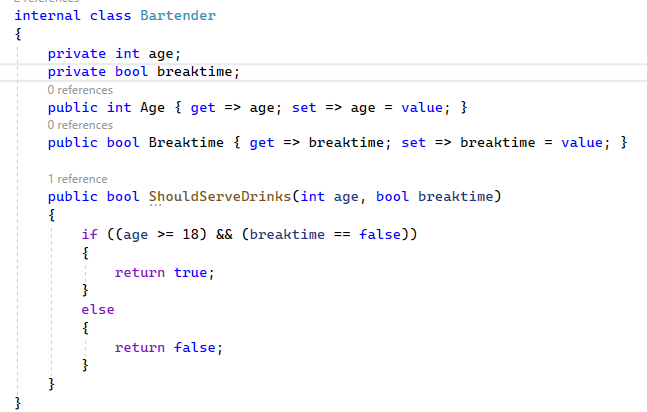
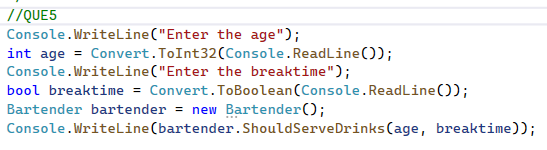
Examples

ShouldServeDrinks(17, true) ➞ false

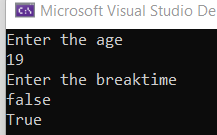
ShouldServeDrinks(19, false) ➞ true

ShouldServeDrinks(30, true) ➞ false

**CODE:**

****

**OUTPUT:**

****

6. For each of the 6 coffee cups I buy, I get a 7th cup free. In total, I get 7 cups. Create a function

that takes n cups bought and return the total number of cups I would get.

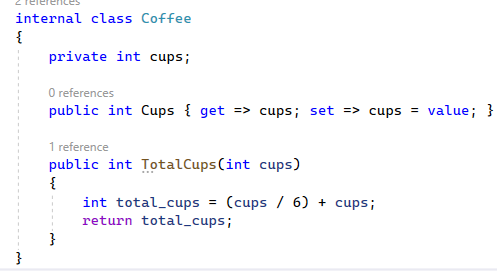
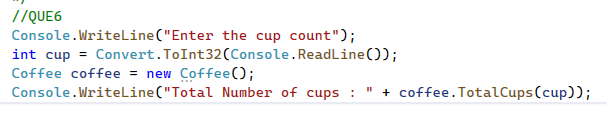
Examples

TotalCups(6) ➞ 7

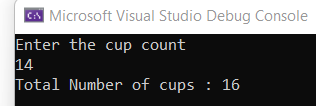
TotalCups(12) ➞ 14

TotalCups(213) ➞ 248

**CODE:**

****

**OUTPUT:**

****

7. Create a function that adds a string ending to each member in an array.

Examples

AddEnding([&quot;clever&quot;, &quot;meek&quot;, &quot;hurried&quot;, &quot;nice&quot;], &quot;ly&quot;)

➞ [&quot;cleverly&quot;, &quot;meekly&quot;, &quot;hurriedly&quot;, &quot;nicely&quot;]

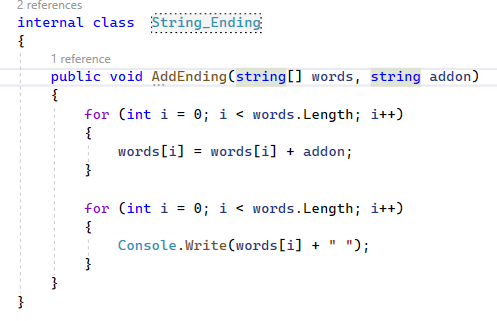
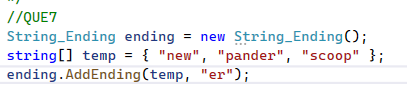
AddEnding([&quot;new&quot;, &quot;pander&quot;, &quot;scoop&quot;], &quot;er&quot;)

➞ [&quot;newer&quot;, &quot;panderer&quot;, &quot;scooper&quot;]

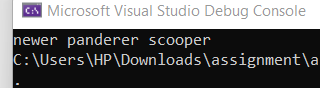
AddEnding([&quot;bend&quot;, &quot;sharpen&quot;, &quot;mean&quot;], &quot;ing&quot;)

➞ [&quot;bending&quot;, &quot;sharpening&quot;, &quot;meaning&quot;]

**CODE:**

****

**OUTPUT:**

****

8. Create a function that returns how many possible arrangements can come from a certain number

of switches (on / off). In other words, for a given number of switches, how many different

patterns of on and off can we have?

Examples

PosCom(1) ➞ 2

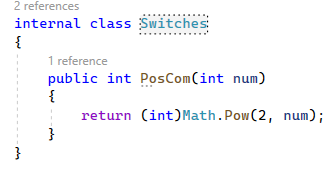
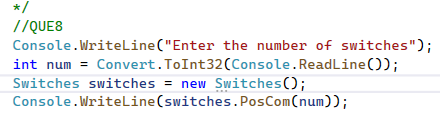
PosCom(3) ➞ 8

PosCom(10) ➞ 1024

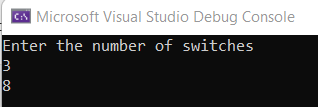
Notes

All numbers will be whole and positive.

**CODE:**

****

**OUTPUT:**

****