public class Exercise1\_2 {

public static void main(String[] args) {

System.out.println("Welcome to Java");

System.out.println("Welcome to Java");

System.out.println("Welcome to Java");

System.out.println("Welcome to Java");

System.out.println("Welcome to Java");

}

}

public class Exercise1\_4 {

public static void main(String[] args) {

System.out.println("a a^2 a^3");

System.out.println("1 1 1");

System.out.println("2 4 8");

System.out.println("3 9 27");

System.out.println("4 16 64");

}

}

public class Exercise1\_6 {

public static void main(String[] args) {

System.out.println(1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9);

}

}

public class Exercise1\_8 {

public static void main(String[] args) {

// Display area

System.out.println(5.5 \* 5.5 \* 3.14159);

// Display perimeter

System.out.println(2 \* 5.5 \* 3.14159);

}

}

import javax.swing.JOptionPane;

public class Exercise2\_1WithDialogBox {

// Main method

public static void main(String[] args) {

// Enter a temperatur in Fahrenheit

String celsiusString = JOptionPane.showInputDialog(null,

"Enter a temperature in Celsius:",

"Exercise2\_1 Input", JOptionPane.QUESTION\_MESSAGE);

// Convert string to double

double celsius = Double.parseDouble(celsiusString);

// Convert it to Celsius

double fahrenheit = (9.0 / 5) \* celsius + 32;

// Display the result

JOptionPane.showMessageDialog(null, "The temperature is " +

fahrenheit + " in Fahrenheit");

}

}

import java.util.Scanner;

public class Exercise2\_2 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Enter radius of the cylinder

System.out.print("Enter radius of the cylinder: ");

double radius = input.nextDouble();

// Enter length of the cylinder

System.out.print("Enter length of the cylinder: ");

double length = input.nextDouble();

double volume = radius \* radius \* 3.14159 \* length;

System.out.println("The volume of the cylinder is " + volume);

}

}

public class Exercise2\_4 {

public static void main(String[] args) {

// Prompt the input

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("Enter a number in pounds: ");

double pounds = input.nextDouble();

double kilograms = pounds \* 0.454;

System.out.println(pounds + " pounds is " + kilograms + " kilograms");

}

}

// Exercise2\_6.java: Summarize all digits in an integer < 1000

public class Exercise2\_6 {

// Main method

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Read a number

System.out.print("Enter an integer between 0 and 1000: ");

int number = input.nextInt();

// Find all digits in number

int lastDigit = number % 10;

int remainingNumber = number / 10;

int secondLastDigit = remainingNumber % 10;

remainingNumber = remainingNumber / 10;

int thirdLastDigit = remainingNumber % 10;

// Obtain the sum of all digits

int sum = lastDigit + secondLastDigit + thirdLastDigit;

// Display results

System.out.println("The sum of all digits in " + number

+ " is " + sum);

}

}

public class Exercise2\_8 {

public static void main(String args[]) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Enter an ASCII code

System.out.print("Enter an ASCII code: ");

int code = input.nextInt();

// Display result

System.out.println("The character for ASCII code "

+ code + " is " + (char)code);

}

}

import java.util.Scanner;

public class Exercise2\_10 {

/\*\* Main method \*/

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Receive the amount entered from the keyboard

System.out.print("Enter an amount in double, for example 11.56 ");

double amount = input.nextDouble();

int remainingAmount = (int)(amount \* 100);

// Find the number of one dollars

int numberOfOneDollars = remainingAmount / 100;

remainingAmount = remainingAmount % 100;

// Find the number of quarters in the remaining amount

int numberOfQuarters = remainingAmount / 25;

remainingAmount = remainingAmount % 25;

// Find the number of dimes in the remaining amount

int numberOfDimes = remainingAmount / 10;

remainingAmount = remainingAmount % 10;

// Find the number of nickels in the remaining amount

int numberOfNickels = remainingAmount / 5;

remainingAmount = remainingAmount % 5;

// Find the number of pennies in the remaining amount

int numberOfPennies = remainingAmount;

// Display results

String output = "Your amount " + amount + " consists of \n" +

numberOfOneDollars + " dollars\n" +

numberOfQuarters + " quarters\n" +

numberOfDimes + " dimes\n" +

numberOfNickels + " nickels\n" +

numberOfPennies + " pennies";

System.out.println(output);

}

}

import javax.swing.JOptionPane;

public class Exercise2\_12a {

public static void main(String args[]) {

// Obtain input

String balanceString = JOptionPane.showInputDialog(null,

"Enter balance:");

double balance = Double.parseDouble(balanceString);

String interestRateString = JOptionPane.showInputDialog(null,

"Enter annual interest rate:");

double annualInterestRate = Double.parseDouble(interestRateString);

double monthlyInterestRate = annualInterestRate / 1200;

double interest = balance \* monthlyInterestRate;

// Display output

JOptionPane.showMessageDialog(null, "The interest is " +

(int)(100\* interest) / 100.0);

}

}

import java.util.Scanner;

public class Exercise2\_12b {

public static void main(String args[]) {

Scanner input = new Scanner(System.in);

// Obtain input

System.out.print("Enter balance: ");

double balance = input.nextDouble();

System.out.print("Enter annual interest rate: ");

double annualInterestRate = input.nextDouble();

double monthlyInterestRate = annualInterestRate / 1200;

double interest = balance \* monthlyInterestRate;

// Display output

System.out.println("The interest is " + (int)(100\* interest) / 100.0);

}

}

import java.util.Scanner;

public class Exercise2\_14 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user to enter weight in pounds

System.out.print("Enter weight in pounds: ");

double weight = input.nextDouble();

// Prompt the user to enter height in inches

System.out.print("Enter height in inches: ");

double height = input.nextDouble();

double bmi = weight \* 0.45359237 / (height \* 0.0254 \* height \* 0.0254);

System.out.print("BMI is " + bmi);

}

}

public class Exercise2\_16 {

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print(

"Enter the amount of water in kilograms: ");

double mass = input.nextDouble();

System.out.print("Enter the initial temperature: ");

double initialTemperature = input.nextDouble();

System.out.print(

"Enter the final temperature: ");

double finalTemperature = input.nextDouble();

double energy =

mass \* (finalTemperature - initialTemperature) \* 4184;

System.out.print("The energy needed is " + energy);

}

}

public class Exercise2\_18 {

// Main method

public static void main(String[] args) {

System.out.println("a b pow(a, b)");

System.out.println("1 2 " + (int)Math.pow(1, 2));

System.out.println("2 3 " + (int)Math.pow(2, 3));

System.out.println("3 4 " + (int)Math.pow(3, 4));

System.out.println("4 5 " + (int)Math.pow(4, 5));

System.out.println("5 6 " + (int)Math.pow(5, 6));

}

}

import java.util.Scanner;

public class Exercise2\_20 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Enter the first point with two double values

System.out.print("Enter x1 and y1: ");

double x1 = input.nextDouble();

double y1 = input.nextDouble();

// Enter the second point with two double values

System.out.print("Enter x2 and y2: ");

double x2 = input.nextDouble();

double y2 = input.nextDouble();

// Compute the distance

double distance = Math.pow((x1 - x2) \* (x1 - x2) +

(y1 - y2) \* (y1 - y2), 0.5);

System.out.println("The distance of the two points is " +

distance);

}

}

import java.util.Scanner;

public class Exercise2\_22 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Enter the side of the hexagon

System.out.print("Enter the side: ");

double side = input.nextDouble();

// Compute the area

double area = 3 \* 1.732 \* side \* side / 2;

System.out.println("The area of the hexagon is " + area);

}

}

import java.util.Scanner;

public class Exercise2\_24 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter speed v: ");

double v = input.nextDouble();

System.out.print("Enter acceleration a: ");

double a = input.nextDouble();

double length = v \* v / (2 \* a);

System.out.println("The minimum runway length for this airplane is " +

length + " meters");

}

}

public class Exercise3\_2 {

/\*\*Main method\*/

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Prompt the user to enter an integer

System.out.print("Enter an integer: ");

int number = input.nextInt();

// Display results

System.out.println("Is " + number + " an even number? " +

(number % 2 == 0));

}

}

import javax.swing.\*;

public class Exercise3\_4 {

public static void main(String[] args) {

int number1 = (int)(System.currentTimeMillis() % 100);

int number2 = (int)(System.currentTimeMillis() \* 7 % 100);

String resultString = JOptionPane.showInputDialog

("What is " + number1 + " + " + number2 + "?");

int result = Integer.parseInt(resultString);

JOptionPane.showMessageDialog(null,

number1 + " + " + number2 + " = " + result + " is " +

(number1 + number2 == result));

