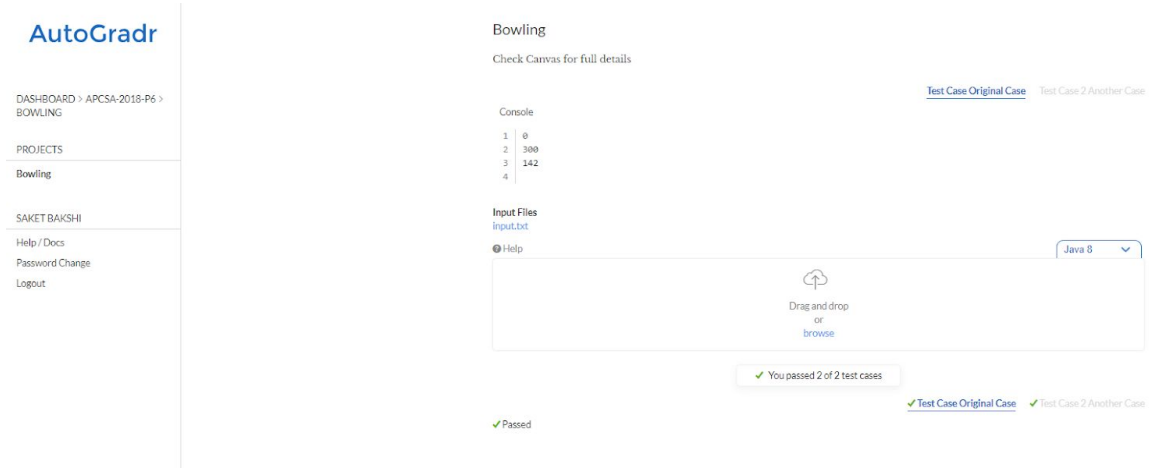


```
PS C:\Users\saket\Git\CSWork\JAVA\Labs\BowlingLabP6BakshiSaket> java Main
0
300
142
PS C:\Users\saket\Git\CSWork\JAVA\Labs\BowlingLabP6BakshiSaket>
```

```
/* Saket Bakshi. AP Computer Science A. Bowling Lab. Due February 8, 2019.
This class creates Frame objects that represent individual rounds of a bowling game.
*/
public class Frame
{
   private int roll1; //first roll value
   private int roll2; //second roll value
   private String type; //tells if Strike or Spare

   /** This class creates Frame objects that represent individual rounds of a bowling game. It has methods to set roll values and the type of frame.
   */
   public Frame()
   {
      roll1 = 0;
      roll2 = 0;
      type = "";
   }

   /** This class creates Frame objects that represent individual rounds of a bowling game. It has methods to set roll values and the type of frame.
   @param wholeFrame the String object that consists of both rolls, whether they are integers, spares, 0's, or strikes.
   */
   public Frame(String wholeFrame)
   {
```

```
        if(wholeFrame.length() == 1) //if the frame has a strike
        {
            roll1 = 10;
            roll2 = 0;
            type = "Strike";
        }
        else if(wholeFrame.substring(1,2).equals("/")) //if the frame has a spare
        {
            roll1 = Integer.parseInt(wholeFrame.substring(0,1)); //uses parseInt to find the first roll
value
            roll2 = 10 - roll1;
            type = "Spare";
        }
        else //otherwise it's a normal frame
        {
            roll1 = Integer.parseInt(wholeFrame.substring(0,1));
            roll2 = Integer.parseInt(wholeFrame.substring(1,2));
            type = "Normal";
        }
    }

    /** sets the integer value of the first roll
    */
    public void setFirstRoll(int roll) {roll1 = roll;}

    /** sets the integer value of the second roll
    */
    public void setSecondRoll(int roll) {roll2 = roll;}

    /** sets the String value of the type of roll. Strikes are "Strike", spares are "Spare", and normal
frames are "".
