1. Andi memiliki sejumlah uang dan dengan uang itu ia ingin membeli beberapa barang. Jika Andi ingin menggunakan uang itu semaksimal mungkin, buatlah sebuah kalkulasi **jumlah maksimal uang yang bisa dipakai** dan **berapa item barang yang bisa dibeli** (jika jumlah uang yang dipakai sama, pilih kombinasi item terbanyak)

Contoh: Uang 1000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nama barang | Harga | | | |
| Kacamata | 500 | 600 | 700 | 800 |
| Baju | 200 | 400 | 350 |  |
| Sepatu | 400 | 350 | 200 | 300 |
| Buku | 100 | 50 | 150 |  |
| ... |  |  |  |  |

Jumlah uang yang dipakai: **1000**

Jumlah item yang bisa dibeli: **4** (kacamata 500, baju 200, sepatu 200, buku 100)

**Jawab:**

1. Di perpustakaan Anda bisa meminjam buku selama beberapa hari, durasinya berbeda-beda setiap buku tergantung dari konten buku tersebut. Jika terlambat mengembalikan, maka akan dikenakan denda sebesar 100/hari.

Berikut adalah contoh buku yang dipinjam seorang anggota perpustakaan.

|  |  |
| --- | --- |
| Nama Buku | Durasi Peminjaman (hari) |
| A | 14 |
| B | 3 |
| C | 7 |
| D | 7 |

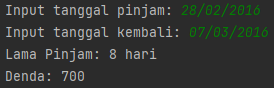
Buatlah kalkulasi penghitungan denda jika buku tersebut dipinjam pada rentang waktu

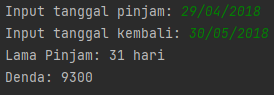
1. 28 Februari 2016 – 7 Maret 2016\*
2. 29 April 2018 – 30 Mei 2018

Jika tidak ada denda, cukup tulis 0 (nol). Asumsikan perpustakaan buka setiap hari, tidak ada hari libur.

***\*Tahun 2016 adalah tahun kabisat***

**Jawab:**

****

****

**Script:**

package BLQ;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Scanner;  
  
public class Soal02 {  
 public static void main(String[] args) throws ParseException {  
 Scanner scan = new Scanner(System.*in*);  
  
 //dd/mm/yyy  
 // 28/02/2016 - 07/03/2016  
 // 29/04/2018 - 30/05/2018  
 System.*out*.print("Input tanggal pinjam: ");  
 String pinjam = scan.nextLine().trim();  
  
 System.*out*.print("Input tanggal kembali: ");  
 String kembali = scan.nextLine().trim();  
  
 *cara1*(pinjam,kembali);  
 }  
  
 static void cara1(String pinjam, String kembali) throws ParseException {  
 SimpleDateFormat timeFormat = new SimpleDateFormat("dd/MM/yyy");  
 Date timePinjam = timeFormat.parse(pinjam);  
 Date timeKembali = timeFormat.parse(kembali);  
  
 long totalMiliSecond = Math.*abs*(timeKembali.getTime()-timePinjam.getTime());  
 long lama = (totalMiliSecond/(1000\*60\*60\*24));  
 int lama2 = (int) lama;  
 System.*out*.println("Lama Pinjam: " + lama2 + " hari");  
 int denda = 0;  
  
 if (lama2 > 14){  
 denda = (lama2 - 14) + (lama2 - 3) + (lama2 - 7) + (lama2 - 7);  
 }  
 else if (lama2 > 7) {  
 denda = (lama2 - 3) + (lama2 - 7) + (lama2 - 7);  
 }  
 else if (lama2 > 3) {  
 denda = (lama2 - 3);  
 }  
 else {  
 denda = 0;  
 System.*out*.println("Tidak ada denda!");  
 }  
 System.*out*.println("Denda: " + denda\*100);  
 }  
}

1. Buatlah fungsi untuk kalkulasi tarif parkir berikut.

|  |  |  |
| --- | --- | --- |
| Masuk | Keluar | Tarif |
| 27 Januari 2019 | 05:00:01 | 27 Januari 2019 | 17:45:03 |  |
| 27 Januari 2019 | 07:03:59 | 27 Januari 2019 | 15:30:06 |  |
| 27 Januari 2019 | 07:05:00 | 28 Januari 2019 | 00:20:21 |  |
| 27 Januari 2019 | 11:14:23 | 27 Januari 2019 | 13:20:00 |  |

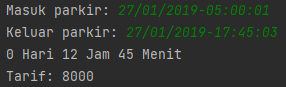
Ketentuan tarif:

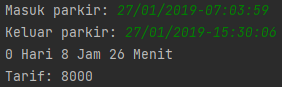
* 8 jam pertama : 1.000/jam
* 8 jam 1 detik – 24 jam : 8.000/flat
* 24 jam 1 detik – 8 jam kemudian : 15.000 flat + 1.000/jam

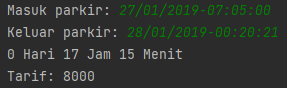
Misalnya parkir selama 30 jam, maka tarifnya adalah:

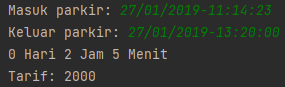
15.000 + (30-24)\*1.000 = 15.000 + 6.000 = 21.000

