

```
tomato      → m = 3 .The size of the soldPrice
3           array
15.00       → soldPrice =[15, 300.90,10.00]
300.90
10.00
```

Sample Output 1

```
1
```

Explanation 1

Given the following:

- `products = ["chocolate", "cheese", "tomato"]`
  - `productPrices = [15.00, 300.90, 23.44]`
  - `productSold = ["chocolate", "cheese", "tomato"]`
  - `soldPrice = [15.00, 300.90, 10.00]`
- Only the *tomato* sale does not match the price list. So, the number of sale prices that were entered incorrectly is 1.

Question - 2  
The Robot Class

We need a *Robot* class that can move around on a two dimensional plane. It needs to be able to change its position, report its position and report its last move as described below. Implement a *Robot* class per the following specifications:

Fields		
Data Type	Name	Description
integer	currentX	The robot's current x-coordinate in the 2D plane.
integer	currentY	The robot's current y-coordinate in the 2D plane.
integer	previousX	The robot's x-coordinate in the 2D plane prior to its most recent movement.
integer	previousY	The robot's y-coordinate in the 2D plane prior to its most recent movement.
Note: The robot's initial location is at (x, y) coordinate (0, 5).		

Parameterized Constructor			

Data Type	Param. Name	Description
integer	x	The value of <i>currentX</i> for the new Robot.
integer	y	The value of <i>currentY</i> for the new Robot.
<b>Note:</b> The robot created by this constructor is considered to have spawned at <i>(0, 5)</i> and moved to <i>(currentX, currentY)</i> so <i>(previousX, previousY)</i> starts as <i>(0, 5)</i> .		

Methods				
Return Type	Method Name	Param. Type	Param. Name	Description
void	moveX	integer	dx	Move the robot from current position <i>(x, y)</i> to new position <i>(x + dx, y)</i> . Remember to maintain <i>previousX</i> .
void	moveY	integer	dy	Move the robot from current position <i>(x, y)</i> to new position <i>(x, y + dy)</i> . Remember to maintain <i>previousY</i> .
void	printCurrentCoordinates			Print two space-separated integers describing the robot's current x and y coordinates.

	void	printLastCoordinates	Print two space-separated integers describing the robot's <i>previousX</i> and <i>previousY</i> coordinates. This will be called after the robot has moved from position $(0, 5)$ at least once.
	void	printLastMove	no parameters Print two space-separated values describing the robot's most recent movement: <ul style="list-style-type: none"> <li>If the last move was <i>move <math>X(dx)</math></i>, print <i>x dx</i> where <i>x</i> is the actual character <i>x</i> and <i>dx</i> is the distance moved in the <i>x</i>-direction</li> </ul>

			durin g the last call to <i>move</i> X. • If the last move was <i>move</i> Y( <i>dy</i> ), print <i>y dy</i> wher e <i>y</i> is the actua l chara cter <i>y</i> and <i>dy</i> is the dista nce move d in the <i>y</i> - direc tion durin g the last call to <i>move</i> Y.
--	--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The code provided has a complete definition for a main method that creates *Robot* objects and tests the class' methods. Implement the *Robot* class according to the criteria above to pass all test cases.

#### Constraints

- $-100 \leq x, y, dx, dy \leq 100$

#### ▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the *Robot* class' constructor and methods.

The first line contains an integer *x*.  
 The second line contains an integer *y*.  
 The next line contains an integer *dx*.  
 The last line contains an integer *dy*.

**Sample Input 0**

```
2
1
1
1
```

**Sample Output 0**

```
0 5
2 1
x 1
3 1
3 1
x 2
5 2
5 2
```

**Explanation 0**

- The *firstRobot* object is initially at position  $(0, 5)$ , so the call to *firstRobot.printCurrentCoordinates()* prints *0 5*.
- For the *secondRobot* object created with the parameterized constructor, it was created and moved from  $(0, 5) \rightarrow (2, 1)$  so *secondRobot.printCurrentCoordinates()* prints *2 1*. Next, we call the following sequence of methods on the *secondRobot* object:
  1. *secondRobot.moveX(1)* moves the robot 1 unit from  $(2, 1) \rightarrow (3, 1)$ .
  2. *secondRobot.printLastMove()* prints *x 1* as its last movement was *moveX(1)*.
  3. *secondRobot.printCurrentCoordinates()* prints *3 1* because it moved from  $(2, 1) \rightarrow (3, 1)$ .
  4. *secondRobot.moveY(1)* moves the robot 1 unit from  $(3, 1) \rightarrow (3, 2)$ .
  5. *secondRobot.printLastCoordinates()* prints *3 1* because its last movement was  $(3, 1) \rightarrow (3, 2)$ , so the coordinates of its last location prior to the movement was  $(3, 1)$ .
    - At this point, test code adds 1 to  $dx \Rightarrow dx = 2$  and subtracts 1 from  $dy \Rightarrow dy = 0$ .
  6. *secondRobot.moveX(2)* moves the robot from the  $(3, 2) \rightarrow (5, 2)$ .
  7. *secondRobot.printLastMove()* prints *x 2* as its last movement was *moveX(2)*.
  8. *secondRobot.printCurrentCoordinates()* prints *5 2* because it moved from  $(3, 2) \rightarrow (5, 2)$ .
  9. *secondRobot.moveY(0)* moves the robot 0 units from  $(5, 2) \rightarrow (5, 2)$ .
  10. *secondRobot.printLastCoordinates()* prints *5 2* because its last movement was  $(5, 2) \rightarrow (5, 2)$ .

**Question - 3**

Which is the right answer to the following?

---

Which of the following packages is imported by default?