    */
    public void setType(String typeOfRoll) {type = typeOfRoll;}

    /** returns the first roll
    @return the first roll
    */
    public int getFirstRoll() {return roll1;}

    /** returns the second roll
    @return the second roll
    */
    public int getSecondRoll() {return roll2;}
```

```
   /** returns the type of roll
   @return the type
   */
   public String getType() {return type;}
}
/* Saket Bakshi. AP Computer Science A. Bowling Lab. Due February 8, 2019.
This class creates Frame objects that represent the last round of a bowling game.
*/
public class FinalFrame extends Frame
{
	private int roll3; //third roll value

	/** This class creates Frame objects that represent the last round of a bowling game. It has a method to return the last roll.
   */
	public FinalFrame()
	{
		super.setFirstRoll(0); //uses superclass to set first and second roll, and type
		super.setSecondRoll(0);
		super.setType("");
		roll3 = 0; //sets default third roll
	}

   /** This class creates Frame objects that represent individual rounds of a bowling game. It has a method to return the last roll.
   @param wholeFrame the String object that consists of both rolls, whether they are integers, spares, 0's, or strikes.
   */
	public FinalFrame(String wholeFrame)
	{
		if(wholeFrame.length() == 2) //if there is no spare or first roll strike
		{
			super.setFirstRoll(Integer.parseInt(wholeFrame.substring(0,1))); //first and second roll are set normally, third roll is 0
			super.setSecondRoll(Integer.parseInt(wholeFrame.substring(1,2)));
			super.setType("");
			roll3 = 0;
		}
		else if(wholeFrame.length() == 3) //if first roll is strike or second roll is a spare
		{
			if(!wholeFrame.substring(0,1).equals("X")) //if the first roll isn't a strike
				super.setFirstRoll(Integer.parseInt(wholeFrame.substring(0,1)));
```

```
                        if(wholeFrame.substring(0,1).equals("X")) //if it is a strike
                        {
                                super.setFirstRoll(10);
                                super.setType("Strike");
                                if(wholeFrame.substring(1,2).equals("X")) //if second roll is a strike
                                        super.setSecondRoll(10);
                                else

super.setSecondRoll(Integer.parseInt(wholeFrame.substring(1,2)));

                                if(wholeFrame.substring(2,3).equals("X")) //if last roll is a strike
                                        roll3 = 10;
                                else if(wholeFrame.substring(2,3).equals("/")) //if last roll is a spare
                                        roll3 = 10 - super.getSecondRoll();
                                else //if last roll is normal
                                        roll3 = Integer.parseInt(wholeFrame.substring(2,3));
                        }
                        else if(wholeFrame.substring(1,2).equals("/")) //if second roll is a spare
                        {
                                super.setSecondRoll(10 - super.getFirstRoll());
                                super.setType("Spare");
                                if(wholeFrame.substring(2,3).equals("X")) //if last roll is a strike
                                        roll3 = 10;
                                else
                                        roll3 = Integer.parseInt(wholeFrame.substring(2,3));
                        }
                }
        }

   /** returns the third roll
   @return the third roll
   */
        public int getThirdRoll() {return roll3;}
}
/* Saket Bakshi. AP Computer Science A. Bowling Lab. Due February 8, 2019.
This class creates organizes Frame and FinalFrame objects to create a game.
*/
public class BowlingLab extends FinalFrame
{
        private int totalPoints; //total points of the game
        private Frame[] normalFrames; //first 9 frames
        private FinalFrame lastFrame; //last frame
```

```
	/** This class creates BowlingLab objects that represent a whole bowling game. It has methods to calculate and return the total points.
   @param frames the Frame array that consists of the first nine rolls of the game.