}

}

import javax.swing.\*;

public class Exercise3\_5WithJOptionPane {

public static void main(String[] args) {

int number1 = (int)(System.currentTimeMillis() % 10);

int number2 = (int)(System.currentTimeMillis() \* 7 % 10);

int number3 = (int)(System.currentTimeMillis() \* 3 % 10);

String answerString = JOptionPane.showInputDialog

("What is " + number1 + " + " + number2 + " + " +

number3 + "?");

int answer = Integer.parseInt(answerString);

JOptionPane.showMessageDialog(null,

number1 + " + " + number2 + " + " + number3 + " = " + answer + " is " +

(number1 + number2 + number3 == answer));

}

}

import java.util.Scanner;

public class Exercise3\_6 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user to enter weight in pounds

System.out.print("Enter weight in pounds: ");

double weight = input.nextDouble();

// Prompt the user to enter height

System.out.print("Enter feet: ");

double feet = input.nextDouble();

System.out.print("Enter inches: ");

double inches = input.nextDouble();

double height = feet \* 12 + inches;

// Compute BMI

double bmi = weight \* 0.45359237 /

((height \* 0.0254) \* (height \* 0.0254));

// Display result

System.out.println("Your BMI is " + bmi);

if (bmi < 16)

System.out.println("You are seriously underweight");

else if (bmi < 18)

System.out.println("You are underweight");

else if (bmi < 24)

System.out.println("You are normal weight");

else if (bmi < 29)

System.out.println("You are over weight");

else if (bmi < 35)

System.out.println("You are seriously over weight");

else

System.out.println("You are gravely over weight");

}

}

public class Exercise3\_8 {

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Enter three numbers

System.out.print("Enter three integers: ");

int num1 = input.nextInt();

int num2 = input.nextInt();

int num3 = input.nextInt();

if (num1 > num2) {

int temp = num1;

num1 = num2;

num2 = temp;

}

if (num2 > num3) {

int temp = num2;

num2 = num3;

num3 = temp;

}

if (num1 > num2) {

int temp = num1;

num1 = num2;

num2 = temp;

}

System.out.println("The sorted numbers are "

+ num1 + " " + num2 + " " + num3);

}

}

import javax.swing.JOptionPane;

public class Exercise3\_10 {

public static void main(String[] args) {

// 1. Generate two random single-digit integers

int number1 = (int)(Math.random() \* 10);

int number2 = (int)(Math.random() \* 10);

// 2. Prompt the student to answer 搘hat is number1 + number2?? String answerString = JOptionPane.showInputDialog

("What is " + number1 + " + " + number2 + "?");

int answer = Integer.parseInt(answerString);

// 4. Grade the annser and display the result

String replyString;

if (number1 + number2 == answer)

replyString = "You are correct!";

else

replyString = "Your answer is wrong.\n" + number1 + " + "

+ number2 + " should be " + (number1 + number2);

JOptionPane.showMessageDialog(null, replyString);

}

}

import java.util.Scanner;

public class Exercise3\_12 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user to enter an integer

System.out.print("Enter an integer: ");

int number = input.nextInt();

if (number % 5 == 0 && number % 6 == 0)

System.out.println(number + " is divisible by both 5 and 6");

else if (number % 5 == 0 ^ number % 6 == 0)

System.out.println(number + " is divisible by both 5 and 6, but not both");

else

System.out.println(number + " is not divisible by either 5 or 6");

}

}

public class Exercise3\_14 {

public static void main(String[] args) {

// Obtain the random number 0 or 1

int number = (int)(Math.random() \* 2);

// Prompt the user to enter a guess

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("Guess head or tail? " +

"Enter 0 for head and 1 for tail: ");

int guess = input.nextInt();

// Check the guess

if (guess == number)

System.out.println("Correct guess");

else if (number == 0)

System.out.println("Sorry, it is a head");

else

System.out.println("Sorry, it is a tail");

}

}

public class Exercise3\_16 {

public static void main(String[] args) {

System.out.println((char)('A' + Math.random() \* 27));

}

}

import java.util.Scanner;

public class Exercise3\_18 {

/\*\* Main method \*/

public static void main(String args[]) {

Scanner input = new Scanner(System.in);

// Prompt the user to enter a year

System.out.print("Enter a year: ");

// Convert the string into an int value

int year = input.nextInt();

// Check if the year is a leap year

boolean isLeapYear =

((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0);

// Display the result in a message dialog box

System.out.println(year + " is a leap year? " + isLeapYear);

}

}

public class Exercise3\_20 {

// Main method

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Enter the temperature in Fahrenheit

System.out.print("Enter the temperature in Fahrenheit: ");

double fahrenheit = input.nextDouble();

if (fahrenheit < -58 || fahrenheit > 41) {

System.out.println("Temperature must be between -58癋 and 41癋");

System.exit(0);

}

// Enter the wind speed miles per hour

System.out.print("Enter the wind speed miles per hour: ");

double speed = input.nextDouble();

if (speed < 2) {

System.out.println("Speed must be greater than or equal to 2");

System.exit(0);

}

// Compute wind chill index

double windChillIndex = 35.74 + 0.6215 \* fahrenheit - 35.75 \*

Math.pow(speed, 0.16) + 0.4275 \* fahrenheit \*

Math.pow(speed, 0.16);

// Display the result

System.out.println("The wind chill index is " + windChillIndex);

}

}

import java.util.Scanner;

public class Exercise3\_22 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Enter a point with two double values

System.out.print("Enter a point with two coordinates: ");

double x = input.nextDouble();

double y = input.nextDouble();

// Compute the distance

double distance = Math.pow(x \* x + y \* y, 0.5);

if (distance <= 10)

System.out.println("Point (" + x + ", " + y +

") is in the circle");

else

System.out.println("Point (" + x + ", " + y +

") is not in the circle");

}

}

public class Exercise3\_24 {

public static void main(String[] args) {

final int NUMBER\_OF\_CARDS = 52;

// Pick a card

int number = (int)(Math.random() \* NUMBER\_OF\_CARDS);

System.out.print("The card you picked is ");

if (number % 13 == 0)

System.out.print("Ace of ");

else if (number % 13 == 10)

System.out.print("Jack of ");

else if (number % 13 == 11)

System.out.print("Queen of ");

else if (number % 13 == 12)

System.out.print("King of ");

else

System.out.print((number % 13) + " of ");

if (number / 13 == 0)

System.out.println("Clubs");

else if (number / 13 == 1)

System.out.println("Diamonds");

else if (number / 13 == 2)

System.out.println("Hearts");

else if (number / 13 == 3)

System.out.println("Spades");

}

}

public class Exercise3\_26 {

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Enter an integer

System.out.print("Enter an integer: ");

int number = input.nextInt();

System.out.println("Is " + number + " divisible by 5 and 6? " +

((number % 5 == 0) && (number % 6 == 0)));

System.out.println("Is " + number + " divisible by 5 or 6? " +

((number % 5 == 0) || (number % 6 == 0)));

System.out.println("Is " + number +

" divisible by 5 or 6, but not both? " +

((number % 5 == 0) ^ (number % 6 == 0)));

}

}

import java.util.Scanner;

public class Exercise3\_28 {

public static void main(String args[]) {

Scanner input = new Scanner(System.in);

System.out.print("Enter r1抯 center x-, y-coordinates, width, and height: ");

double x1 = input.nextDouble();

double y1 = input.nextDouble();

double w1 = input.nextDouble();

double h1 = input.nextDouble();

System.out.print("Enter r2抯 center x-, y-coordinates, width, and height: ");

double x2 = input.nextDouble();

double y2 = input.nextDouble();

double w2 = input.nextDouble();

double h2 = input.nextDouble();

double xDistance = x1 - x2 >= 0 ? x1 - x2 : x2 - x1;

double yDistance = y1 - y2 >= 0 ? y1 - y2 : y2 - y1;

if (xDistance <= (w1 - w2) / 2 && yDistance <= (h1 - h2) / 2)

System.out.println("r2 is inside r1");

else if (xDistance <= (w1 + w2) / 2 && yDistance <= (h1 + h2) / 2)

System.out.println("r2 overlaps r1");

else

System.out.println("r2 does not overlap r1");

}

}

import java.util.Scanner;

public class Exercise3\_30 {

public static void main(String[] args) {

// Prompt the user to enter the time zone offset to GMT

Scanner input = new Scanner(System.in);

System.out.print("Enter the time zone offset to GMT: ");

long timeZoneOffset = input.nextInt();

// Obtain the total milliseconds since the midnight, Jan 1, 1970

long totalMilliseconds = System.currentTimeMillis();

