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 | | | |
| Kaca\_mata | 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** (kaca\_mata 500, baju 200, sepatu 200, buku 100)

1. Di perpustakaan, anda bias 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***

***Tanggal Peminjaman: 28 February 2016-07 March 2016***

***DendaBuku B:500***

***DendaBuku C:100***

***DendaBuku D:100***

***Total Denda:700***

***Tanggal Peminjaman: 29 April 2018-30 May 2018***

***DendaBuku A:1700***

***DendaBuku B:2800***

***DendaBuku C:2400***

***DendaBuku D:2400***

***Total Denda:9300***

**public** **class** Soal2 {

**static** **int** *dendabuku* = 100;

**static** **int** *dendatotal*=0;

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

// **TODO** Auto-generated method stub

**int** hari\_peminjaman = 8;

*methodPerhitungandenda*("28 February 2016","07 March 2016");

*methodPerhitungandenda*("29 April 2018","30 May 2018");

}

//procedute hitung semua denda

**public** **static** **void** methodPerhitungandenda(String tanggal\_awal,String tanggal\_akhir) { //method procedure

System.***out***.println("Tanggal Peminjaman: "+tanggal\_awal+"-"+tanggal\_akhir);

DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("dd MMMM yyyy");

DateTimeFormatter formatter2 = DateTimeFormatter.*ofPattern*("dd MMMM yyyy");

LocalDate start = LocalDate.*parse*(tanggal\_awal, formatter);

LocalDate end = LocalDate.*parse*(tanggal\_akhir, formatter);

**long** lama\_peminjaman = start.until(end, ChronoUnit.***DAYS***);;

**int** batas\_hari\_peminjaman\_buku\_a=14;

**int** batas\_hari\_peminjaman\_buku\_b = 3;

**int** batas\_hari\_peminjaman\_buku\_c =7;

**int** batas\_hari\_peminjaman\_buku\_d =7;

**int** hari\_denda\_buku\_a= (**int**) (lama\_peminjaman-batas\_hari\_peminjaman\_buku\_a);

**int** hari\_denda\_buku\_b= (**int**) (lama\_peminjaman-batas\_hari\_peminjaman\_buku\_b);

**int** hari\_denda\_buku\_c= (**int**) (lama\_peminjaman-batas\_hari\_peminjaman\_buku\_c);

**int** hari\_denda\_buku\_d= (**int**) (lama\_peminjaman-batas\_hari\_peminjaman\_buku\_d);

**if**(hari\_denda\_buku\_a>0) {

*dendatotal*=*hitungDendaBuku*(hari\_denda\_buku\_a,"Buku A");

}

**if**(hari\_denda\_buku\_b>0) {

*dendatotal* +=*hitungDendaBuku*(hari\_denda\_buku\_b,"Buku B");

}

**if**(hari\_denda\_buku\_c>0) {

*dendatotal* += *hitungDendaBuku*(hari\_denda\_buku\_c,"Buku C");

}

**if**(hari\_denda\_buku\_d>0) {

*dendatotal* += *hitungDendaBuku*(hari\_denda\_buku\_d,"Buku D");

}

System.***out***.println("Total Denda:"+*dendatotal*);

System.***out***.println();

}

**public** **static** **int** hitungDendaBuku(**int** hari\_peminjaman,String nama\_buku) {

**int** bayar\_denda\_buku = hari\_peminjaman\* *dendabuku*;

System.***out***.println("Denda"+ nama\_buku +":"+ bayar\_denda\_buku);

**return** bayar\_denda\_buku;

}

}

1. Buatlah fungsi untuk kalkulasi tarif parkir berikut

|  |  |  |
| --- | --- | --- |
| Masuk | Keluar | Tarif |
| 27 Januari 2019 | 05:00:01 | 27 Januari 2019 | 17:45:03 | 8000 |
| 27 Januari 2019 | 07:03:59 | 27 Januari 2019 | 15:30:06 | 8000 |
| 27 Januari 2019 | 07:05:00 | 28 Januari 2019 | 00:20:21 | 8000 |
| 27 Januari 2019 | 11:14:23 | 27 Januari 2019 | 13:20:00 | 2000 |

Ketentuan tariff:

