public class WhatsTheOutput {
    private static final short MASTS = 4;

    public static void main(String[] args) {
        final int somePeopleNeverChange;
        float aBoat, aShip;
        double upInPain = 88.8;
        long devilInDisguise;

        aBoat = (float)14.8;
        devilInDisguise = (long) (upInPain - MASTS / 2 * 11.1);
        aShip = (int)aBoat % MASTS + MASTS / 8 + 2;
        upInPain = upInPain / aShip;
        somePeopleNeverChange = (short)upInPain;
        aBoat *= -MASTS;

        System.out.println(aBoat);
        System.out.println(++devilInDisguise);
        System.out.println(aShip);
        System.out.println(upInPain);
        System.out.println((MASTS + somePeopleNeverChange));
    }
}
public class WhatsTheMethodOutput {

private static final short CENTURY = 2000;

public static void main(String[] args) {
    int thisYear = 2010;
    int nextYear;
    int thisMonth;

    nextYear = ++thisYear + CENTURY % 333;
    thisMonth = leap(nextYear, 28);
    System.out.println(thisYear + " " + nextYear + " " + thisMonth);
}

private static int leap(int february, double twentyNine) {
    double thisYear;

    thisYear = february + twentyNine / 8;
    february = 2;
    System.out.println((thisYear + twentyNine));
    return ((int) thisYear + february + (short) twentyNine);
}
}
public class WhatsTheSelectionOutput {

    private static final short PLANTATION = 9;

    public static void main(String[] args) {
        int poppySeed = 27;
        int mountainPass;

        poppySeed %= 14;

        if ((poppySeed - PLANTATION) > PLANTATION ||
            PLANTATION - poppySeed > 0) {
            mountainPass = selectYourRuler(poppySeed % 3);
            System.out.print("Price per KG is " + poppySeed);
        } else {
            mountainPass = selectYourRuler(--poppySeed % 5);
            mountainPass += selectYourRuler(poppySeed++ % 5);
            System.out.println("Price per KG is " + (poppySeed+mountainPass));
        }
    }

    private static int selectYourRuler(int inchesOfIndex) {
        switch (inchesOfIndex) {
            case 0:
            case 3:
                System.out.println("Workers unite");
                break;
            case 2:
                System.out.print("Workers rule");
            case 1:
                System.out.println(" but rulers don't work");
                break;
            case 5:
                System.out.println("Hammers and sickles");
                break;
            default:
                System.out.println("Free the people");
                break;
        }
        return(-inchesOfIndex);
    }

    public static void main(String[] args) {
    }

    }
public class WhatsTheLoopSelectionOutput {

    private static final short PLANTATION = 9;
    private static final short HIGH = 42;

    public static void main(String[] args) {
        int poppySeed = 27;
        int mountainPass;
        int steps;

        mountainPass = HIGH;
        System.out.println("The mountain pass starts at " + mountainPass);
        while (poppySeed < 4 * PLANTATION) {
            if (poppySeed < PLANTATION || poppySeed % 2 == 1) {
                mountainPass = poppySeed % 6;
                for (steps = 0; steps < mountainPass; steps++) {
                    poppySeed += steps;
                }
                poppySeed -= ----mountainPass;
                System.out.println("Low price per KG is " + poppySeed);
            } else {
                do {
                    poppySeed += mountainPass;
                    mountainPass = mountainPass * mountainPass;
                } while (mountainPass <= PLANTATION);
                System.out.println("High price per KG is " + poppySeed);
                poppySeed = HIGH;
            }
        }
        System.out.println("The mountain pass ends at " + mountainPass);
    }
}
public class WhatsTheArrayOutput {
  private static final int NUMBER_OF_GOBLINS = 2;
  public static void main(String[] args) {
    final int[] magic = {4, 3, 2, 1};
    int[] potionPot = new int[NUMBER_OF_GOBLINS];
    int blackCat;
    int ratsBlood;
    int witchBrew;

    for (blackCat = 0; blackCat < potionPot.length; blackCat++) {
      potionPot[blackCat] = 0;
      ratsBlood = blackCat;
      while (ratsBlood < magic.length) {
        potionPot[blackCat] += magic[ratsBlood];
        ratsBlood += (blackCat + 1);
      }
    }

    System.out.print("The magic ingredients are ");
    for (blackCat = 0; blackCat < potionPot.length; blackCat++) {
      System.out.print(potionPot[blackCat] + " ");
    }
    System.out.println();

    witchBrew = magic[potionPot[0] % magic.length];
    System.out.println("The antidote is " + witchBrew);
  }
}
public class WhatsTheObject {

    public static void main(String[] args) {
        MotherDuck myDucks;

        myDucks = new MotherDuck(4);
        myDucks.makeDucklings();
        myDucks.getTheDucksInARow();

        System.out.println("I wish I could draw a picture of the objects");
    }
}

public class MotherDuck {

    private Duckling[] babyDucks;

    public MotherDuck(int numberOfBabyDucks) {
        babyDucks = new Duckling[numberOfBabyDucks];
    }

    public void makeDucklings() {
        int index;

        for (index = 0; index < babyDucks.length; index++) {
            babyDucks[index] = new Duckling(index + 1);
        }
    }

    public void getTheDucksInARow() {
        int index;

        for (index = 0; index < babyDucks.length; index++) {
            babyDucks[index].setMother(this);
        }
    }
}

public class Duckling {

    private MotherDuck myMother;
    private int duckNumber;

    public Duckling(int duckNumber) {
        this.duckNumber = duckNumber;
        myMother = null;
    }

    public void setMother(MotherDuck mommy) {
        myMother = mommy;
    }
}
public class WhatsTheRecursionOutput {
  //---
  private static final int MARGE = 7;
  private static final int BART = 5;
  //---
  public static void main(String[] args) {
    System.out.println("Homer wants "+ eatMyShorts(MARGE) + " donuts");
  }
  //---
  private static int eatMyShorts(int donuts) {
    int doh;
    System.out.println("There are "+ donuts + " donuts");
    if (donuts % BART != 0) {
      doh = eatMyShorts(--donuts);
    } else {
      doh = donuts;
    }
    return(doh);
  }
  //---
}