// Obtain the total seconds since the midnight, Jan 1, 1970

long totalSeconds = totalMilliseconds / 1000;

// Compute the current second in the minute in the hour

long currentSecond = totalSeconds % 60;

// Obtain the total minutes

long totalMinutes = totalSeconds / 60;

// Compute the current minute in the hour

long currentMinute = totalMinutes % 60;

// Obtain the total hours

long totalHours = totalMinutes / 60;

// Compute the current hour

long currentHour = (totalHours + timeZoneOffset) % 24;

// Display results

System.out.print("Current time is " + (currentHour % 12) + ":"

+ currentMinute + ":" + currentSecond);

if (currentHour < 12)

System.out.println(" AM");

else

System.out.println(" PM");

}

}

public class Exercise4\_2 {

public static void main(String[] args) {

int correctCount = 0; // Count the number of correct answers

int count = 0; // Count the number of questions

java.util.Scanner input = new java.util.Scanner(System.in);

long startTime = System.currentTimeMillis();

while (count < 10) {

// 1. Generate two random single-digit integers

int number1 = 1 + (int)(Math.random() \* 15);

int number2 = 1 + (int)(Math.random() \* 15);

// 2. Prompt the student to answer 搘hat is number1 ?number2?? System.out.print("What is " + number1 + " + " + number2 + "? ");

int answer = input.nextInt();

// 3. Grade the answer and display the result

String replyString;

if (number1 + number2 == answer) {

replyString = "You are correct!";

correctCount++;

}

else {

replyString = "Your answer is wrong.\n" + number1 + " + "

+ number2 + " should be " + (number1 + number2);

}

System.out.println(replyString);

// Increase the count

count++;

}

System.out.println("Correct count is " + correctCount);

long endTime = System.currentTimeMillis();

System.out.println("Time spent is " + (endTime - startTime) / 1000 + " seconds");

}

}

public class Exercise4\_4 {

public static void main(String[] args) {

System.out.println("Miles\t\tKilometers");

System.out.println("-------------------------------");

// Use while loop

int miles = 1;

while (miles <= 10) {

System.out.println(miles + "\t\t" + miles \* 1.609);

miles++;

}

/\*\* Alternatively use for loop

for (int miles = 1; miles <= 10; miles++) {

System.out.println(miles + "\t\t" + miles \* 1.609);

}

\*/

}

}

public class Exercise4\_6 {

public static void main(String[] args) {

System.out.printf("%10s%10s | %10s%10s\n", "Miles", "Kilometers", "Kilometers", "Miles");

System.out.println("---------------------------------------------");

// Use while loop

int miles = 1; int kilometers = 20; int count = 1;

while (count <= 10) {

System.out.printf("%10d%10.3f | %10d%10.3f\n", miles, miles \* 1.609, kilometers, kilometers / 1.609);

miles++; kilometers += 5; count++;

}

/\* Use for loop

int miles = 1; int kilometers = 20;

for (int count = 1; count <= 10; miles++, kilometers += 5, count++) {

System.out.printf("%10d%10.3f | %10d%10.3f\n", miles, miles \* 1.609, kilometers, kilometers / 1.609);

}

\*/

}

}

import java.util.\*;

public class Exercise4\_8 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user to enter the number of students

System.out.print(

"Enter the number of students: ");

int numOfStudents = input.nextInt();

System.out.print(

"Enter a student name: ");

String student1 = input.next();

System.out.print(

"Enter a student score: ");

double score1 = input.nextDouble();

for (int i = 0; i < numOfStudents - 1; i++) {

System.out.print(

"Enter a student name: ");

String student = input.next();

System.out.print(

"Enter a student score: ");

double score = input.nextDouble();

if (score > score1) {

student1 = student;

score1 = score;

}

}

System.out.println("Top student " +

student1 + "'s score is " + score1);

}

}

public class Exercise4\_10 {

public static void main(String[] args) {

int count = 1;

for (int i = 100; i <= 1000; i++)

if (i % 5 == 0 && i % 6 == 0)

System.out.print((count++ % 10 != 0) ? i + " ": i + "\n");

}

}

/\*\* Find the smallest number such that n\*n < 12000 \*/

public class Exercise4\_12 {

// Main method

public static void main(String[] args) {

int i = 1;

while (i \* i <= 12000 ) {

i++;

}

System.out.println("This number is " + i);

}

}

public class Exercise4\_14 {

public static void main(String[] args) {

int count = 1;

for (int i = '!'; i < '~'; i++) {

System.out.print((count++ % 10 != 0) ? (char)i + " " :

(char)i + "\n");

}

}

}

public class Exercise4\_16 {

// Main method

public static void main(String args[]) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Prompt the user to enter a positive integer

System.out.print(

"Enter a positive integer: ");

int number = input.nextInt();

// Find all the smallest factors of the integer

System.out.println("The factors for " + number + " is");

int factor = 2;

while (factor <= number) {

if (number % factor == 0) {

number = number / factor;

System.out.println(factor);

}

else {

factor++;

}

}

}

}

public class Exercise4\_20 {

// Main method

public static void main(String[] args) {

int count = 1; // Count the number of prime numbers

int number = 2; // A number to be tested for primeness

boolean isPrime = true; // If the current number is prime?

System.out.println("The prime numbers from 2 to 1000 are \n");

// Repeatedly test if a new number is prime

while (number <= 1000) {

// Assume the number is prime

isPrime = true;

// Set isPrime to false, if the number is prime

for (int divisor = 2; divisor <= number / 2; divisor++) {

if (number % divisor == 0) { // If true, the number is prime

isPrime = false;

break; // Exit the for loop

}

}

// Print the prime number and increase the count

if (isPrime) {

if (count%8 == 0) {

// Print the number and advance to the new line

System.out.println(number);

}

else

System.out.print(number + " ");

count++; // Increase the count

}

// Check if the next number is prime

number++;

}

}

}

import javax.swing.JOptionPane;

public class Exercise4\_22 {

public static void main(String[] args) {

int numOfYears;

double loanAmount;

java.util.Scanner input = new java.util.Scanner(System.in);

// Enter loan amount

System.out.print(

"Enter loan amount, for example 120000.95: ");

loanAmount = input.nextDouble();

// Enter number of years

System.out.print(

"Enter number of years as an integer, \nfor example 5: ");

numOfYears = input.nextInt();

// Enter yearly interest rate

System.out.print(

"Enter yearly interest rate, for example 8.25: ");

// Convert string to double

double annualInterestRate =

input.nextDouble();

// Obtain monthly interest rate

double monthlyInterestRate = annualInterestRate/1200;

// Compute mortgage

double monthlyPayment = loanAmount\*monthlyInterestRate /

(1 - (Math.pow(1 / (1 + monthlyInterestRate), numOfYears \* 12)));

double balance = loanAmount;

double interest;

double principal;

System.out.println("Loan Amount: " + loanAmount);

System.out.println("Number of Years: " + numOfYears);

System.out.println("Interest Rate: " + annualInterestRate + "%");

System.out.println();

System.out.println("Monthly Payment: " + (int)(monthlyPayment \* 100) / 100.0 );

System.out.println("Total Payment: " + (int)(monthlyPayment \* 12 \* numOfYears \* 100) / 100.0 + "\n" );

// Display the header

System.out.println("Payment#\tInterest\tPrincipal\tBalance");

int i;

for (i = 1; i <= numOfYears \* 12; i++) {

interest = (int)(monthlyInterestRate \* balance \* 100) / 100.0;

principal = (int)((monthlyPayment - interest) \* 100) / 100.0;

balance = (int)((balance - principal) \* 100) / 100.0;

System.out.println(i + "\t\t" + interest

+ "\t\t" + principal + "\t\t" + balance);

}

}

}

public class Exercise4\_24 {

public static void main(String[] args) {

double sum = 0;

for (int i = 1; i <= 97; i += 2)

sum += 1.0 \* i / (i + 2);

System.out.println("sum is " + sum);

}

}

public class Exercise4\_26 {

public static void main(String[] args) {

double e = 1;

double item = 1;

for (int i = 1; i <= 100000; i++) {

item = item / i;

e += item;

if (i == 10000 || i == 20000 || i == 30000 || i == 40000 ||

i == 50000 || i == 60000 || i == 70000 || i == 80000 ||

i == 90000 || i == 100000)

System.out.println("The e is " + e + " for i = " + i);

}

}

}

public class Exercise4\_28 {

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Prompt the user to enter input

System.out.print("Enter a year: ");

int year = input.nextInt();

System.out.print("Enter the first day of the year: ");

int firstDay = input.nextInt();

int numberOfDaysInMonth = 0;