**Jawab:**

****

****

****

****

**Script:**

package BLQ;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Scanner;  
  
public class Soal03 {  
 public static void main(String[] args) throws ParseException {  
 Scanner scan = new Scanner(System.*in*);  
  
 // dd/MM/yyyy-HH:mm  
 // 27/01/2019-05:00:01 27/01/2019-17:45:03  
 // 27/01/2019-07:03:59 27/01/2019-15:30:06  
 // 27/01/2019-07:05:00 28/01/2019-00:20:21  
 // 27/01/2019-11:14:23 27/01/2019-13:20:00  
 System.*out*.print("Masuk parkir: ");  
 String masuk = scan.nextLine().trim();  
  
 System.*out*.print("Keluar parkir: ");  
 String keluar = scan.nextLine().trim();  
  
 scan.close();  
 *cara2*(masuk,keluar);  
 }  
 static void cara2(String masuk, String keluar) throws ParseException{  
 long tarif = 0;  
 int countHari = 0;  
 int countJam = 0;  
 int perHari = 15000;  
 int perJamPertama = 1000;  
 int perJamBerikutnya = 8000;  
  
 SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy-HH:mm:ss");  
 Date dateMasuk = dateFormat.parse(masuk);  
 Date dateKeluar = dateFormat.parse(keluar);  
  
 long diffMSec = dateKeluar.getTime()-dateMasuk.getTime();  
 long diffMin = diffMSec/(60\*1000);  
  
 while (diffMin >= 1440){  
 countHari += 1;  
 diffMin -= 1440;  
 }  
 while (diffMin >= 60){  
 countJam += 1;  
 diffMin -= 60;  
 }  
 tarif = (countHari \* perHari);  
 if (countJam <= 8){  
 tarif = (tarif +(countJam \* perJamPertama));  
 }  
 else if (countJam > 8) {  
 tarif = (tarif + perJamBerikutnya);  
 }  
 System.*out*.println(countHari + " Hari " + countJam + " Jam " + diffMin + " Menit");  
 System.*out*.println("Tarif: " + tarif);  
 }  
}

1. Buatlah fungsi untuk menampilkan **n** bilangan prima pertama

**Jawab:**

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal04 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("n: ");  
 int panjang = scanner.nextInt();  
  
 int[] arrPrima = new int[panjang];  
 int calonprima = 1;  
 for (int i = 0; i < arrPrima.length;) {  
 int prima = 0;  
 for (int j = 1; j <= calonprima; j++) {  
 if (calonprima%j == 0){  
 prima++;  
 }  
 }  
 if (prima == 2){  
 arrPrima[i] = calonprima;  
 System.*out*.print(calonprima + " ");  
 i++;  
 }  
 calonprima++;  
 }  
 }  
}

1. Buatlah fungsi untuk menampilkan **n** bilangan fibonacci pertama

**Jawab:**

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal05 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("n: ");  
 int panjang = scanner.nextInt();  
  
 int satu = 0;  
 int dua = 1;  
 int[] arrFibo = new int[panjang];  
 for (int i = 0; i < arrFibo.length; i++) {  
 if (i==0){  
 arrFibo[i] = satu;  
 }  
 else if (i == 1) {  
 arrFibo[i] = dua;  
 }  
 else {  
 arrFibo[i] = arrFibo[i-2] + arrFibo[i-1];  
 }  
 System.*out*.print(arrFibo[i] + " ");  
 }  
 }  
}

1. Tanpa menggunakan fungsi reverse, buatlah fungsi untuk menentukan apakah sebuah kata adalah palindrome\* atau tidak

***\*palindrome adalah kata yang jika dibalik tetap sama. Contohnya “katak”, “12021”, “malam”***

**Jawab:**

****

****

****

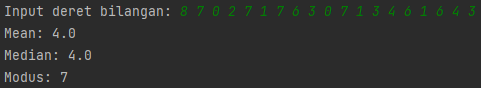
**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal06 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("kata: ");  
 String kata = scanner.nextLine();  
  
 boolean isPalindrom = true;  
 for (int i = 0; i < kata.length(); i++) {  
 char karakterkiri = kata.charAt(i);  
 char karakterkanan = kata.charAt(kata.length()-1-i);  
  
 if (karakterkiri != karakterkanan){  
 System.*out*.println("Bukan kata palindrom");  
 isPalindrom = false;  
 break;  
 }  
 }  
  
 if (isPalindrom == true){  
 System.*out*.println("Kata palindrom");  
 }  
 }  
}

1. Tentukan mean, median, dan modus dari deret berikut. Jika ada lebih dari 2 modus, ambil angka yang nilainya paling kecil

**8 7 0 2 7 1 7 6 3 0 7 1 3 4 6 1 6 4 3**

**Jawab:**

****

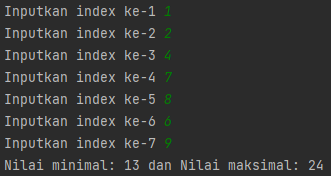
**Script:**

package BLQ;  
  
import java.text.ParseException;  
import java.util.Arrays;  
import java.util.Scanner;  
  
public class Soal07 {  
 public static void main(String[] args) throws ParseException {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.print("Input deret bilangan: ");  
 String[] deret = scan.nextLine().trim().split(" ");  
  
 scan.close();  
 *cara2*(deret);  
 }  
  
 static void cara2(String[] deret) throws ParseException{  
 int[] deretInt = new int[deret.length];  
 for (int i = 0; i < deret.length; i++) {  
 deretInt[i] = Integer.*parseInt*(deret[i]);  
 }  
  