* 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

**import** java.text.SimpleDateFormat;

**import** java.time.Instant;

**import** java.time.LocalDateTime;

**import** java.time.ZoneId;

**import** java.time.format.DateTimeFormatter;

**import** java.time.temporal.ChronoUnit;

**public** **class** Soal3 {

**public** **static** **void** Calculate\_biaya\_parkir(String tanggal\_masuk,String tanggal\_keluar) {

System.***out***.println("Tanggal Masuk: "+tanggal\_masuk);

System.***out***.println("Tanggal Keluar: "+tanggal\_keluar);

String[] arrayMasuk = tanggal\_masuk.split(" ");

String[] arrayKeluar = tanggal\_keluar.split(" ");

String[] Bulan = {"Januari","Februari","Maret","April",

"Mei","Juni","Juli","Agustus","September","Oktober","November","Desember"};

String formatBulan = "";

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

**if** (Bulan[i].equals(arrayMasuk[1])){

formatBulan = "0"+(i+1);

**break**;

}

}

String startDateString = arrayMasuk[2] +"-"+formatBulan+"-"+arrayMasuk[0]+" "+arrayMasuk[4];

String endDateString = arrayKeluar[2] +"-"+formatBulan+"-"+arrayKeluar[0]+" "+arrayKeluar[4];

DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("yyyy-MM-dd HH:mm:ss");

Instant startInstant = LocalDateTime.*parse*(startDateString, formatter).atZone(ZoneId.*systemDefault*()).toInstant();

Instant endInstant = LocalDateTime.*parse*(endDateString, formatter).atZone(ZoneId.*systemDefault*()).toInstant();

**long** diffHours = ChronoUnit.***HOURS***.between(startInstant, endInstant);

**long** diffminutes = ChronoUnit.***MINUTES***.between(startInstant, endInstant);

**long** sisaMenit = diffminutes % 60;

**long** diffminutesinSec = ChronoUnit.***SECONDS***.between(startInstant,endInstant);

**long** sisaSec = diffminutesinSec % 60 ;

**if** (diffHours >= 24 && sisaMenit>=0 && sisaSec > 0){

System.***out***.println("Biaya Parkir:Rp."+(15000 + (diffHours - 24) \* 1000));

}**else** **if**(diffHours==24 && sisaMenit==0 && sisaSec == 0) {

System.***out***.println("Biaya Parkir:Rp."+8000);

}

**else** **if** (diffHours >= 24 && sisaMenit>=0 && sisaSec == 0){

System.***out***.println("Biaya Parkir:Rp."+(15000 + (diffHours - 24) \* 1000));

}

**else** **if**(diffHours >=8 && diffHours<24 && sisaMenit>=0 && (sisaSec==0||sisaSec > 0)){

System.***out***.println("Biaya Parkir:Rp."+8000);

}

**else** **if**(diffHours<8){

System.***out***.println("Biaya Parkir:Rp."+(diffHours \* 1000));

}

System.***out***.println();

}

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

*Calculate\_biaya\_parkir*("27 Januari 2019 | 05:00:01","27 Januari 2019 | 17:45:03");

*Calculate\_biaya\_parkir*("27 Januari 2019 | 07:03:59","27 Januari 2019 | 15:30:06");

*Calculate\_biaya\_parkir*("27 Januari 2019 | 07:05:00","28 Januari 2019 | 00:20:21");

*Calculate\_biaya\_parkir*("27 Januari 2019 | 11:14:23","27 Januari 2019 | 13:20:00");

}

}

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

Jawaban:

**package** assignment\_blq;

**import** java.util.Scanner;

**public** **class** Soal4 {

**public** **static** **void** first\_n\_PrimeNumber(**int** n) {

**int** status = 1;

**int** num = 3;

**if** (n >= 1) {

System.***out***.println(n + " Bilangan Prima Pertama:");

System.***out***.println(2);

}

**for** (**int** i = 2; i <= n;) {

**for** (**int** j = 2; j <= Math.*sqrt*(num); j++) {

**if** (num % j == 0) {

status = 0;

**break**;

}

}

**if** (status != 0) {

System.***out***.println(num);

i++;

}

status = 1;

num++;

}

System.***out***.println();

}

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

*first\_n\_PrimeNumber*(5);

*first\_n\_PrimeNumber*(10);