// Display calendar for each month

for (int month = 1; month <= 12; month++) {

// Display Calendar title

switch (month) {

case 1: System.out.print("January 1, " + year + " is ");

numberOfDaysInMonth = 31;

break;

case 2: System.out.print("February 1, " + year + " is ");

if (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0))

numberOfDaysInMonth = 29;

else

numberOfDaysInMonth = 28;

break;

case 3: System.out.print("March 1, " + year + " is ");

numberOfDaysInMonth = 31;

break;

case 4: System.out.print("April 1, " + year + " is ");

numberOfDaysInMonth = 30;

break;

case 5: System.out.print("May 1, " + year + " is ");

numberOfDaysInMonth = 31;

break;

case 6: System.out.print("June 1, " + year + " is ");

numberOfDaysInMonth = 30;

break;

case 7: System.out.print("July 1, " + year + " is ");

numberOfDaysInMonth = 31;

break;

case 8: System.out.print("August 1, " + year + " is ");

numberOfDaysInMonth = 31;

break;

case 9: System.out.print("September 1, " + year + " is ");

numberOfDaysInMonth = 30;

break;

case 10: System.out.print("October 1, " + year + " is ");

numberOfDaysInMonth = 31;

break;

case 11: System.out.print("November 1, " + year + " is ");

numberOfDaysInMonth = 30;

break;

case 12: System.out.print("December 1, " + year + " is ");

numberOfDaysInMonth = 31;

break;

}

switch (firstDay) {

case 0: System.out.println("Sunday"); break;

case 1: System.out.println("Monday"); break;

case 2: System.out.println("Tuesday"); break;

case 3: System.out.println("Wednesday"); break;

case 4: System.out.println("Thursday"); break;

case 5: System.out.println("Friday"); break;

case 6: System.out.println("Saturday"); break;

}

// Get the start day for the next month

firstDay = (firstDay + numberOfDaysInMonth) % 7;

}

}

}

import java.util.Scanner;

public class Exercise4\_30 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter the amount to be saved for each month: ");

double monthlyDeposit = input.nextDouble();

System.out.print("Enter the annual interest rate: ");

double annualInterestRate = input.nextDouble();

double monthlyInterestRate = annualInterestRate / 1200;

System.out.print("Enter the number of months: ");

int numberOfMonths = input.nextInt();

double currentValue = monthlyDeposit \* (1 + monthlyInterestRate);

for (int i = 1; i < numberOfMonths; i++) {

currentValue = (currentValue + monthlyDeposit) \* (1 + monthlyInterestRate);

}

System.out.println("After the " + numberOfMonths +

"th month, the account value is " + currentValue);

}

}

public class Exercise4\_32 {

public static void main(String[] args) {

// Generate the first digit

int first = (int)(Math.random() \* 10);

// Generate the second digit

int second = (int)(Math.random() \* 10);;

while (first == second) {

second = (int)(Math.random() \* 10);

}

// Prompt the user to enter a guess

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("Enter your two digits lottery pick: ");

int guess = input.nextInt();

// Check the guess

if (guess / 10 == first && guess % 10 == second) {

System.out.println("Exact match: you win $10,000");

}

else if (guess % 10 == first && guess / 10 == second) {

System.out.println("Match all digits: you win $3,000");

}

else if (guess % 10 == first || guess % 10 == second

|| guess / 10 == first || guess / 10 == second) {

System.out.println("Match one digit: you win $1,000");

}

else {

System.out.println("Sorry, no match");

}

System.out.println("Lottery is " + first + second);

}

}

public class Exercise4\_34 {

public static void main(String[] args) {

int count = 0;

while (count <= 2 || count <= -2) {

// Generate scissor, rock, paper

int computerNumber = (int)(Math.random() \* 3);

// Prompt the user to enter scissor, rock, or paper

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("scissor (0), rock (1), paper (2): ");

int userNumber = input.nextInt();

// Check the guess

switch (computerNumber) {

case 0:

if (userNumber == 0) {

System.out.println("It is a draw");

}

else if (userNumber == 1) {

System.out.println("You won");

count++;

}

else if (userNumber == 2) {

System.out.println("You lost");

count--;

}

break;

case 1:

if (userNumber == 0) {

System.out.println("You lost");

count--;

}

else if (userNumber == 1) {

System.out.println("It is a draw");

}

else if (userNumber == 2) {

System.out.println("You won");

count++;

}

break;

case 2:

if (userNumber == 0) {

System.out.println("You won");

count++;

}

else if (userNumber == 1) {

System.out.println("You lost");

count--;

}

else if (userNumber == 2) {

System.out.println("It is a draw");

}

break;

}

}

if (count > 2) {

System.out.println("You won more than two times");

}

else {

System.out.println("The computer won more than two times");

}

}

}

import java.util.Scanner;

public class Exercise4\_36 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Prompt the user to enter an integer

System.out.print("Enter the first 9-digit of an ISBN number as integer: ");

int number = input.nextInt();

// Calculate checksum

int t = number;

int i = 9;

int sum = 0;

while (t != 0) {

sum += (t % 10) \* i;

i--;

t = t / 10;

}

int checksum = sum % 11;

System.out.print("The ISBN number is ");

// Display leading zeros

int temp = 100000000;

while (number / temp == 0) {

System.out.print("0");

temp = temp / 10;

}

System.out.print(number);

if (checksum == 10)

System.out.print("X");

else

System.out.print(checksum);

}

}

/\*\* Java has the Integer.toHexString(int) for converting a

\* decimal value to a binary as a string.

\*/

import java.util.Scanner;

public class Exercise4\_38 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter an integer: ");

int decimal = input.nextInt();

String hexString = "";

int value = decimal;

while (value != 0) {

int single = value % 16;

if (single == 15)

hexString = "F" + hexString;

else if (single == 14)

hexString = "E" + hexString;

else if (single == 13)

hexString = "D" + hexString;

else if (single == 12)

hexString = "C" + hexString;

else if (single == 11)

hexString = "B" + hexString;

else if (single == 10)

hexString = "A" + hexString;

else

hexString = single + hexString;

value = value / 16;

}

System.out.println(decimal + "'s hex representation is " +

hexString);

}

}

public class Exercise4\_40 {

public static void main(String[] args) {

int headCount = 0;

int tailCount = 0;

for (int i = 0; i < 100000; i++) {

int number = (int)(Math.random() \* 100000) % 2;

if (number == 0)

headCount++;

else

tailCount++;

}

System.out.println("head count: " + headCount);

System.out.println("tail count: " + tailCount);

}

}

public class Exercise4\_42 {

public static void main(String[] args) {

double commission = 0;

double salesAmount = 1;

java.util.Scanner input = new java.util.Scanner(System.in);

// Enter the desired commission

System.out.print(

"Enter the desired commission: ");

// Convert string to double

double commissionSought = input.nextDouble();

for (salesAmount = 1; commission < commissionSought;

salesAmount++) {

// Compute commission

if (salesAmount >= 10001)

commission = 5000 \* 0.08 + 5000 \* 0.1 + (salesAmount - 10000) \* 0.12;

else if (salesAmount >= 5001)

commission = 5000 \* 0.08 + (salesAmount - 5000) \* 0.10;

else

commission = salesAmount \* 0.08;

}

// Display the sales amount

System.out.println("The sales amount " + salesAmount +

" is needed to make a commission of $" + commissionSought);

}

}

public class Exercise4\_44 {

public static void main(String[] args) {

final int NUMBER\_OF\_TRIALS = 10000000;

int numberOfHits = 0;

for (int i = 0; i < NUMBER\_OF\_TRIALS; i++)

{

double x = Math.random() \* 2.0 - 1;

double y = Math.random() \* 2.0 - 1;

if (x < 0 || isInRegion3(x, y))

numberOfHits++;

}

System.out.println("The probability in Region 1 and 3 is " +

(1.0 \* numberOfHits / NUMBER\_OF\_TRIALS));

}

static boolean isInRegion3(double x, double y) {

if (x > 1 || x < 0 || y > 1 || y < 0)

return false;

else {

double slope = (1.0 - 0) / (0 - 1.0);

double x1 = x + -y \* slope;

if (x1 <= 1)

return true;

else

return false;

}

}

}

import java.util.Scanner;

public class Exercise4\_46 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter an integer: ");

int value = input.nextInt();

System.out.print("The 16 bits are ");

int mask = 1;

for (int i = 15; i >= 0; i--) {

int temp = value >> i;

int bit = temp & mask;

System.out.print(bit);

}

}

}

// Exercise5\_2.java: Create a method for summarizing digits in an int

public class Exercise5\_2 {

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("Enter a number: ");

int value = input.nextInt();