   @param ultimateFrame the last frame of the game
   */
	public BowlingLab(Frame[] frames, FinalFrame ultimateFrame)
	{
		totalPoints = 0;
		normalFrames = frames;
		lastFrame = ultimateFrame;
	}

	/** returns the total points
   @return the total points
   */
	public int getTotal() {return totalPoints;}

	/** This calculates the total points of all the frames
	*/
	public void calculatePoints()
	{
		for(int i = 0; i < 7; i++) //for frames 1 to 7
		{
			if(normalFrames[i].getType().equals("Normal")) //if a normal frame, this simply adds pins fallen to the total score
				totalPoints += normalFrames[i].getFirstRoll() + normalFrames[i].getSecondRoll();
			else if(normalFrames[i].getType().equals("Spare")) //if a spare, this checks the next roll and appropriately adds points
				totalPoints += 10 + normalFrames[i+1].getFirstRoll();
			else if(normalFrames[i].getType().equals("Strike")) //if a strike, this checks if the next roll is a strike and appropriately adds points
			{
				if(!normalFrames[i+1].getType().equals("Strike"))
					totalPoints += 10 + normalFrames[i+1].getFirstRoll() + normalFrames[i+1].getSecondRoll();
				else
					totalPoints += 10 + 10 + normalFrames[i+2].getFirstRoll();
			}
		}

		//for frame 8
```

```
		if(normalFrames[7].getType().equals("Normal")) //if a normal frame, this simply adds pins fallen to the total score
			totalPoints += normalFrames[7].getFirstRoll() + normalFrames[7].getSecondRoll();
		else if(normalFrames[7].getType().equals("Spare")) //if a spare, this checks the next roll and appropriately adds points
				totalPoints += 10 + normalFrames[8].getFirstRoll();
		else if(normalFrames[7].getType().equals("Strike")) //if a strike, checks if frame 9 is a strike and then frame 10, if needed
		{
			if(!normalFrames[8].getType().equals("Strike"))
					totalPoints += 10 + normalFrames[8].getFirstRoll() + normalFrames[8].getSecondRoll();
				else
					totalPoints += 10 + 10 + lastFrame.getFirstRoll();
		}

		//for frame 9
		if(normalFrames[8].getType().equals("Normal")) //if a normal frame, this simply adds pins fallen to the total score
			totalPoints += normalFrames[8].getFirstRoll() + normalFrames[8].getSecondRoll();
		else if(normalFrames[8].getType().equals("Spare")) //if a spare, this checks the next roll and appropriately adds points
				totalPoints += 10 + lastFrame.getFirstRoll();
		else if(normalFrames[8].getType().equals("Strike")) //if a strike, checks first two rolls of frame 10
			totalPoints += 10 + lastFrame.getFirstRoll() + lastFrame.getSecondRoll();

		//for frame 10
		if(lastFrame.getType().equals("")) //if a normal frame, this simply adds pins fallen to the total score
			totalPoints += lastFrame.getFirstRoll() + lastFrame.getSecondRoll();
		else
		{
			if(lastFrame.getType().equals("Strike")) //if first roll is a strike, this appropriately adds values of all 3 rolls
				totalPoints += 10 + lastFrame.getSecondRoll() + lastFrame.getThirdRoll();
			else if(lastFrame.getType().equals("Spare")) //if second roll is a spare, this appropriately adds values of the third roll as well
				totalPoints += 10 + lastFrame.getThirdRoll();
		}
```

```
	}
}
/* Saket Bakshi. AP Computer Science A. Bowling Lab. Due February 8, 2019.
This class tests BowlingLab class objects for multiple bowling games.
*/
import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;

public class BowlingLabTester
{
	public static void main(String[] args) throws FileNotFoundException
	{
		File inFile = new File("input.txt"); //inputs file
		Scanner scanned = new Scanner(inFile); //inputs scanner object

		int testCases = scanned.nextInt(); //finds number of test cases
		scanned.nextLine();

		for(int g = 0; g < testCases; g++)
		{
			String line = scanned.nextLine().replaceAll("-", "0"); //replaces 0's in game
with actual integer 0's

			String[] frames = line.split(","); //separates frame of each game
			Frame[] frameGame = new Frame[9]; //makes array of first 9 frames
			for(int i = 0; i < 9; i++)
			{
				frameGame[i] = new Frame(frames[i]); //initializes each object of
the frame
			}
			FinalFrame lastFrame = new FinalFrame(frames[9]); //defines the last
frame of the game

			BowlingLab playedGame = new BowlingLab(frameGame, lastFrame);
//makes a BowlingLab game object
			playedGame.calculatePoints(); //calculates points
			System.out.println(playedGame.getTotal()); //prints points to console
		}
		scanned.close();
	}
}
```