}

}

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

**public** **class** Soal5 {

**public** **static** **void** first\_n\_fibonaciNumber(**int** n){

**int** firstTerm = 0, secondTerm = 1;

System.***out***.println(n + " Bilangan Fibonaci Pertama:");

**for** (**int** i = 0; i < n; ++i) {

**if**(i==n-1) {

System.***out***.print(firstTerm);

}**else** {

System.***out***.print(firstTerm + ", ");

}

// compute the next term

**int** nextTerm = firstTerm + secondTerm;

firstTerm = secondTerm;

secondTerm = nextTerm;

}

System.***out***.println();

}

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

*first\_n\_fibonaciNumber*(5);

*first\_n\_fibonaciNumber*(10);

}

}

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”***

**public** **class** Soal6 {

**public** **static** String palindromeCheck(String kata){

Boolean kata\_flag= **true**;

System.***out***.println("Input Kata: "+ kata);

String[] pisahKata = kata.split("");

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

System.***out***.printf("%2s",pisahKata[i]);

}

System.***out***.println();

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

{

**if**(!pisahKata[i].toLowerCase().equals(pisahKata[(pisahKata.length-1)-i].toLowerCase()))

{

//System.out.println("Input 1 bukan palindrome");

kata\_flag=**false**;

**break**;

}

}

**if**(kata\_flag) {

**return** "Input Kata palindrome";

}**else** {

**return** "Input Kata bukan palindrome";

}

}

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

// **TODO** Auto-generated method stub

System.***out***.println(*palindromeCheck*("katak"));

System.***out***.println(*palindromeCheck*("12021"));

System.***out***.println(*palindromeCheck*("malam"));

}

}

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**

**Input Array: [8, 7, 0, 2, 7, 1, 7, 6, 3, 0, 7, 1, 3, 4, 6, 1, 6, 4, 3]**

**Mean: 4.00**

**Median: 4.00**

**Mode: 7.00**

**import** java.io.\*;

**import** java.util.\*;

**public** **class** Soal7 {

**public** **static** **double** getMean(**int**[] arr) {

**double** mean = 0;

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

mean += arr[i];

mean /= arr.length;

**return** mean;

}

**public** **static** **double** getMedian(**int**[] arr) {

**int**[] arrCopy = arr.clone();

Arrays.*sort*(arrCopy);

**double** median = 0;

**int** n = arrCopy.length;

**if** (n % 2 == 1) {

median = arrCopy[n / 2];

} **else** {

median = (arrCopy[n / 2] + arrCopy[n / 2 - 1]) / 2.0;

}

**return** median;

}

**public** **static** **double** getMode(**int**[] arr) {

**int**[] arrCopy = arr.clone();

Arrays.*sort*(arrCopy);

Map<Integer, Integer> freq = **new** TreeMap<>();

**for** (**int** i : arrCopy) {

**if** (!freq.containsKey(i))

freq.put(i, 1);

**else**

freq.put(i, freq.get(i) + 1);

}

**double** mode = arrCopy[0];

**int** maxFreq = 1;

**for** (Map.Entry<Integer, Integer> me : freq.entrySet()) {

**if** (me.getValue() > maxFreq) {

maxFreq = me.getValue();

mode = me.getKey();

}

}

**return** mode;

}

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

**int**[] arr1= {8, 7, 0, 2, 7, 1, 7, 6, 3, 0, 7, 1, 3, 4, 6, 1, 6, 4, 3};

System.***out***.println("Input Array: "+ Arrays.*toString*(arr1));

**double** mean = *getMean*(arr1);

**double** median = *getMedian*(arr1);

**double** mode = *getMode*(arr1);

System.***out***.printf("Mean: %.2f\n", mean);

System.***out***.printf("Median: %.2f\n", median);

System.***out***.printf("Mode: %.2f\n", mode);