System.out.println("The sum of digits for " + value +

" is " + sumDigits(value));

}

public static int sumDigits(long n) {

int temp = (int)Math.abs(n);

int sum = 0;

while (temp != 0) {

int remainder = temp % 10;

sum += remainder;

temp = temp / 10;

}

return sum;

}

}

public class Exercise5\_4 {

public static void main(String[] args) {

System.out.print("Enter an integer: ");

java.util.Scanner input = new java.util.Scanner(System.in);

int number = input.nextInt();

reverse(number);

}

public static void reverse(int number) {

while (number != 0) {

int remainder = number % 10;

System.out.print(remainder);

number = number / 10;

}

System.out.println();

}

}

public class Exercise5\_6 {

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("Enter line number: ");

int lineNumber = input.nextInt();

displayPattern(lineNumber);

}

public static void displayPattern(int n) {

for (int row = 1; row <= n; row++) {

// Print spaces

for (int i = row; i < n; i++)

System.out.print(" ");

// Print numbers

for (int i = row; i >= 1; i--)

System.out.print(" " + i);

System.out.println();

}

}

}

public class Exercise5\_8 {

public static void main(String[] args) {

System.out.println("Celsius\t\tFahrenheit\t|\tFahrenheit\tCelsius");

System.out.println("---------------------------------------------");

double celsius = 40; double farenheit = 120;

for (int i = 1; i <= 10; celsius--, farenheit -= 10, i++) {

System.out.println(celsius + "\t\t" + celsiusToFahrenheit(celsius) + "\t|\t" + farenheit + "\t\t" + fahrenheitToCelsius(farenheit));

}

}

public static double celsiusToFahrenheit(double celsius) {

return (9.0 / 5.0) \* celsius + 32;

}

public static double fahrenheitToCelsius(double fahrenheit) {

return (5.0 / 9) \* (fahrenheit - 32);

}

}

public class Exercise5\_10 {

public static void main(String[] args) {

int count = 0;

for (int number = 2; number < 10000; number++)

if (isPrime(number))

count++;

System.out.println("The number of prime number < 10000 is "

+ count);

}

/\*\* Check whether number is prime \*/

public static boolean isPrime(int number) {

for (int divisor = 2; divisor <= number / 2; divisor++) {

if (number % divisor == 0) { // If true, number is not prime

return false; // number is not a prime

}

}

return true; // number is prime

}

}

public class Exercise5\_12 {

public static void main(String[] args) {

printChars('1', 'Z', 10);

}

public static void printChars(char ch1, char ch2, int numberPerLine) {

int count = 1;

for (int i = ch1; i <= ch2; i++, count++)

if (count % numberPerLine == 0)

System.out.println((char)i);

else

System.out.print((char)i);

}

}

public class Exercise5\_14 {

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("Enter i: ");

int i = input.nextInt();

System.out.println(m(i));

// System.out.println("i\t\tm(i)");

// for (int i = 2; i <= 10000; i++)

// System.out.println(i + "\t\t" + m(i));

}

// New solution after 6E, 2/04/07

public static double m(int n) {

double pi = 0;

double term;

for (int i = 1; i <= n; i += 2) {

term = 4.0 \* (1.0 / (2 \* i - 1) - 1.0 / (2 \* i + 1));

pi += term;

}

return pi;

}

// old solution prior to 6E

// public static double m(int n) {

// double pi = 0;

// double term;

// int sign = 1;

//

// for (int i = 1; i <= n; i++) {

// term = sign \* 4.0 / (2 \* i - 1);

// pi += term;

// sign = -1 \* sign;

// }

//

// return pi;

// }

}

public class Exercise5\_16 {

public static void main(String[] args) {

for (int year = 2000; year <= 2010; year++) {

System.out.println(year + " has " + numberOfDaysInAYear(year));

}

}

public static int numberOfDaysInAYear(int year) {

if (isLeapYear(year)) {

return 366;

}

else {

return 365;

}

}

/\*\* Determine if it is a leap year \*/

static boolean isLeapYear(int year) {

return year % 400 == 0 || (year % 4 == 0 && year % 100 != 0);

}

}

public class Exercise5\_18 {

public static void main(String[] args) {

System.out.println("RealNumber\t\tSquareRoot");

System.out.println("-------------------------------");

for (int num = 0; num <= 20; num++) {

System.out.println(num + "\t\t\t" + Math.sqrt(num));

}

}

}

public class Exercise5\_20 {

public static void main(String[] args) {

System.out.println("Degree\tSin\t\tCos");

for (int degree = 0; degree <= 360; degree += 10) {

System.out.printf("%3d\t%6.4f\t\t%6.4f\n", degree,

Math.sin(degree \* Math.PI / 180),

Math.cos(degree \* Math.PI / 180));

}

}

/\*

public static double format(double d, int position) {

return Math.round(d \* Math.pow(10, position)) / Math.pow(10, position);

} \*/

}

public class Exercise5\_22 {

// Main method

public static void main(String[] args) {

System.out.print("Enter a positive number: ");

java.util.Scanner input = new java.util.Scanner(System.in);

double number = input.nextDouble();

System.out.println(

"The square root for 9 is " + SquareRoot.sqrt(number));

}

}

// This class contains sqrt method

class SquareRoot {

// Find the square root of the value

public static double sqrt(double num) {

double nextGuess = 1.0;

double lastGuess;

do {

lastGuess = nextGuess;

nextGuess = (lastGuess + (num / lastGuess)) \* 0.5;

} while (Math.abs(nextGuess - lastGuess) >= 0.00001);

return nextGuess;

}

}

import javax.swing.JOptionPane;

public class Exercise5\_24 {

public static void main(String[] args) {

// Obtain the total milliseconds since the midnight, Jan 1, 1970

long totalMilliseconds = System.currentTimeMillis();

// Obtain the total seconds since the midnight, Jan 1, 1970

long totalSeconds = (totalMilliseconds / 1000);

// Compute the current second in the minute in the hour

int currentSecond = (int)(totalSeconds % 60);

// Obtain the total minutes

long totalMinutes = totalSeconds / 60;

// Compute the current minute in the hour

int currentMinute = (int)(totalMinutes % 60);

// Obtain the total hours

long totalHours = totalMinutes / 60;

// Compute the current hour

int currentHour = (int)(totalHours % 24);

// Compute the total days

int totalDays = (int)(totalHours / 24);

if (currentHour > 0) totalDays++;

// Obtain Year

int currentYear = 2000;

do {

currentYear++;

} while (getTotalDaysInYears(currentYear) < totalDays);

// Obtain Month

int totalNumOfDaysInTheYear = totalDays -

getTotalDaysInYears(currentYear - 1);

int currentMonth = 0;

do {

currentMonth++;

} while (getTotalDaysInMonths(currentYear, currentMonth)

< totalNumOfDaysInTheYear);

// Obtain Day

int currentDay = totalNumOfDaysInTheYear -

getTotalDaysInMonths(currentYear, currentMonth - 1);

// Display results

String output = "Current date and time is " +

currentMonth + "/" + currentDay + "/" + currentYear + " " +

currentHour + ":" +

+ currentMinute + ":" + currentSecond + " GMT";

JOptionPane.showMessageDialog(null, output);

}

/\*\* Get the total number of days from Jan 1, 1970 to the specified year\*/

static int getTotalDaysInYears(int year) {

int total = 0;

// Get the total days from 1970 to the specified year

for (int i = 1970; i <= year; i++)

if (isLeapYear(i))

total = total + 366;

else

total = total + 365;

return total;

}

/\*\* Get the total number of days from Jan 1 to the month in the year\*/

static int getTotalDaysInMonths(int year, int month) {

int total = 0;

// Add days from Jan to the month

for (int i = 1; i <= month; i++)

total = total + getNumOfDaysInMonth(year, i);

return total;

}

/\*\* Get the number of days in a month \*/

static int getNumOfDaysInMonth(int year, int month) {

if (month == 1 || month==3 || month == 5 || month == 7 ||

month == 8 || month == 10 || month == 12)

return 31;

if (month == 4 || month == 6 || month == 9 || month == 11)

return 30;

if (month == 2)

if (isLeapYear(year))

return 29;

else

return 28;

return 0; // If month is incorrect.