 ///MEAN///  
 double sum = Arrays.*stream*(deretInt).sum();  
 double mean = sum / deretInt.length;  
 System.*out*.println("Mean: " + mean);  
  
 ///MEDIAN///  
 Arrays.*sort*(deretInt);  
 double median;  
 if (deretInt.length % 2 == 0){  
 int tengah1 = deretInt.length / 2 - 1;  
 int tengah2 = deretInt.length / 2;  
 median = (deretInt[tengah1] + deretInt[tengah2]) / 2.0;  
 }  
 else {  
 int tengah = deretInt.length / 2;  
 median = deretInt[tengah];  
 }  
 System.*out*.println("Median: " + median);  
  
 ///MODUS///  
 int maxFreq = 0;  
 int modus = 0;  
 for (int i = 0; i < deretInt.length; i++) {  
 int freq = 1;  
 for (int j = i+1; j < deretInt.length; j++) {  
 if (deretInt[i] == deretInt[j]){  
 freq++;  
 }  
 }  
 if (freq > maxFreq){  
 maxFreq = freq;  
 modus = deretInt[i];  
 }  
 }  
 System.*out*.print("Modus: ");  
 if (maxFreq == 1){  
 System.*out*.println("Tidak ada modus");  
 }  
 else {  
 System.*out*.println(modus);  
 }  
 }  
}

1. Tentukan nilai minimal dan maksimal dari penjumlahan 4 komponen deret ini

**1 2 4 7 8 6 9**

**Jawab:**

****

**Script:**

package BLQ;  
  
import java.util.Arrays;  
import java.util.Scanner;  
  
public class Soal08 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 // 1 2 4 7 8 6 9  
 int[] array = new int[7];  
 for (int i = 0; i < array.length; i++) {  
 System.*out*.print("Inputkan index ke-"+(i+1)+" ");  
 array[i]= scanner.nextInt();  
 }  
 Arrays.*sort*(array);  
 int terkecil = 0;  
 int terbesar = 0;  
 for (int i = 0; i <array.length ; i++) {  
 if (i<=3){  
 terkecil=terkecil+array[i];  
 }  
 else{  
 terbesar=terbesar+array[i];  
 }  
 }  
 System.*out*.println("Nilai minimal: " + terkecil + " dan " + "Nilai maksimal: " + terbesar);  
 }  
}

1. N = 3 3 6 9

N = 4 4 8 12 16

N = 5 5 10 15 20 25

**Jawab:**

****

****

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal09 {  
 public static void main(String[] args) {  
 Scanner input = new Scanner(System.*in*);  
 //N = 3 → 3 6 9  
 //N = 4 → 4 8 12 16  
 //N = 5 → 5 10 15 20 25  
 int jumlah\_deret,i;  
  
 System.*out*.print("n: ");  
 jumlah\_deret = input.nextInt();  
  
 for (i=1; i<=jumlah\_deret; i++) {  
 System.*out*.print(i\*jumlah\_deret + " ");  
 }  
  
 System.*out*.println();  
 }  
}

|  |  |
| --- | --- |
| Input | Output |
| Susilo Bambang Yudhoyono | S\*\*\*o B\*\*\*g Y\*\*\*o |
| Rani Tiara | R\*\*\*i T\*\*\*a |

**Jawab:**

****

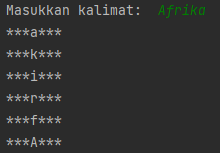
****

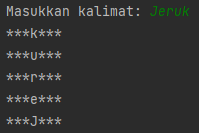
**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal10 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 //Susilo Bambang Yudhoyono  
 //Rani Tiara  
 System.*out*.print("Masukkan nama: ");  
 String nama = scanner.nextLine();  
  
 String[] splitkata = nama.split(" ");  
  
 for (int i = 0; i < splitkata.length; i++) {  
 String[] splithuruf = splitkata[i].split("");  
 for (int j = 0; j < splithuruf.length; j++) {  
 if (j==0){  
 System.*out*.print(splithuruf[j]);  
 }  
 if ( j== splithuruf.length-1) {  
 System.*out*.print("\*\*\*" + splithuruf[j] + " ");  
 }  
 }  
 }  
 }  
}

|  |  |
| --- | --- |
| Input: Afrika | Input: Jeruk |
| Output:  \*\*\*a\*\*\*  \*\*\*k\*\*\*  \*\*\*i\*\*\*  \*\*\*r\*\*\*  \*\*\*f\*\*\*  \*\*\*A\*\*\* | Output:  \*\*\*k\*\*\*  \*\*\*u\*\*\*  \*\*\*r\*\*\*  \*\*\*e\*\*\*  \*\*\*J\*\*\* |

**Jawab:**

****

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal11 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Masukkan kalimat: ");  
 String kalimat = scanner.nextLine();  
  
 String replace = kalimat.replace(" ","");  
  
 String[] split = replace.split("");  
 for (int i = replace.length()-1; i >= 0 ; i--) {  
 System.*out*.println("\*\*\*"+split[i]+"\*\*\*");  
 }  
 }  
}