}

}

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

**1 2 4 7 8 6 9**

**Jawaban:**

**4 ANGKA PENJUMLAHAN MINIMUM ADALAH : 13**

**4 ANGKA PENJUMLAHAN MAXIMUM ADALAH : 30**

**import** java.util.Scanner;

**public** **class** Soal8 {

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

// **TODO** Auto-generated method stub

**int** sumterkecil = 0;

**int** sumterbesar = 0;

**int** [] arr\_angka = {1,2,4,7,8,6,9};

**int** juml\_angka\_tambah = 4;

// proses sorting dari terkecil hingga terbesar

**for** (**int** i = 0; i < arr\_angka.length; i++) {

**for** (**int** j = 0; j < arr\_angka.length; j++) {

**if** (arr\_angka[i] < arr\_angka[j]) {

**int** urut = arr\_angka[i];

arr\_angka[i] = arr\_angka[j];

arr\_angka[j] = urut;

}

}

}

**for** (**int** i = 0; i < juml\_angka\_tambah; i++) {

sumterkecil = sumterkecil + arr\_angka[i];

}

// proses sorting dari terbesar hingga terkecil

**for** (**int** i = 0; i < arr\_angka.length; i++) {

**for** (**int** j = 0; j < arr\_angka.length; j++) {

**if** (arr\_angka[i] > arr\_angka[j]) {

**int** urut = arr\_angka[i];

arr\_angka[i] = arr\_angka[j];

arr\_angka[j] = urut;

}

}

}

**for** (**int** i = 0; i < juml\_angka\_tambah; i++) {

sumterbesar = sumterbesar + arr\_angka[i];

}

System.***out***.printf("4 ANGKA PENJUMLAHAN TERKECIL ADALAH : "+sumterkecil);

System.***out***.println();System.***out***.println();

System.***out***.printf("4 ANGKA PENJUMLAHAN TERBESAR ADALAH : "+sumterbesar);

}

}

1. N = 3 🡪 3 6 9

N = 4 🡪 4 8 12 16

N = 5 🡪 5 10 15 20 25

**package** assignment\_blq;

**public** **class** Soal9 {

**public** **static** **void** deret\_penjumlahan(**int** n){

**int** total=n;

System.***out***.print("N = "+n+" -> ");

**for** (**int** i = 0; i < n; ++i) {

**if**(i==n-1) {

System.***out***.print(total);

}**else** {

System.***out***.print(total + " ");

}

total=total+n;

}

System.***out***.println();

}

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

*deret\_penjumlahan*(3);

*deret\_penjumlahan*(4);

*deret\_penjumlahan*(5);

}

}

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



**public** **class** Soal10 {

**public** **static** **void** maskingSentence(String sentence) {

String[] subStrings = sentence.split(" ");

String maskString="";

**for**(String subString: subStrings) {

**if**(subString.length()<2) {

maskString=maskString+subString+" ";

}**else** {

maskString=maskString+subString.substring(0,1)+"\*\*\*"+subString.substring(subString.length()-1)+" ";

}

}

System.***out***.println(maskString.trim());

}

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

*maskingSentence*("Susilo Bambang Yudhoyono");

*maskingSentence*("Rani Tiara");

}

}

1. Input: Afrika

Output:

\*\*\*a\*\*\*

\*\*\*k\*\*\*

\*\*\*i\*\*\*

\*\*\*r\*\*\*

\*\*\*f\*\*\*

\*\*\*A\*\*\*

Input: Jeruk

Output:

\*\*k\*\*

\*\*u\*\*

\*\*r\*\*

\*\*e\*\*

\*\*J\*\*

**import** java.util.Scanner;

**public** **class** Soal11 {

**public** **static** **void** cetakKataBintang(String kata) {

System.***out***.println("Input Kata: "+kata);

String[] pisahKata1 = kata.split("");

//jumlah bintang (\*) yang dicetak

**int** bintang = pisahKata1.length/2;

**for** (**int** i = pisahKata1.length-1; i >=0; i--)

{

**for**(**int** j =0 ; j<bintang; j++)

{

System.***out***.print("\*");

}

System.***out***.print(pisahKata1[i]);

**for**(**int** j =pisahKata1.length-1+bintang ; j>pisahKata1.length-1; j--)

{

System.***out***.print("\*");

}

System.***out***.println();

}

System.***out***.println();

}

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

// **TODO** Auto-generated method stub

*cetakKataBintang*("Afrika");

*cetakKataBintang*("Jeruk");

}

}

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***

Input Array: [1, 2, 1, 3, 4, 7, 1, 1, 5, 6, 1, 8]