}

/\*\* Determine if it is a leap year \*/

static boolean isLeapYear(int year) {

if ((year % 400 == 0) || ((year % 4 == 0) && (year % 100 != 0)))

return true;

return false;

}

}

public class Exercise5\_26T {

public static void main(String[] args) {

int count = 1;

for (int i = 2; true; i++) {

// Display each number in five positions

if (isPrime(i) && isPalindrome(i)) {

System.out.print(i + " ");

if (count % 10 == 0) {

System.out.println();

}

if (count == 100) {

break;

}

count++; // Increase count

}

}

}

public static boolean isPrime(int num) {

if ((num == 1) || (num == 2)) {

return true;

}

for (int i = 2; i <= num / 2; i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

static int reversal(int num) {

int result = 0;

while (num != 0) {

int lastDigit = num % 10;

result = result \* 10 + lastDigit;

num = num / 10;

}

return result;

}

static boolean isPalindrome(int num) {

return num == reversal(num);

}

}

public class Exercise5\_28 {

public static void main(String[] args) {

long beginTime = System.currentTimeMillis();

for (int p = 2; p <= 31; p++) {

int i = (int) (Math.pow(2, p) - 1);

// Display each number in five positions

if (isPrime(i)) {

System.out.println(p + "\t" + i);

}

}

long endTime = System.currentTimeMillis();

System.out.println("Time spent is " + (endTime - beginTime)

+ " milliseconds");

}

public static boolean isPrime(int num) {

if ((num == 1) || (num == 2)) {

return true;

}

for (int i = 2; i <= num / 2; i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

static int reversal(int num) {

int result = 0;

while (num != 0) {

int lastDigit = num % 10;

result = result \* 10 + lastDigit;

num = num / 10;

}

return result;

}

static boolean isPalindrome(int num) {

return num == reversal(num);

}

}

public class Exercise5\_30 {

public static void main(String[] args) {

int p1 = 2;

for (int p = 2; p <= 1000; p++) {

if (isPrime(p)) {

if (p - p1 == 2) {

System.out.println("(" + p1 + ", " + p + ")");

}

p1 = p;

}

}

}

public static boolean isPrime(int num) {

if ((num == 1) || (num == 2)) {

return true;

}

for (int i = 2; i <= num / 2; i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

}

public class Exercise5\_32 {

public static void main(String[] args) {

int winCount = 0;

for (int i = 0; i < 10000; i++) {

int dice = getDice();

if (dice == 7 || dice == 11) {

winCount++;

}

else if (dice == 2 || dice == 3 || dice == 12) {

//System.out.println("You lose");

}

else {

int point = dice;

//System.out.println("point is " + point);

do {

dice = getDice();

} while (dice != 7 && dice != point);

if (dice == 7) {

//System.out.println("You lose");

}

else {

winCount++;

}

}

}

System.out.println(winCount);

}

// Get a dice

public static int getDice() {

int i1 = 1 + (int)(Math.random() \* 6);

int i2 = 1 + (int)(Math.random() \* 6);

// System.out.println("You rolled " + i1 + " + " + i2 + " = " + (i1 + i2));

return i1 + i2;

}

}

import java.util.Scanner;

public class Exercise5\_34 {

/\*\* Main method \*/

public static void main(String[] args) {

// Prompt the user to enter year

Scanner input = new Scanner(System.in);

// Prompt the user to enter year

System.out.print("Enter full year (e.g., 2001): ");

int year = input.nextInt();

// Prompt the user to enter month

System.out.print("Enter month in number between 1 and 12: ");

int month = input.nextInt();

// Print calendar for the month of the year

printMonth(year, month);

}

/\*\* Print the calendar for a month in a year \*/

static void printMonth(int year, int month) {

// Print the headings of the calendar

printMonthTitle(year, month);

// Print the body of the calendar

printMonthBody(year, month);

}

/\*\* Print the month title, e.g., May, 1999 \*/

static void printMonthTitle(int year, int month) {

System.out.println(" " + getMonthName(month)

+ " " + year);

System.out.println("-----------------------------");

System.out.println(" Sun Mon Tue Wed Thu Fri Sat");

}

/\*\* Get the English name for the month \*/

static String getMonthName(int month) {

String monthName = null;

switch (month) {

case 1: monthName = "January"; break;

case 2: monthName = "February"; break;

case 3: monthName = "March"; break;

case 4: monthName = "April"; break;

case 5: monthName = "May"; break;

case 6: monthName = "June"; break;

case 7: monthName = "July"; break;

case 8: monthName = "August"; break;

case 9: monthName = "September"; break;

case 10: monthName = "October"; break;

case 11: monthName = "November"; break;

case 12: monthName = "December";

}

return monthName;

}

/\*\* Print month body \*/

static void printMonthBody(int year, int month) {

// Get start day of the week for the first date in the month

int startDay = getStartDay(year, month);

// Get number of days in the month

int numberOfDaysInMonth = getNumberOfDaysInMonth(year, month);

// Pad space before the first day of the month

int i = 0;

for (i = 0; i < startDay; i++)

System.out.print(" ");

for (i = 1; i <= numberOfDaysInMonth; i++) {

if (i < 10)

System.out.print(" " + i);

else

System.out.print(" " + i);

if ((i + startDay) % 7 == 0)

System.out.println();

}

System.out.println();

}

/\*\* Get the start day of month/1/year \*/

static int getStartDay(int year, int month) {

return getDayofWeek(year, month, 1);

}

public static int getDayofWeek(int year, int month, int dayOfMonth) {

if (month == 1) {

month = 13;

year--;

}

else if (month == 2) {

month = 14;

year--;

}

int k = year % 100;

int j = (int)(year / 100);

int dayOfWeek = (int)(dayOfMonth + (int)((month + 1) \* 26.0 / 10)

+ k + (int)(k / 4.0) + (int)(j / 4.0) + 5 \* j) % 7;

return (dayOfWeek + 6) % 7;

}

/\*\* Get the number of days in a month \*/

static int getNumberOfDaysInMonth(int year, int month) {

if (month == 1 || month == 3 || month == 5 || month == 7 ||

month == 8 || month == 10 || month == 12)

return 31;

if (month == 4 || month == 6 || month == 9 || month == 11)

return 30;

if (month == 2) return isLeapYear(year) ? 29 : 28;

return 0; // If month is incorrect

}

/\*\* Determine if it is a leap year \*/

static boolean isLeapYear(int year) {

return year % 400 == 0 || (year % 4 == 0 && year % 100 != 0);

}

}

import java.util.Scanner;

public class Exercise5\_36 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Enter the number of sides

System.out.print("Enter the number of sides: ");

int numberOfSides = input.nextInt();

System.out.print("Enter the side: ");

double side = input.nextDouble();

System.out.println("The area of the polygon is " +

area(numberOfSides, side));

}

public static double area(int n, double side) {

return n \* side \* side / Math.tan(Math.PI / n) / 4;

}

}

// Exercise6\_2.java: Enter 10 integers and

// display the numbers in reverse order

public class Exercise6\_2 {

public static void main (String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

int[] num = new int[10];

for (int i = 0; i < 10; i++) {

// Read a number

System.out.print(

"Read a number: ");

num[i] = input.nextInt();

}

// Display the array

for (int i = 9; i >= 0; i--) {

System.out.println(num[i]);

}

}

}

// Exercise6\_4.java: Analyze scores

public class Exercise6\_4 {

// Main method

public static void main(String[] args) {

double[] scores = new double[10];

double sum = 0;

int count = 0;

java.util.Scanner input = new java.util.Scanner(System.in);

do {

System.out.print("Enter a new score: ");

scores[count] = input.nextDouble();

if (scores[count] >= 0)

sum += scores[count];

} while (scores[count++] >= 0);

System.out.println("count is " + count);

double average = (sum) / (count - 1);

int numOfAbove = 0;

int numOfBelow = 0;

for (int i = 0; i < count - 1; i++)

if (scores[i] >= average)

numOfAbove++;

else

numOfBelow++;

System.out.println("Average is " + average);

System.out.println("Number of scores above or equal to the average "

+ numOfAbove);

System.out.println("Number of scores below the average "

+ numOfBelow);

}

}

public class Exercise6\_6 {

/\*\* Main method \*/

public static void main(String[] args) {

final int NUM\_OF\_PRIMES = 50;

// Store prime numbers

int[] primeNumbers = new int[NUM\_OF\_PRIMES];

int count = 0; // Count the number of prime numbers

int number = 2; // A number to be tested for primeness

boolean isPrime = true; // Is the current number prime?