1. Input : 1 2 1 3 4 7 1 1 5 6 1 8

Output : 1 1 1 1 1 2 3 4 5 6 7 8

***\*selesaikan dengan tidak menggunakan fungsi sort***

**Jawab:**

****

**Script:**

package BLQ;  
  
public class Soal12 {  
 public static void main(String[] args) {  
 int arr[] = {1,2,1,3,4,7,1,1,5,6,1,8};  
  
 System.*out*.print("Output: ");  
  
 for (int i = 0; i < arr.length; i++) {  
  
 for (int j = i + 1; j < arr.length; j++) {  
  
 int temp = 0;  
 if (arr[j] < arr[i]) {  
  
 temp = arr[i];  
 arr[i] = arr[j];  
 arr[j] = temp;  
 }  
 }  
 System.*out*.print(arr[i] + " ");  
 }  
 }  
}

1. Berapa derajat sudut terkecil yang dibentuk 2 jarum jam?

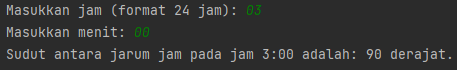
Jam 3:00 90

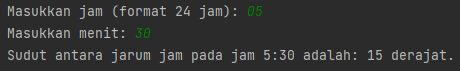
Jam 5:30 15

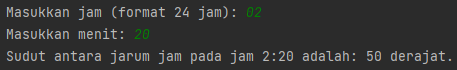
Jam 2:20 50

***\*detik tidak dipertimbangkan***

**Jawab:**

****

****

****

**Script:**

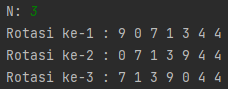
package BLQ;  
  
import java.util.Scanner;  
  
public class Soal13 {  
 public static void main(String[] args) {  
 // Membuat objek Scanner untuk mendapatkan input dari pengguna  
 Scanner scanner = new Scanner(System.*in*);  
  
 //Jam 3:00 → 90  
 //Jam 5:30 → 15  
 //Jam 2:20 → 50  
  
 // Meminta pengguna memasukkan nilai jam  
 System.*out*.print("Masukkan jam (format 24 jam): ");  
 int jam = scanner.nextInt();  
  
 // Meminta pengguna memasukkan nilai menit  
 System.*out*.print("Masukkan menit: ");  
 int menit = scanner.nextInt();  
  
 // Memanggil metode untuk menghitung sudut jarum jam  
 int sudut = *hitungSudutJarumJam*(jam, menit);  
  
 // Menampilkan hasil  
 System.*out*.println("Sudut antara jarum jam pada jam " + jam + ":" + (menit < 10 ? "0" : "") + menit + " adalah: " + sudut + " derajat.");  
  
 // Menutup objek Scanner  
 scanner.close();  
 }  
  
 // Metode untuk menghitung sudut antara jarum jam  
 public static int hitungSudutJarumJam(int jam, int menit) {  
 // Konversi jam ke format 12 jam  
 jam = jam % 12;  
  
 // Menghitung sudut jarum jam  
 int sudutJarumJam = Math.*abs*((60 \* jam - 11 \* menit) / 2);  
  
 // Mengembalikan nilai sudut  
 return sudutJarumJam;  
 }  
}

1. Deret: **3 9 0 7 1 2 4**

N = 3 7 1 2 4 3 9 0

N = 1 9 0 7 1 2 4 1

**Jawab:**

****

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal14 {  
 public static void main(String[] args) {  
 int[] array = {3,9,0,7,1,2,4};  
  
 //N = 3 → 7 1 2 4 3 9 0  
 //N = 1 → 9 0 7 1 2 4 1  
  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("N: ");  
 int rotasi = scanner.nextInt();  
 for (int i = 0; i < rotasi; i++) {  
 int temp = array[0];  
 for (int j = 0; j < array.length - 1; j++) {  
 array[j] = array[j + 1];  
 }  
 array[4] = temp;  
  
 System.*out*.print("Rotasi ke-" + (i+1) + " : ");  
 for (int j = 0; j < array.length; j++) {  
 System.*out*.print(array[j] + " ");  
 }  
 System.*out*.println();  
 }  
 }  
}

1. Ubah format waktu “03:40:44 PM” menjadi format 24 jam (15:40:44)

**Jawab:**

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal15 {  
 public static void main(String[] args) {  
 //Format 12 jam -> panjang 11 15:40:44  
 //Format 24 jam -> panjang 8 03:40:44 PM  
  
 //03:40:44 PM ---> 15:40:44 (24 jam)  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Masukkan Waktu : ");  
 String input = scanner.nextLine();  
  
 String hasil = "";  
 if (input.length() == 11){//Format 12 jam ke format 24 jam  
 if (input.charAt(9) == 'A'){ //AM  
 if (input.substring(0,2).equals("12")){  
 hasil = "00" + input.substring(2,8);  
 } else {  
 hasil = input.substring(0,8);  
 }  
 } else if(input.charAt(9) == 'P'){ //PM  
 if (input.substring(0,2).equals("12")){  
 hasil = input.substring(0,8);  
 } else {  
 int jam = Integer.*parseInt*(input.substring(0,2));  
 jam += 12;  
 hasil = jam + input.substring(2,8);  
 }  
 }  
 } else if (input.length() == 8){//Format 24 jam ke format 12 jam  
 int jam = Integer.*parseInt*(input.substring(0,2));  
 if (jam == 0){  
 hasil = "12" + input.substring(2,8) + " AM";  
 } else if (jam == 12) {  
 hasil = input.substring(0,8) + " PM";  
 } else if (jam > 0 && jam < 12){  
 hasil = input.substring(0,8) + " AM";  
 } else {  
 jam -= 12;  
 hasil = String.*format*("%02d",jam) + input.substring(2,8) + " PM";  
 }  
 }  
 System.*out*.println(hasil);  
 }  
}