Array elements after sorting: 1 1 1 1 1 2 3 4 5 6 7 8

**import** java.util.Arrays;

**public** **class** Soal12 {

**public** **static** **void** arraySort(**int**[] deret){

// creating an instance of an array

System.***out***.println("Input Array: "+Arrays.*toString*(deret));

System.***out***.print("Array elements after sorting: ");

// sorting logic

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

**for** (**int** j = i + 1; j < deret.length; j++) {

**int** tmp = 0;

**if** (deret[i] > deret[j]) {

tmp = deret[i];

deret[i] = deret[j];

deret[j] = tmp;

}

}

// prints the sorted element of the array

**if**(i==deret.length-1) {

System.***out***.print(deret[i]);

}**else** {

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

}

}

System.***out***.println();

}

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

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

*arraySort*(arr);

}}

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***

**public** **class** Soal13 {

**static** **int** calcAngle(**double** h, **double** m) {

**if** (h < 0 || m < 0 || m > 60) {

System.***out***.println("Input Salah");

**return** -1;

}

**if** (h == 12)

h = 0;

**if** (m == 60) {

m = 0;

h += 1;

}

**if** (h > 12) {

h = h - 12;

}

**int** hour\_angle = (**int**) (0.5 \* (h \* 60 + m));

**int** minute\_angle = (**int**) (6 \* m);

**int** angle = Math.*abs*(hour\_angle - minute\_angle);

angle = Math.*min*(360 - angle, angle);

**return** angle;

}

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

System.***out***.println("Jam 3:00 -> " + *calcAngle*(3, 0) + " derajat");

System.***out***.println("Jam 5:30 -> " + *calcAngle*(5, 30) + " derajat");

System.***out***.println("Jam 2:20 -> " + *calcAngle*(2, 20) + " derajat");

}

}

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

Jawaban:

N = 3 -> 3 9 0 7 1 2 5

proses shifting ke kiri:

9 0 7 1 2 5 3

0 7 1 2 5 3 9

7 1 2 5 3 9 0

N = 1 -> 7 1 2 5 3 9 0

proses shifting ke kiri:

1 2 5 3 9 0 7

**import** java.util.Scanner;

**public** **class** Soal14 {

**public** **static** **void** geserKiriDeret(**int**[] deret,**int** n){

System.***out***.print("N = "+n+" ->");

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

System.***out***.printf("%2s",deret[i]);

}

System.***out***.println();

System.***out***.println("proses shifting ke kiri:");

// proses shifting ke kiri

**int** temp;

**for** (**int** i = 0; i < n; i++) {

temp=deret[0];

**for** (**int** j = 0; j < deret.length-1; j++) {

deret[j]=deret[j+1];

}

deret[deret.length-1]=temp;

//looping cetak angka

**for** (**int** k = 0; k < deret.length; k++) {

System.***out***.printf("%2s",deret[k]);

}

System.***out***.println();

}

System.***out***.println();

}

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

**int**[] deret1 ={3,9,0,7,1,2,5} ;

*geserKiriDeret*(deret1,3);

*geserKiriDeret*(deret1,1);

}

}

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

**public** **class** Soal15 {

**static** **void** convert\_12h\_to\_24h(String str)

{

**int** h1 = (**int**)str.charAt(1) - '0';

**int** h2 = (**int**)str.charAt(0) - '0';

**int** hh = (h2 \* 10 + h1 % 10);

**if** (str.charAt(8) == 'A')

{

**if** (hh == 12)

{

System.***out***.print("00");

**for** (**int** i = 2; i <= 7; i++)

System.***out***.print(str.charAt(i));

}

**else**

{

**for** (**int** i = 0; i <= 7; i++)

System.***out***.print(str.charAt(i));

}

}

**else**

{

**if** (hh == 12)

{

System.***out***.print("12");

**for** (**int** i = 2; i <= 7; i++)

System.***out***.print(str.charAt(i));

}

**else**

{

hh = hh + 12;

System.***out***.print(hh);

**for** (**int** i = 2; i <= 7; i++)

System.***out***.print(str.charAt(i));

}

}

}

**public** **static** **void** main (String[] args)

{

*convert\_12h\_to\_24h*("03:40:44 PM");

}

}

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 (mengandung ikan)