System.out.println("The first 50 prime numbers are \n");

// Repeatedly find prime numbers

while (count < NUM\_OF\_PRIMES) {

// Assume the number is prime

isPrime = true;

// Exercise3\_21 if number is prime

for (int i = 0; i < count &&

primeNumbers[i] <= Math.sqrt(number); i++) {

//If true, the number is not prime

if (number % primeNumbers[i] == 0) {

// Set isPrime to false, if the number is not prime

isPrime = false;

break; // Exit the for loop

}

}

// Print the prime number and increase the count

if (isPrime) {

primeNumbers[count] = number;

count++; // Increase the count

if (count % 10 == 0) {

// Print the number and advance to the new line

System.out.println(number);

}

else

System.out.print(number + "\t");

}

// Check if the next number is prime

number++;

}

}

}

// Exercise6\_8.java:

public class Exercise6\_8 {

// Main method

public static void main(String[] args) {

int[] list1 = {1, 2, 3, 4, 5, 6};

double[] list2 = {6.0, 4.4, 1.9, 2.9, 3.4, 3.5};

System.out.println(average(list1));

System.out.println(average(list2));

}

public static int average(int[] array) {

int sum = 0;

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

sum += array[i];

return sum / array.length;

}

public static double average(double[] array) {

double sum = 0;

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

sum += array[i];

return sum / array.length;

}

}

// Exercise6\_10.java: Write a method that finds the index of

// the smallest element in an array of integers.

public class Exercise6\_10 {

public static void main(String[] args) {

double[] numbers = new double[10];

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("Enter ten numbers: ");

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

numbers[i] = input.nextDouble();

System.out.println("The index of the min is " +

indexOfSmallestElement(numbers));

}

public static int indexOfSmallestElement(double[] list) {

double min = list[0];

int minIndex = 0;

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

if (min > list[i]) {

min = list[i];

minIndex = i;

}

return minIndex;

}

}

public class Exercise6\_12 {

public static void main(String[] args) {

int[] myList = {1, 2, 3, 4, 5, 6, 7, 8};

reverse(myList);

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

System.out.print(myList[i] + " ");

}

public static int[] reverse(int[] list) {

for (int i = 0, j = list.length - 1; i < list.length / 2; i++, j--) {

int temp = list[i];

list[i] = list[j];

list[j] = temp;

}

return list;

}

}

public class Exercise6\_14 {

public static void main(String args[]) {

System.out.print("Enter five integers: ");

java.util.Scanner input = new java.util.Scanner(System.in);

double number = input.nextDouble();

System.out.println(

"The square root for 9 is " + SquareRoot.sqrt(number));

System.out.println(gcd(33, 33, 33, 11, 22));

}

public static int gcd(int... numbers) {

int gcd = numbers[0];

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

gcd = gcd(gcd, numbers[i]);

return gcd;

}

/\*\* Return the gcd of two integers \*/

public static int gcd(int n1, int n2) {

int gcd = 1; // Initial gcd is 1

int k = 1; // Possible gcd

while (k <= n1 && k <= n2) {

if (n1 % k == 0 && n2 % k == 0)

gcd = k; // Update gcd

k++;

}

return gcd; // Return gcd

}

}

public class Exercise6\_16 {

public static void main(String[] args) {

int[] list = new int[100000];

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

list[i] = (int)(Math.random() \* 1000000);

}

int key = (int)(Math.random() \* 1000000);

long startTime = System.currentTimeMillis();

System.out.println(LinearSearch.linearSearch(list, key));

long endTime = System.currentTimeMillis();

long executionTime = endTime - startTime;

System.out.println("Execution time for linear search is " +

executionTime);

java.util.Arrays.sort(list);

startTime = System.currentTimeMillis();

System.out.println(BinarySearch.binarySearch(list, key));

endTime = System.currentTimeMillis();

executionTime = endTime - startTime;

System.out.println("Execution time for binary search is " +

executionTime);

}

}

// Bubble sort

public class Exercise6\_18 {

public static void main (String[] args) {

double[] myList = {5.0, 4.4, 1.9, 2.9, 3.4, 2.9, 3.5};

System.out.println("My list before sort is: ");

//prints the original list

printList(myList);

bubbleSort(myList);

//prints the sorted list

System.out.println("My list after sort is: ");

printList(myList);

}

static void printList(double[] list) {

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

System.out.println(list[i]);

}

static void bubbleSort(double[] list) {

boolean changed = true;

do {

changed = false;

for (int j = 0; j < list.length - 1; j++)

if (list[j] > list[j+1]) {

//swap list[j] wiht list[j+1]

double temp = list[j];

list[j] = list[j + 1];

list[j + 1] = temp;

changed = true;

}

} while (changed);

}

}

public class Exercise6\_20 {

public static void main(String[] args) {

// Queen positions

int[] queens = new int[8]; // queens are placed at (i, queens[i])

for (int i = 0; i < 8; i++)

queens[i] = -1; // -1 indicates that no queen is currently placed in the

// ith row

queens[0] = 0; // Initially, place a queen at (0, 0) in the 0th row

// k - 1 indicates the number of queens placed so far

// We are looking for a position in the kth row to place a queen

int k = 1;

while (k >= 0 && k <= 7) {

// Find a position to place a queen in the kth row

int j = findPosition(k, queens);

if (j < 0) {

queens[k] = -1;

k--; // back track to the previous row

} else {

queens[k] = j;

k++;

}

}

printResult(queens);

}

public static int findPosition(int k, int[] queens) {

int start = queens[k] == -1 ? 0 : queens[k] + 1;

for (int j = start; j < 8; j++) {

if (isValid(k, j, queens))

return j; // (k, j) is the place to put the queen now

}

return -1;

}

/\*\* Return true if you a queen can be placed at (k, j) \*/

public static boolean isValid(int k, int j, int queens[]) {

// See if (k, j) is a possible position

// Check jth column

for (int i = 0; i < k; i++)

if (queens[i] == j)

return false;

// Check major diagnol

for (int row = k - 1, column = j - 1; row >= 0 && column >= 0; row--, column--)

if (queens[row] == column)

return false;

// Check minor diagnol

for (int row = k - 1, column = j + 1; row >= 0 && column <= 7; row--, column++)

if (queens[row] == column)

return false;

return true;

}

/\*\* Print a two-dimensional board to display the queens \*/

public static void printResult(int[] queens) {

for (int i = 0; i < 8; i++)

System.out.println(i + ", " + queens[i]);

System.out.println();

// Display the output

for (int i = 0; i < 8; i++) {

for (int j = 0; j < queens[i]; j++)

System.out.print("| ");

System.out.print("|Q|");

for (int j = queens[i] + 1; j < 8; j++)

System.out.print(" |");

System.out.println();

}

}

}

public class Exercise6\_22 {

public static void main(String[] args) {

int count = 0; // How many solutions are found?

// Queen positions

int[] queens = new int[8]; // queens are placed at (i, queens[i])

for (int i = 0; i < 8; i++)

queens[i] = -1; // -1 indicates that no queen is currently placed in the

// ith row

queens[0] = 0; // Initially, place a queen at (0, 0) in the 0th row

// k - 1 indicates the number of queens placed so far

// We are looking for a position in the kth row to place a queen

int k = 1;

while (k >= 0) {

// Find a position to place a queen in the kth row

int j = findPosition(k, queens);

if (j < 0) {

queens[k] = -1;

k--; // back track to the previous row

} else {

queens[k] = j;

if (k == 7) {

count++; // One more solution found

System.out.println("Solution " + count + ":");

printResult(queens);

} else {

k++;

}

}

}

System.out.println("How many solutions? " + count);

}

public static int findPosition(int k, int[] queens) {

int start = queens[k] == -1 ? 0 : queens[k] + 1;

for (int j = start; j < 8; j++) {

if (isValid(k, j, queens))

return j; // (k, j) is the place to put the queen now

}

return -1;

}

/\*\* Return true if you a queen can be placed at (k, j) \*/

public static boolean isValid(int k, int j, int queens[]) {

// See if (k, j) is a possible position

// Check jth column

for (int i = 0; i < k; i++)

if (queens[i] == j)

return false;

// Check major diagnol

for (int row = k - 1, column = j - 1; row >= 0 && column >= 0; row--, column--)

if (queens[row] == column)

return false;

// Check minor diagnol

for (int row = k - 1, column = j + 1; row >= 0 && column <= 7; row--, column++)

if (queens[row] == column)

return false;

return true;

}

/\*\* Print a two-dimensional board to display the queens \*/

public static void printResult(int[] queens) {

// Display the output

for (int i = 0; i < 8; i++) {

for (int j = 0; j < queens[i]; j++)

System.out.print("| ");

System.out.print("|Q|");

for (int j = queens[i] + 1; j < 8; j++)

System.out.print(" |");

System.out.println();

}

}

}

public class Exercise6\_24 {

public static void main(String[] args) {

final int NUMBER\_OF\_CARDS = 52;

String[] suits = {"Clubs", "Diamonds", "Hearts", "Spades"};