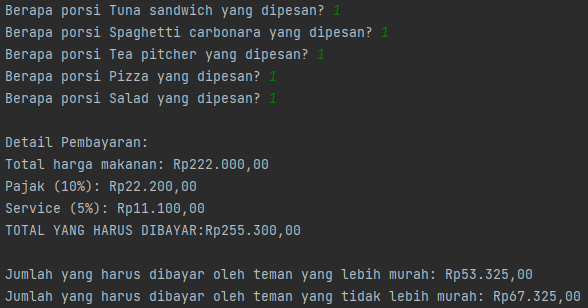
1. Kamu dan 3 temanmu makan-makan di restoran dan kalian memesan beberapa menu yang nanti pembayarannya akan dibagi rata. Namun ada 1 orang yang alergi ikan, sehingga disepakati untuk 1 jenis makanan yang mengandung ikan itu hanya 3 orang yang akan membayar. Pajak 10% dari harga makanan dan service 5% dari harga makanan. Tentukan berapa yang harus dibayar oleh masing-masing temanmu (ingat ya ada 1 orang yang membayar lebih murah karena alergi).

Contoh:

Makanan yang dipesan:

1. Tuna sandwich 42K
2. Spaghetti carbonara 50K
3. Tea pitcher 30K
4. Pizza 70K
5. Salad 30K
6. ...

**Jawab:**

****

**Script:**

package BLQ;  
  
import java.text.NumberFormat;  
import java.util.Locale;  
import java.util.Scanner;  
  
public class Soal16 {  
 public static void main(String[] args) {  
 // Daftar makanan dan harganya  
 String[] menu = {"Tuna sandwich", "Spaghetti carbonara", "Tea pitcher", "Pizza", "Salad"};  
 int[] harga = {42000, 50000, 30000, 70000, 30000};  
  
 // Inisialisasi total harga makanan  
 int totalHargaMakanan = 0;  
  
 // Inisialisasi harga tuna  
 double hargaTuna = 0;  
  
 // Membuat objek Scanner untuk mendapatkan input dari pengguna  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Menghitung total harga makanan  
 for (int i = 0; i < harga.length; i++) {  
 System.*out*.print("Berapa porsi " + menu[i] + " yang dipesan? ");  
 int porsi = scanner.nextInt();  
 totalHargaMakanan += (harga[i] \* porsi);  
  
 // Menghitung harga tuna khusus  
 if (menu[i].equals("Tuna sandwich")) {  
 hargaTuna = harga[i] \* porsi;  
 }  
 }  
  
 // Menghitung pajak dan service  
 double pajak = 0.1 \* totalHargaMakanan;  
 double service = 0.05 \* totalHargaMakanan;  
 double totalYangHarusDibayar = pajak + service + totalHargaMakanan;  
  
 // Menampilkan total harga makanan, pajak, dan service dengan format rupiah  
 NumberFormat formatter = NumberFormat.*getCurrencyInstance*(new Locale("id", "ID"));  
 System.*out*.println("\nDetail Pembayaran:");  
 System.*out*.println("Total harga makanan: " + formatter.format(totalHargaMakanan));  
 System.*out*.println("Pajak (10%): " + formatter.format(pajak));  
 System.*out*.println("Service (5%): " + formatter.format(service));  
 System.*out*.println("TOTAL YANG HARUS DIBAYAR:" + formatter.format(totalYangHarusDibayar) + "\n");  
  
 // Menampilkan jumlah yang harus dibayar oleh teman yang lebih murah dan yang tidak lebih murah  
 double pembayaranTemanMurah = (totalYangHarusDibayar - hargaTuna) / 4;  
 double pembayaranTemanBiasa = (totalYangHarusDibayar - pembayaranTemanMurah) / 3;  
  
 System.*out*.println("Jumlah yang harus dibayar oleh teman yang lebih murah: " + formatter.format(pembayaranTemanMurah));  
 System.*out*.println("Jumlah yang harus dibayar oleh teman yang tidak lebih murah: " + formatter.format(pembayaranTemanBiasa));  
  
 // Menutup objek Scanner  
 scanner.close();  
 }  
}

1. Hattori sedang berlatih untuk menjadi ninja yang baik dengan berlari melewati gunung dan lembah. Yang didefinisikan sebagai gunung dan lembah adalah:

* Gunung: urutan Naik dan Turun yang bermula di 0 mdpl dan berakhir di 0 mdpl
* Lembah: urutan Turun dan Naik yang bermula di 0 mdpl dan berakhir di 0 mdpl

Hattori mencatat perjalanannya dengan simbol **N** saat ia menanjak dan **T** saat ia turun dalam sebuah urutan seperti berikut.

**N N T N N N T T T T T N T T T N T N**

Berapa Gunung dan Lembah yang sudah dilewati Hattori?