2. Spaghetti Carbonara 50K

3. Tea pitcher 30K

4. Pizza 70K

5. Salad 30K

6. …

**Tuna Sandwich (ikan):Rp.42000.0**

**Spaghetti Carbonara:Rp.50000.0**

**Tea pitcher:Rp.30000.0**

**Pizza:Rp.70000.0**

**Salad:Rp.30000.0**

**Total sebelum Pajak: Rp.222000.0**

**Total setelah Pajak: Rp.255300.0**

**Si A membayar:Rp.67850.0**

**Si B membayar:Rp.67850.0**

**Si C membayar:Rp.67850.0**

**Si D membayar:Rp.51750.0**

**public** **class** Soal16 {

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

// **TODO** Auto-generated method stub

**double** total=0;

**double**[] menu = **new** **double**[5];

menu[0]= 42000;//Tuna Sandwich (ikan)

System.***out***.println("Tuna Sandwich (ikan):Rp."+menu[0]);

menu[1]= 50000;// Spaghetti Carbonara

System.***out***.println("Spaghetti Carbonara:Rp."+menu[1]);

menu[2]= 30000;// Tea pitcher

System.***out***.println("Tea pitcher:Rp."+menu[2]);

menu[3]= 70000;// Pizza

System.***out***.println("Pizza:Rp."+menu[3]);

menu[4]= 30000;// Salad

System.***out***.println("Salad:Rp."+menu[4]);

**double** menu0\_total = (menu[0]\*1.15);

**double** menu1\_total = (menu[1]\*1.15);

**double** menu2\_total = (menu[2]\*1.15);

**double** menu3\_total = (menu[3]\*1.15);

**double** menu4\_total = (menu[4]\*1.15);

total= menu[0]+menu[1]+menu[2]+menu[3]+menu[4];

**double** totalincludetax= menu0\_total+menu1\_total+menu2\_total+menu3\_total+menu4\_total;

System.***out***.println("Total sebelum Pajak: Rp."+total);

System.***out***.println("Total setelah Pajak: Rp."+totalincludetax);

**double** jumlah\_a= (totalincludetax-menu0\_total)/4;

**double** jumlah\_alergi=Math.*ceil*(menu0\_total/3);

**double** a= jumlah\_a+jumlah\_alergi;

**double** b= jumlah\_a+jumlah\_alergi;

**double** c= jumlah\_a+jumlah\_alergi;

**double** d = jumlah\_a;

System.***out***.println("Si A membayar:Rp."+a);

System.***out***.println("Si B membayar:Rp."+b);

System.***out***.println("Si C membayar:Rp."+c);

System.***out***.println("Si D membayar:Rp."+d);

}

}

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?

Gunung: 1

Lembah: 1

**public** **class** Soal17 {

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

// **TODO** Auto-generated method stub

String arr= "N N T N N N T T T T T N T T T N T N";

String[] arrInput = arr.split(" ");

**int** mdpl=0;

**int** lembah=0;

**int** gunung=0;

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

**if**(arrInput[i].equalsIgnoreCase("N")) {

**if**(i==0) {

mdpl++;

}**else** {

**if**(mdpl+1==0) {

lembah++;

mdpl++;

}**else** {

mdpl++;

}

}

}**else** **if**(arrInput[i].equalsIgnoreCase("T")) {

**if**(i==0) {

mdpl--;

}**else** {

**if**(mdpl-1==0) {

gunung++;

mdpl--;

}**else** {

mdpl--;

}

}

}

}

System.***out***.println("Gunung: "+gunung);

System.***out***.println("Lembah: "+lembah);

}

}

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?

Kalori ke-1: 30.0

Jam ke-1: 9.0

Kalori ke-2: 20.0

Jam ke-2: 13.0

Kalori ke-3: 50.0

Jam ke-3: 15.0

Kalori ke-4: 80.0

Jam ke-4: 17.0

Jumlah Waktu Berolahraga: 60.0 menit

Jumlah Air yang diminum: 700 cc

**public** **class** Soal18 {

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

// **TODO** Auto-generated method stub

**double**[] kalori = **new** **double**[4];

kalori[0]=30;

kalori[1]=20;

kalori[2]=50;

kalori[3]=80;