String[] ranks = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9",

"10", "Jack", "Queen", "King"};

// found indicates whether a suit has been picked

boolean[] found = new boolean[4];

// Count the number of picks

int numberOfPicks = 0;

// Count occurrence in each suit

int count = 0;

while (count < 4) {

numberOfPicks++;

int index = (int)(Math.random() \* NUMBER\_OF\_CARDS);

if (!found[index / 13]) {

found[index / 13] = true;

count++;

String suit = suits[index / 13];

String rank = ranks[index % 13];

System.out.println(rank + " of " + suit);

}

}

System.out.println("Number of picks: " + numberOfPicks);

}

}

public class Exercise6\_26 {

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Enter values for list1

System.out.print("Enter list1: ");

int size1 = input.nextInt();

int[] list1 = new int[size1];

for (int i = 0; i < list1.length; i++)

list1[i] = input.nextInt();

// Enter values for list2

System.out.print("Enter list2: ");

int size2 = input.nextInt();

int[] list2 = new int[size2];

for (int i = 0; i < list2.length; i++)

list2[i] = input.nextInt();

if (equals(list1, list2)) {

System.out.println("Two lists are strictly identical");

}

else {

System.out.println("Two lists are not strictly identical");

}

}

public static boolean equals(int[] list1, int[] list2) {

if (list1.length != list2.length)

return false;

for (int i = 0; i < list1.length; i++)

if (list1[i] != list2[i])

return false;

return true;

}

}

import java.util.Scanner;

public class Exercise6\_28 {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter ten integers: ");

final int N = 10;

int[] numbers = new int[N];

for (int i = 0; i < N; i++)

numbers[i] = input.nextInt();

for (int i = 0; i < 8; i++)

for (int j = i + 1; j < 8; j++)

System.out.println(numbers[i] + " " + numbers[j]);

}

}

public class Exercise6\_30 {

public static void main (String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("Enter the number of values: ");

int size = input.nextInt();

int[] values = new int[size];

System.out.print("Enter the values: ");

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

values[i] = input.nextInt();

if (isConsecutiveFour(values))

System.out.println("The series has consecutive fours");

else

System.out.println("The series has no consecutive fours");

}

public static boolean isConsecutiveFour(int[] values) {

for (int i = 0; i < values.length - 3; i++) {

boolean isEqual = true;

for (int j = i; j < i + 3; j++) {

if (values[j] != values[j + 1]) {

isEqual = false;

break;

}

}

if (isEqual) return true;

}

return false;

}

public static boolean isConsecutiveFourVersionII(int[] values) {

outer: for (int i = 0; i < values.length - 3; i++) {

for (int j = i; j < i + 3; j++)

if (values[j] != values[j + 1])

continue outer; // See Supplement Part III.E statement labels

return true;

}

return false;

}

}

public class Exercise6\_32 {

public static void main (String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

// Enter values for list1

System.out.print("Enter a list: ");

int size = input.nextInt();

int[] list = new int[size];

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

list[i] = input.nextInt();

partition(list);

System.out.print("After the partition, the list is ");

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

System.out.print(list[i] + " ");

}

/\*\* Partition the array list[first..last] \*/

public static int partition(int[] list) {

int first = 0;

int last = list.length - 1;

int pivot = list[first]; // Choose the first element as the pivot

int low = first + 1; // Index for forward search

int high = last; // Index for backward search

while (high > low) {

// Search forward from left

while (low <= high && list[low] <= pivot)

low++;

// Search backward from right

while (low <= high && list[high] > pivot)

high--;

// Swap two elements in the list

if (high > low) {

int temp = list[high];

list[high] = list[low];

list[low] = temp;

}

}

while (high > first && list[high] >= pivot)

high--;

// Swap pivot with list[high]

if (pivot > list[high]) {

list[first] = list[high];

list[high] = pivot;

return high;

}

else {

return first;

}

}

}

import java.util.Scanner;

public class Exercise6\_36UsingContinueStatement {

public static void main(String[] args) {

// Read a Sudoku puzzle

int[][] grid = readAPuzzle();

if (!isValid(grid))

System.out.println("Invalid input");

else {

int count = search(grid);

if (count == 0)

System.out.println("No solution");

else

System.out.println("There are " + count + " solutions");

}

}

/\*\* Read a Sudoku puzzle from the keyboard \*/

public static int[][] readAPuzzle() {

// Create a Scanner

Scanner input = new Scanner(System.in);

System.out.println("Enter a Sudoku puzzle:");

int[][] grid = new int[9][9];

for (int i = 0; i < 9; i++)

for (int j = 0; j < 9; j++)

grid[i][j] = input.nextInt();

return grid;

}

/\*\* Obtain a list of free cells from the puzzle \*/

public static int[][] getFreeCellList(int[][] grid) {

// 81 is the maximum number of free cells

int[][] tempList = new int[81][2];

int numberOfFreeCells = 0;

for (int i = 0; i < 9; i++)

for (int j = 0; j < 9; j++)

if (grid[i][j] == 0) {

tempList[numberOfFreeCells][0] = i;

tempList[numberOfFreeCells][1] = j;

numberOfFreeCells++;

}

// Trim freeCellList

int[][] freeCellList = new int[numberOfFreeCells][2];

for (int i = 0; i < numberOfFreeCells; i++) {

freeCellList[i][0] = tempList[i][0];

freeCellList[i][1] = tempList[i][1];

}

return freeCellList;

}

/\*\* Print the values in the grid \*/

public static void printGrid(int[][] grid) {

for (int i = 0; i < 9; i++) {

for (int j = 0; j < 9; j++)

System.out.print(grid[i][j] + " ");

System.out.println();

}

}

/\*\* Search for a solution \*/

public static int search(int[][] grid) {

int[][] freeCellList = getFreeCellList(grid); // Free cells

int k = 0; // Start from the first free cell

boolean found = false; // Solution found?

int count = 0; // Multiple solutions: Count for 2 solutions

while (!found) {

int i = freeCellList[k][0];

int j = freeCellList[k][1];

if (grid[i][j] == 0)

grid[i][j] = 1; // Start with 1

if (isValid(i, j, grid)) {

if (k + 1 == freeCellList.length) { // No more free cells

// Multiple solutions: backtrack

count++;

if (count <= 3) {

System.out.println("Sample solution " + (count) + ":");

printGrid(grid); // Multiple solutions: print a solution

}

if (grid[i][j] < 9) {

grid[i][j] = grid[i][j] + 1; // Check the next possible value

} else {

while (grid[i][j] == 9) {

grid[i][j] = 0; // Reset to free cell

if (k == 0) {

return count; // No possible value

}

k--; // Backtrack

i = freeCellList[k][0];

j = freeCellList[k][1];

grid[i][j] = grid[i][j] + 1; // Check the next possible value

}

continue; // Stay in the loop

}

} else { // Move to the next free cell

k++;

}

} else if (grid[i][j] < 9) {

grid[i][j] = grid[i][j] + 1; // Check the next possible value

} else { // grid[i][j] is 9, backtrack

while (grid[i][j] == 9) {

grid[i][j] = 0; // Reset to free cell

if (k == 0) {

return count; // No possible value

}

k--; // Backtrack

i = freeCellList[k][0];

j = freeCellList[k][1];

}

grid[i][j] = grid[i][j] + 1; // Check the next possible value

}

}

return count; // A solution is found

}

/\*\* Check whether grid[i][j] is valid in the grid \*/

public static boolean isValid(int i, int j, int[][] grid) {

// Check whether grid[i][j] is valid at the i's row

for (int column = 0; column < 9; column++)

if (column != j && grid[i][column] == grid[i][j])

return false;

// Check whether grid[i][j] is valid at the j's column

for (int row = 0; row < 9; row++)

if (row != i && grid[row][j] == grid[i][j])

return false;

// Check whether grid[i][j] is valid in the 3 by 3 box

for (int row = (i / 3) \* 3; row < (i / 3) \* 3 + 3; row++)

for (int col = (j / 3) \* 3; col < (j / 3) \* 3 + 3; col++)

if (row != i && col != j && grid[row][col] == grid[i][j])

return false;

return true; // The current value at grid[i][j] is valid

}

/\*\* Check whether the fixed cells are valid in the grid \*/

public static boolean isValid(int[][] grid) {

for (int i = 0; i < 9; i++)

for (int j = 0; j < 9; j++)

if (grid[i][j] != 0 && !isValid(i, j, grid))

return false;

return true; // The fixed cells are valid

}

}

5 201

25 2 1 100.5 4 320.5

125 2 2 40 3 85

175 2 0 125 3 75

75 1 0 125

181 1 2 125