**Jawab:**

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal17 {  
 public static void main(String[] args) {  
 //N N T N N N T T T T T N T T T N T N  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("jalur : ");  
 String[] arrJalur = scanner.nextLine().trim().toLowerCase().split(" ");  
  
 int jmlGunung = 0;  
 int jmlLembah = 0;  
 int mdpl = 0;  
  
 for (int i = 0; i < arrJalur.length; i++) {  
 String jalurSekarang = arrJalur[i];  
 if (jalurSekarang.equals("n")){  
 mdpl++;  
 if (mdpl == 0){  
 jmlLembah++;  
 }  
 } else if(jalurSekarang.equals("t")){  
 mdpl--;  
 if (mdpl == 0){  
 jmlGunung++;  
 }  
 }  
 }  
  
 System.*out*.println("Gunung : " + jmlGunung + ", Lembah : " + jmlLembah);  
 }  
}

1. Donna sangat menyukai kue, tetapi ia ingin badannya tetap bagus sehingga setiap Donna memakan kue dengan **n** kalori, ia perlu berolahraga selama **0.1 x n x j** menit, dengan **j** adalah selisih waktu dari ia memakan kue dan berolahraga (dalam satuan jam). Agar tetap terhidrasi, Donna akan meminum 100 cc air setiap 30 menit berolahraga dan 500 cc di akhir olahraga. Berikut ini adalah jam makan dan kalori dari kue yang dimakan Donna hari ini.

|  |  |
| --- | --- |
| Jam | Kalori |
| 9 | 30 |
| 13 | 20 |
| 15 | 50 |
| 17 | 80 |

Jika Donna mulai olahraga jam 18, berapa cc air yang akan diminum Donna sepanjang berolahraga?

**Jawab:**

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal18 {  
 public static void main(String[] args) {  
 // Jam makan dan kalori dari kue  
 int[] jamMakan = {9, 13, 15, 17};  
 int[] kaloriKue = {30, 20, 50, 80};  
  
 // Membuat objek Scanner untuk mendapatkan input dari pengguna  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Input waktu mulai olahraga Donna  
 System.*out*.print("Jam mulai olahraga Donna (format 24 jam): ");  
 int waktuMulaiOlahraga = scanner.nextInt();  
  
 // Menutup objek Scanner  
 scanner.close();  
  
 // Menghitung total waktu olahraga dan waktu setiap aktivitas makan  
 double totalWaktuOlahraga = 0;  
  
 // Menghitung waktu dan kalori dari makanan yang dimakan  
 for (int i = 0; i < jamMakan.length; i++) {  
 // Selisih waktu dari memakan kue dan waktu olahraga (dalam jam)  
 double selisihWaktu = waktuMulaiOlahraga - jamMakan[i];  
  
 // Menghitung waktu olahraga untuk setiap kue yang dimakan  
 double waktuOlahragaPerKue = 0.1 \* kaloriKue[i] \* selisihWaktu;  
  
 // Menambahkan waktu olahraga per kue ke total waktu olahraga  
 totalWaktuOlahraga += waktuOlahragaPerKue;  
 }  
  
 // Menambahkan waktu olahraga untuk minum air setiap 30 menit  
 totalWaktuOlahraga += *hitungWaktuMinumAir*(30, waktuMulaiOlahraga);  
  
 // Menambahkan waktu olahraga untuk minum air di akhir olahraga  
 totalWaktuOlahraga += *hitungWaktuMinumAir*(0, waktuMulaiOlahraga);  
  
 // Menghitung jumlah air yang diminum Donna  
 int jumlahAir = *hitungJumlahAir*(totalWaktuOlahraga);  
  
 System.*out*.println("Donna akan meminum " + jumlahAir + " cc air selama berolahraga.");  
 }  
  
 // Fungsi untuk menghitung waktu minum air  
 private static double hitungWaktuMinumAir(int waktuMinumAir, int waktuMulaiOlahraga) {  
 return waktuMinumAir / 60.0; // Mengubah waktu dari menit ke jam  
 }  
  
 // Fungsi untuk menghitung jumlah air yang diminum Donna  
 private static int hitungJumlahAir(double totalWaktuOlahraga) {  
 // Donna akan meminum 100 cc air setiap 30 menit  
 int jumlahAirPer30Menit = 100;  
 int intervalMinumAir = 30; // dalam menit  
  
 // Menghitung jumlah air yang diminum Donna setiap 30 menit  
 int jumlahAir = (int) (totalWaktuOlahraga / intervalMinumAir) \* jumlahAirPer30Menit;  
  
 // Donna juga meminum 500 cc di akhir olahraga  
 jumlahAir += 500;  
  
 return jumlahAir;  
 }  
}

1. Tentukan apakah kalimat ini adalah Pangram\* atau bukan

“Sphinx of black quartz, judge my vow”

“Brawny gods just flocked up to quiz and vex him”

“Check back tomorrow; I will see if the book has arrived.”

