/\*

\*Project: Candy Game with School Students

\*Author: Michael Jordan Sallmen

\*/

import java.util.\*;

public class CandyGame23 {

public static void main(String[] args) {

Random rand = new Random();

Scanner scan = new Scanner(System.in);

boolean isTheGameOver;

int classLow = 15, // lower limit which represents the smallest possbile class size

classHigh = 30, // upper limit which represents the maximum students per class

students, //getInteger(scan, classLow, classHigh);

lowPieces, // lowest possible # of pieces that can be distributed to start the game.

highPieces, // highest possible # of pieces that can be distributed to start the game.

print, // whether or not the player wants to print the results 1 = yes, 0 = no

totalMoves = 0;// number of rounds played

// get value for number of students in the class and make clear to user what they're typing in an int for

System.out.println("How many students are in the class?");

students = getInteger(scan, classLow, classHigh);

// create array for the class to store pieces of candy

int [] classList = new int[students];// will use this to store the current amount of candy each student has

int [] extraPieceCounter = new int[students];// number of times a student has to receive an extra piece of candy to get back to an even number

int [] piecesToPass = new int[students];// number of pieces each player will pass

// get the smallest number of pieces, must be between 4 and 10 inclusive and even

System.out.println("What is the smallest number of pieces of candy that will be distributed?");

lowPieces = getEvenInteger (scan, 4, 10);

//get the largest number of pieces, must be between [lowPieces + 2, highPieces]

System.out.println("\nWhat is the largest number of pieces of candy that will be distributed?");

highPieces = getEvenInteger(scan, lowPieces + 2, lowPieces +50);

// Pass out candy to begin the game

disbributeCandy(rand, classList, lowPieces, highPieces);

System.out.println("Here is the amount of candy each student received to begin the game.");

printArray(classList);

//System.out.println("\n");

// Find out if player wants the results to be printed after each round.

System.out.println("\nWe are about to begin the game.");

System.out.println("Would you like to print the results after each round?\n"

+ "Enter 0 for no or 1 for yes.");

print = getInteger(scan, 0, 1);

// Initially testing, to see if all pieces happen to be the same on the first deal.

isTheGameOver = gameIsOver(classList);

// continue to play while game isn't over (isTheGameOver = false)

while (isTheGameOver == false){

totalMoves++;

passTheCandy(classList, piecesToPass, extraPieceCounter);

isTheGameOver = gameIsOver(classList);

if (print == 1){

printArray(classList);

}

}

// print out results for player

System.out.println("\nGAME IS OVER");

System.out.println("\nfinal hand results...");

printArray(classList);

System.out.println("\nEXTRA CANDY:");

printArray(extraPieceCounter);

System.out.println("\nTOTAL MOVES: " + totalMoves + "\n");

}

// The following method will get any integer in a given range.

public static int getInteger(Scanner s, int lowLimit, int highLimit){

int q;

do{

System.out.println("Enter an integer between [" + lowLimit + "," + highLimit + "] inclusive!");

q = s.nextInt();

if (q < lowLimit || q > highLimit){

System.out.println("\nInvalid Integer. Try again.");

}

} while (q < lowLimit || q > highLimit);

System.out.println();

return q;

}

// The following method will get an EVEN integer within the range that the user will specify.

public static int getEvenInteger(Scanner s, int lowRange, int highRange){

int w;

do{

System.out.println("Enter an EVEN integer between [" + lowRange + "," + highRange + "] inclusive!");

w = s.nextInt();

if (w < lowRange || w > highRange){

System.out.println("\nInteger out of Range.");

}

if (w % 2 != 0){

System.out.println("You entered an odd integer.");

}

} while (w < lowRange || w > highRange || w % 2 != 0);

return w;

}

// The following method will hand out the candy to start the game.

public static void disbributeCandy(Random r, int [] studentList, int lowRange, int highRange){

int numberOfStudents = studentList.length;

int candy;

for (int e = 0; e < numberOfStudents; e ++){

do {

candy = r.nextInt(highRange+1);//+1 allows random to select the higher range number

} while (candy == 0 || candy % 2 != 0 || candy < lowRange || candy > highRange);

studentList[e] = candy;

}

}

// If every element in the array is the same, then the game will end.

public static boolean gameIsOver(int [] studentList){

boolean gameOver = true;// initially set to true ( all elements are equal)

int testNumber = studentList[0];// assign value of first element in the array to test

for (int i = 1; i < studentList.length; i++){

if (studentList[i] != testNumber){

gameOver = false;

break;

