For questions 1 - 5, fill in the code in the class given.

```
public class Boy extends Actor {

      public void act() {
            move();
            checkForEdges();
            checkForCollisions();
      }

      public void move() {
            setLocation(getX() + 5, getY() - 3);
      }

      public void checkForEdges() {
            if( getX() == 0) {
            }
            else if (getX() > getWorld().getWidth()) {
            }

            if(getY() == 0) {
            }
            else if (getY() > getWorld().getHeight()) {
            }
      }

      public void checkForCollisions() {
            if(getOneIntersectingObject(Enemy.class) != null) {
                  World w = getWorld();
                  java.util.List<Girl> girls = w.getObjects(Girl.class);
                  w.removeObjects(girls);
                  Greenfoot.stop();
            }
      }
}
```

1. Fill in the move method so that the Boy moves in some direction. Think also about how the answer would change if you were required to move the Boy in a certain way. For example, move strictly horizontally across the screen, or strictly vertically on the screen.
2. Write the series of if-statements needed to check if the Boy is at the edges of the world.

3. Write the Java code required to check if the Boy has collided with an Enemy. If so, the scenario should stop.

4. Add to the code that before the scenario stops, all of the objects of type Girl should be removed from the world.

5. Instead of the code you wrote for questions 3 & 4, re-write the code for checkForCollisions to check for collisions with a Flower object. If the Boy collides with a flower, he should pick it up (remove it from the world). If the Boy has collected more than 2 flowers, another flower should be added to the world for every new flower he picks up. Otherwise, an Enemy should be inserted into the World at a random location.

```
public class Boy extends Actor {

      private int flowerCount;

      public Boy() {
            flowerCount = 0;
```

```
	}

	public void act() {
		move();
		checkForEdges();
		checkForCollisions();
	}

	public void move() {
		setLocation(getX() + 5, getY() - 3);
	}

	public void checkForEdges() {
		if( getX() == 0) {
		}
		else if (getX() > getWorld().getWidth()) {
		}

		if(getY() == 0) {
		}
		else if (getY() > getWorld().getHeight()) {
		}
	}

	public void checkForCollisions() {
		Actor flower = getOneIntersectingObject(Flower.class);
		if(flower != null) {
			flowerCount = flowerCount + 1;
			getWorld().removeObject(flower);

			if(flowerCount > 2) {
				getWorld().addObject(new Flower(), 50, 50);
			}
			else {
				getWorld().addObject(new Enemy(),
					Greenfoot.getRandomNumber(getWorld().getWidth()),
					Greenfoot.getRandomNumber(getWorld().getHeight());
			}
		}
	}
}
```

6. Below is a class for an Actor subclass named Dancer. You should program the Dancer class to keep track of how many steps it has danced. When it has danced 45 steps, the Dancer object should reset its count back to zero and set the Dancer object’s rotation to be 45 more to the right than it was previously.

```
public class Dancer extends Actor {
	private int steps;

	public Dancer() {
		steps = 0;
	}

	public void act() {
		steps = steps + 1;
		if(steps == 45) {
			steps = 0;
			setRotation(getRotation() + 45);
		}
	}
}
```

7. Write the code that goes through an image and tells if any of the pixels have a red value of 45.

```
public boolean isThereA45Red() {
	GreenfootImage image = getWorld().getBackground();

	for( int y = 0; y < image.getHeight(); y++ ) {
		for(int x = 0; x < image.getWidth(); x++) {
			if(image.getColorAt(x,y).getRed() == 45) {
				return true;
			}
		}
	}
	return false;
}
```