***\*Pangram adalah kata atau kalimat yang mengandung setiap abjad alphabet, contohnya “A quick brown fox jumps over the lazy dog”***

**Jawab:**

****

****

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal19 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
   
 //Sphinx of black quartz, judge my vow  
 //Brawny gods just flocked up to quiz and vex him  
 //Check back tomorrow; I will see if the book has arrived  
  
 System.*out*.print("Kalimat: ");  
 String str= in.nextLine();  
  
 int count[] = new int[26];  
 boolean flag = true;  
 char ch;  
  
 for(int i=0; i<str.length(); i++){  
 ch = str.charAt(i);  
  
 if(ch == ' ')  
 continue;  
  
 if(ch>='A' && ch<='Z'){  
 count[ch-'A']++;  
 } else if(ch>='a' && ch<='z'){  
 count[ch-'a']++;  
 }  
 }  
  
 for(int i=0; i<count.length; i++){  
 if(count[i] == 0){  
 flag = false;  
 break;  
 }  
 }  
  
 if(flag)  
 System.*out*.println("Pangram");  
 else  
 System.*out*.println("Bukan Pangram");  
  
 }  
}

1. A dan B bermain suit Gunting Batu Kertas. Dalam suit tersebut, jika menang maka maju 2 langkah dan jika kalah maka mundur satu langkah. Jika sudah tidak ada jarak antara 2 pemain, maka pemain yang terakhir menanglah yang dinyatakan sebagai pemenang.

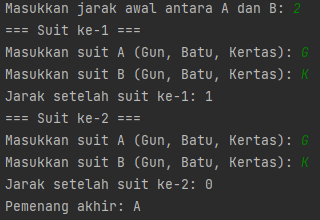
Contohnya:

* Jarak awal : 2
* A : G G G
* B : K K B

Jika posisi awal A dan B terpisah 2 langkah, maka setelah suit B mundur 1 langkah dan A maju 2 langkah, sehingga posisi mereka sekarang terpisah 1 langkah. Jika kemudian A menang lagi, maka B mundur 1 langkah dan A maju 2 langkah, sehingga tidak ada jarak lagi antara A dan B. Pemenang ditentukan dari pertama kalinya jarak antara A dan B menjadi nol, jadi meskipun pada suit ke-3 A kalah, putaran ke-3 ini diabaikan.

Dari kombinasi jarak awal dan suit yang dilakukan A dan B, tentukan siapakah pemenang game suit tersebut. Jika sampai akhir suit jarak antara mereka belum sama dengan nol, maka hasilnya dinyatakan “Draw”.

**Jawab:**

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal20 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Masukkan jarak awal antara A dan B: ");  
 int jarakAwal = scanner.nextInt();  
  
 int jarak = jarakAwal;  
 String pemenangTerakhir = "";  
  
 for (int i = 1; i <= 3 && jarak != 0; i++) {  
 System.*out*.println("=== Suit ke-" + i + " ===");  
  
 System.*out*.print("Masukkan suit A (Gun, Batu, Kertas): ");  
 String suitA = scanner.next();  
  
 System.*out*.print("Masukkan suit B (Gun, Batu, Kertas): ");  
 String suitB = scanner.next();  
  
 char suitAChar = suitA.charAt(0);  
 char suitBChar = suitB.charAt(0);  
  
 if (suitAChar == 'B' && suitBChar == 'G' || suitAChar == 'K' && suitBChar == 'B'  
 || suitAChar == 'G' && suitBChar == 'K') {  
 // A menang, maju 2 langkah dan B mundur 1 langkah  
 jarak -= 1;  
 pemenangTerakhir = "A";  
 } else if (suitBChar == 'B' && suitAChar == 'G' || suitBChar == 'K' && suitAChar == 'B'  
 || suitBChar == 'G' && suitAChar == 'K') {  
 // B menang, mundur 1 langkah dan A maju 2 langkah  
 jarak -= 1;  
 pemenangTerakhir = "B";  
 }  
  
 System.*out*.println("Jarak setelah suit ke-" + i + ": " + jarak);  
 }  
  
 if (jarak == 0) {  
 System.*out*.println("Pemenang akhir: " + pemenangTerakhir);  
 } else {  
 System.*out*.println("Pemenang akhir: Seri");  
 }  
  
 scanner.close();  
 }  
}

1. **Beware the Manhole**. Dalam game ini, player berjalan atau melompat sepanjang lintasan tertentu. Pada lintasan tersebut terdapat lubang yang hanya bisa dilewati dengan cara dilompati. Stat untuk berjalan dan melompat adalah sebagai berikut (stamina/ST dan distance/D):

* Walk : ST +1, D +1
* Jump : ST -2, D +3

Stamina awal selalu adalah 0.

Tentukan urutan Walk & Jump minimal yang diperlukan player untuk sampai tepat ke ujung lintasan. Jika tidak ada kombinasi yang memungkinkan (misalnya ST player 0 tepat sebelum melewati lubang), maka nyatakan hasilnya sebagai “FAILED”.

Contoh:

☺\_ \_ \_ \_ \_ O \_ \_ \_ Finish

**Jawaban**: W W W W J J

**Penjelasan**: player berjalan dulu untuk mengisi ST. Setelah 2 Walk, player memiliki ST yang cukup untuk melakukan 1 kali Jump, tapi jika ini dilakukan maka player tidak akan punya cukup ST untuk lompat melewati lubang. Begitu pula jika player melompat dari posisi ke-3, maka player akan berakhir masuk lubang. Karenanya, player baru bisa melompat dengan aman setelah posisi ke-4 dan akan mendarat tepat setelah lubang. Setelahnya, player masih memiliki cukup ST untuk melompat tepat ke Finish sehingga langkah terakhirnya adalah Jump.