**double**[] jam = **new** **double**[4];

jam[0]=9;

jam[1]=13;

jam[2]=15;

jam[3]=17;

**for**(**int** i=0;i<4;i++){

System.***out***.println("Kalori ke-"+(i+1)+": " +kalori[i]);

System.***out***.println("Jam ke-"+(i+1)+": " +jam[i]);

}

System.***out***.println();

**double** start\_olahraga=18;

**double** waktu\_olahraga=0;

**for**(**int** i=0;i<4;i++){

waktu\_olahraga=waktu\_olahraga+(0.1\*kalori[i]\*(start\_olahraga-jam[i]));

}

System.***out***.println("Jumlah Waktu Berolahraga: "+waktu\_olahraga+" menit");

**int** water=(**int**)((Math.*round*(waktu\_olahraga/30))\*100)+500 ;

System.***out***.println("Jumlah Air yang diminum: "+water+" cc");

}

}

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”***

Input Kalimat: Sphinx of black quartz, judge my vow

Kalimat ini adalah Pangram

Input Kalimat: Brawny gods just flocked up to quiz and vex him

Kalimat ini adalah Pangram

Input Kalimat: Check back tomorrow; I will see if the book has arrived.

Kalimat ini adalah Bukan Pangram

**public** **class** Soal19 {

**static** **int** *size* = 26;

// Function to check if ch is a letter

**static** **boolean** isLetter(**char** ch) {

**if** (!Character.*isLetter*(ch))

**return** **false**;

**return** **true**;

}

// Function to check if a string

// contains all the letters from

// a to z

**static** **void** allLetter(String str, **int** len) {

Boolean flag=**true**;

System.***out***.println("Input Kalimat: "+str);

str = str.toLowerCase();

**boolean**[] present = **new** **boolean**[*size*];

**for** (**int** i = 0; i < len; i++) {

**if** (*isLetter*(str.charAt(i))) {

// Mark current letter as present

**int** letter = str.charAt(i) - 'a';

present[letter] = **true**;

}

}

// Traverse for every letter

// from a to z

**for** (**int** i = 0; i < *size*; i++) {

**if** (!present[i]) {

flag=**false**;

**break**;

}

}

**if**(flag) {

System.***out***.println("Kalimat ini adalah Pangram");

}**else** {

System.***out***.println("Kalimat ini adalah Bukan Pangram");

}

}

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

// Given string str

String str1 = "Sphinx of black quartz, judge my vow";

String str2 = "Brawny gods just flocked up to quiz and vex him";

String str3 = "Check back tomorrow; I will see if the book has arrived.";

*allLetter*(str1, str1.length());

*allLetter*(str2, str2.length());

*allLetter*(str3, str3.length());

}

}

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”.

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.

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.

Jawaban no.22:

Masukkan jumlah lilin: 7

Masukkan input lilin: 3 3 9 6 7 8 23

2 2 7 3 2 0 10

Lilin Ke-6 Pertama Habis

**import** java.util.Scanner;

**public** **class** Soal22 {

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

**int** t1 = 1, t2 = 1;

Scanner s = **new** Scanner(System.***in***);

System.***out***.print("Masukkan jumlah lilin: ");

**int** n = s.nextInt();

System.***out***.print("Masukkan input lilin: ");

**int**[] lilin = **new** **int**[n];

**for** (**int** i = 0; i < n; i++) {

lilin[i]= s.nextInt();

}

**for** (**int** i = 0; i <n; i++){

**if**(i==0)

{

lilin[i] = lilin[i]-t1;

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

}

**else** **if**(i==1)

{

lilin[i] = lilin[i]-t2;

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

}

**else**

{

**int** sum = t1 + t2;

lilin[i]=lilin[i]-sum;

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

t1 = t2;

t2 = sum;

}

}

System.***out***.println();

**for** (**int** i = 0; i <n; i++){

**if**(lilin[i] <= 0)

{

System.***out***.println("Lilin Ke-"+(i+1) + " Pertama Habis");

**break**;

}

}

}

}

**PROJECT:**

**CRUD**

1. Perpustakaan
2. Rental Mobil
3. Rumah Makan
4. Penilaian Mahasiswa
5. Puskesmas
6. Apotek
7. Toko Bangunan
8. Reservasi Objek Wisata