}

}

return gameOver;

}

//This Method actually plays the game once all the arrays are set up.

public static void passTheCandy(int[] studentList, int[] candyToPass, int[] extraPieces/\*, boolean gameOver\*/){

// step 1. split each pile in half and move half into candyToPass

for (int student = 0; student < studentList.length; student++){

studentList[student] = studentList[student]/2;

candyToPass[student] = studentList[student];// already been deivided by 2 so don't need to divide again

}

// step 2. add the CandyToPass value to the person to the left

for (int i = 0; i < studentList.length; i ++){

if (i == studentList.length - 1){// if last person in class it goes to the 1st person

studentList[0] += candyToPass[i];

}

else{

studentList[i+1] += candyToPass[i];

}

}

// step 3. check to see if value in studentList is odd and add 1 to students pile if it is AND add 1 to extraPieces

for (int x = 0; x < studentList.length; x ++){

if (studentList[x] % 2 != 0){

studentList[x] += 1;

extraPieces[x] += 1;

}

}

}

// prints the extrapieces array that counts how many times a student had an odd number

public static void printExtraPieces(int[] extraPieces){

printArray(extraPieces);

}

//print each element of array on same line with field width size of 4

public static void printArray(int [] intArray){

for(int i = 0; i < intArray.length; i ++){

System.out.printf("%-4d", intArray[i]);

}

System.out.println();

}

}

/\*Sample output:

run:

How many students are in the class?

Enter an integer between [15,30] inclusive!

16

What is the smallest number of pieces of candy that will be distributed?

Enter an EVEN integer between [4,10] inclusive!

5

You entered an odd integer.

Enter an EVEN integer between [4,10] inclusive!

3

Integer out of Range.

You entered an odd integer.

Enter an EVEN integer between [4,10] inclusive!

4

What is the largest number of pieces of candy that will be distributed?

Enter an EVEN integer between [6,54] inclusive!

33

You entered an odd integer.

Enter an EVEN integer between [6,54] inclusive!

44

Here is the amount of candy each student received to begin the game.

4 6 20 22 26 12 12 12 18 38 8 32 26 12 30 44

We are about to begin the game.

Would you like to print the results after each round?

Enter 0 for no or 1 for yes.

Enter an integer between [0,1] inclusive!

1

24 6 14 22 24 20 12 12 16 28 24 20 30 20 22 38

32 16 10 18 24 22 16 12 14 22 26 22 26 26 22 30

32 24 14 14 22 24 20 14 14 18 24 24 24 26 24 26

30 28 20 14 18 24 22 18 14 16 22 24 24 26 26 26

28 30 24 18 16 22 24 20 16 16 20 24 24 26 26 26

28 30 28 22 18 20 24 22 18 16 18 22 24 26 26 26

28 30 30 26 20 20 22 24 20 18 18 20 24 26 26 26

28 30 30 28 24 20 22 24 22 20 18 20 22 26 26 26

28 30 30 30 26 22 22 24 24 22 20 20 22 24 26 26

28 30 30 30 28 24 22 24 24 24 22 20 22 24 26 26

28 30 30 30 30 26 24 24 24 24 24 22 22 24 26 26

28 30 30 30 30 28 26 24 24 24 24 24 22 24 26 26

28 30 30 30 30 30 28 26 24 24 24 24 24 24 26 26

28 30 30 30 30 30 30 28 26 24 24 24 24 24 26 26

28 30 30 30 30 30 30 30 28 26 24 24 24 24 26 26

28 30 30 30 30 30 30 30 30 28 26 24 24 24 26 26

28 30 30 30 30 30 30 30 30 30 28 26 24 24 26 26

28 30 30 30 30 30 30 30 30 30 30 28 26 24 26 26

28 30 30 30 30 30 30 30 30 30 30 30 28 26 26 26

28 30 30 30 30 30 30 30 30 30 30 30 30 28 26 26

28 30 30 30 30 30 30 30 30 30 30 30 30 30 28 26

28 30 30 30 30 30 30 30 30 30 30 30 30 30 30 28

28 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30

30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30

GAME IS OVER

final hand results...

30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30

EXTRA CANDY:

21 22 5 5 5 7 9 9 6 8 10 8 10 14 15 4

TOTAL MOVES: 24

BUILD SUCCESSFUL (total time: 51 seconds)

\*/