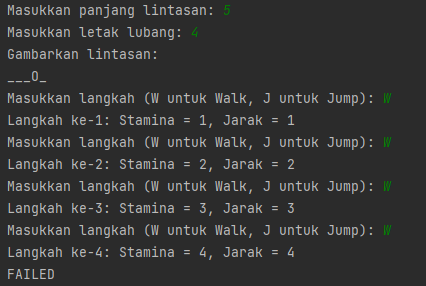
Contoh 2:

☺ O \_ \_ \_ \_ O \_ \_ \_ Finish

FAILED

**Penjelasan**: Player tidak memiliki cukup ST untuk melompat melewati lubang.

**Jawab:**

****

**Script:**

package BLQ;  
  
import java.util.Scanner;  
  
public class Soal21 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Masukkan panjang lintasan: ");  
 int lintasanPanjang = scanner.nextInt();  
  
 // Menentukan letak lubang (misalnya, di tengah lintasan)  
 System.*out*.print("Masukkan letak lubang: ");  
 int letakLubang = scanner.nextInt();  
  
 System.*out*.println("Gambarkan lintasan:");  
 *printLintasan*(lintasanPanjang, letakLubang);  
  
 int st = 0; // Stamina awal  
 int d = 0; // Distance awal  
  
 // Menyimpan urutan langkah  
 int[] steps = new int[lintasanPanjang]; // Ukuran disesuaikan dengan panjang lintasan  
  
 // Iterasi sampai mencapai ujung lintasan atau kehabisan stamina atau masuk lubang  
 int i = 0;  
 while (d < lintasanPanjang && st >= 0) {  
 System.*out*.print("Masukkan langkah (W untuk Walk, J untuk Jump): ");  
 char langkah = scanner.next().toLowerCase().charAt(0); // Ubah ke huruf kecil  
  
 // Cobalah untuk melompat jika langkah adalah 'j' dan stamina mencukupi  
 if (langkah == 'j') {  
 st -= 2;  
 d += 3;  
 steps[i++] = 1; // 1 menandakan langkah Jump  
 } else if (langkah == 'w'){  
 // Jika tidak melompat atau tidak cukup stamina, berjalan saja  
 st += 1;  
 d += 1;  
 steps[i++] = 0; // 0 menandakan langkah Walk  
 }  
  
 // Cetak nilai stamina dan jarak setiap langkah  
 System.*out*.println("Langkah ke-" + i + ": Stamina = " + st + ", Jarak = " + d);  
  
 // Cek apakah stamina habis atau mencapai ujung lintasan  
 if (st == 0 || d >= lintasanPanjang || d == letakLubang) {  
 break; // Keluar dari loop jika permainan selesai  
 }  
 }  
  
 // Jika berhasil mencapai ujung lintasan  
 if (d >= lintasanPanjang) {  
 System.*out*.println("Selamat, Anda berhasil mencapai ujung lintasan!");  
 *printResult*(steps);  
 } else {  
 // Jika gagal mencapai ujung lintasan  
 System.*out*.println("FAILED");  
 }  
 }  
  
 public static void printLintasan(int lintasanPanjang, int letakLubang) {  
 for (int i = 0; i < lintasanPanjang; i++) {  
 if (i + 1 == letakLubang) {  
 System.*out*.print("O");  
 } else {  
 System.*out*.print("\_");  
 }  
 }  
 System.*out*.println();  
 }  
  
 public static void printResult(int[] result) {  
 System.*out*.println("Langkah yang diperlukan untuk mencapai ujung lintasan:");  
 for (int step : result) {  
 System.*out*.print((step == 0 ? "Walk" : "Jump") + " ");  
 }  
 }  
}

1. Sederet lilin memiliki perbandingan laju meleleh mengikuti deret Fibonacci.

Diketahui deret Fibonacci adalah sebagai berikut:

1 1 2 3 5 8 13

Sehingga dapat dikatakan

* Lilin 1 dan 2 meleleh sepanjang 1 per detik
* Lilin ke-3 meleleh sepanjang 2 per detik
* Lilin ke-6 meleleh sepanjang 8 per detik
* Dan seterusnya…

Jika diberikan panjang lilin awal masing-masing adalah

3 3 9 6 7 8 23

Tentukan lilin mana yang paling pertama habis meleleh.

**Jawab:**

****

**Script:**

package BLQ;  
  
public class Soal22 {  
 public static void main(String[] args) {  
 int[] panjangLilin = {3, 3, 9, 6, 7, 8, 23};  
  
 int lilinPertamaHabisMeleleh = *lilinPertamaHabisMeleleh*(panjangLilin);  
  
 System.*out*.println("Lilin pertama yang habis meleleh: Lilin " + (lilinPertamaHabisMeleleh + 1));  
 }  
  
 public static int lilinPertamaHabisMeleleh(int[] panjangLilin) {  
 int n = panjangLilin.length;  
 int[] fibonacci = new int[n];  
  
 // Menghitung deret Fibonacci  
 fibonacci[0] = 1;  
 fibonacci[1] = 1;  
 for (int i = 2; i < n; i++) {  
 fibonacci[i] = fibonacci[i - 1] + fibonacci[i - 2];  
 }  
  
 // Mencari lilin pertama yang habis meleleh  
 int i = 0;  
 while (i < n && panjangLilin[i] >= fibonacci[i]) {  
 panjangLilin[i] -= fibonacci[i];  
 i++;  
 }  
  
 return i;  
 }  
}