**MULTIPLICACIÓN DE MATRICES**

**Nombre del archivo fuente:** MultiplicacionMatrices.C, MultiplicacionMatrices.CPP ó MultiplicacionMatrices.Java

Dos matrices A y B son multiplicables si el número de columnas de A coincide con el número de filas de B.

**Am x n x B n x p = C m x p**

El elemento Cij de la matriz producto se obtiene multiplicando cada elemento de la fila i de la matriz A por cada elemento de la columna j de la matriz B y sumándolos.



Hacer un programa que pida por pantalla las variables m, n y p. Con estos datos crear la matriz A de dimensiones m x n y la matriz B con dimensiones n x p. Los elementos de la matriz A llenarla con la siguiente formula: A[i][j] = (i + 1) \* j, Los elementos de la matriz B llenarla con la siguiente formula: B[i][j] = (j + 1) \* i. Luego mostrar la matriz resultante de la multiplicación de A x B.

**ENTRADA**

La entrada son tres líneas con un número entero cada una, que indican las dimensiones de la matriz A( número de la línea 1 \* número de la línea 2) y de la matriz B (número de la línea 2 \* número de la línea tres).

**Salida**

La salida es una línea que indica comienzo de la matriz A(\*\*\* A \*\*\*) y en la siguientes líneas mostrará los datos de la matriz. Igual para la matriz B, el mensaje(\*\*\* B \*\*\*) y en las siguientes líneas muestra la matriz B.

**Ejemplo**: al ejecutar el programa, queda esperando que el usuario ingrese los valores m, n y p, por ejemplo: 3 <Enter>, 4 <Enter> y 2 <Enter>. Luego debe mostrar la siguiente matriz:



**Nota**: Su programa debe tener una salida exacta a la que muestra el ejemplo, no coloque mensajes ni información adicional.

import java.util.Scanner;

public class MultiplicacionMatriz {

 public static void main(String[] args) {

 Scanner sc = new Scanner(System.in);

 int m = sc.nextInt();

 int n = sc.nextInt();

 int p = sc.nextInt();

 int a[][] = new int[m][n];

 int b[][] = new int[n][p];

 int c[][] = new int[m][p];

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

 for (int j = 0; j < n; j++) {

 a[i][j] = (i + 1) \* j;

 }

 }

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

 for (int j = 0; j < p; j++) {

 b[i][j] = (j + 1) \* i;

 }

 }

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

 for (int j = 0; j < p; j++) {

 for (int k = 0; k < n; k++) {

 c[i][j] += a[i][k] \* b[k][j];

 }

 }

 }

 System.out.println("\*\*\* A \*\*\*");

 show(a, m, n);

 System.out.println("\*\*\* B \*\*\*");

 show(b, n, p);

 System.out.println("\*\*\* C \*\*\*");

 show(c, m, p);

 }

 public static void show(int mat[][], int m, int n) {

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

 for (int j = 0; j < n; j++) {

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

 }

 System.out.println();

 }

 }

}

**UBICACIÓN DE ESTRELLAS**

**Nombre del archivo fuente:** UbicaEstrellas.java

Trabajas para los Laboratorios de Propulsión por Reacción Sputnik. En este momento es necesario que escribas un programa que lea una matriz, la cual contiene una representación digitalizada de una fotografía del cielo. Cada elemento de la matriz representa la cantidad de luz que existe en determinada región de la imagen digitalizada. El rango de intensidad va de 0 a 20. El programa permitirá localizar las regiones donde se ubica una estrella, partiendo de la siguiente información: Una estrella se encuentra en el área cubierta por el elemento i,j de la matriz si se cumple la siguiente condición:

(MD(i,j) + suma de intensidades circundantes)/5 > 10.0

Donde MD representa la matriz digitalizada.

**Entrada** La primera línea son dos enteros menores que 10 que indican el número de filas y columnas de la matriz, las siguientes líneas contienen enteros que representan las intensidades de cada posición de la matriz, separados por un blanco.

**Salida** La salida deseada es la matriz que contiene un asterisco en la posición donde está localizada una estrella, y un blanco donde no la hay. La matriz debe estar circundada por un borde que indique las coordenadas de cada estrella. La coordenada inicial es 1.

Cada elemento de la matriz debe estar separado por un espacio en blanco

**ENTRADA**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 8 |   |   |   |   |   |   |
| 0 | 3 | 4 | 20 | 15 | 0 | 6 | 8 |
| 5 | 13 | 6 | 8 | 2 | 0 | 2 | 3 |
| 2 | 6 | 2 | 2 | 3 | 0 | 10 | 0 |
| 0 | 0 | 4 | 15 | 4 | 1 | 1 | 20 |
| 0 | 0 | 7 | 2 | 6 | 9 | 10 | 4 |
| 5 | 0 | 6 | 10 | 6 | 4 | 8 | 0 |

**SALIDA**



import java.io.BufferedInputStream;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class UbicarEstrellas {

 static int n = 0;

 static int m = 0;

 static String cadena = "";

 //static Scanner leer = new Scanner(System.in);

 static BufferedReader leer = new BufferedReader(new InputStreamReader(System.in));

 public static void buscar(int n, int m) throws IOException {

 String ncadena = "";

 int tc = 0;

 int MD[][] = new int[n][m];

 int datos[] = new int[m];

 // System.out.println("Cadena :" + ncadena + "M" +m );

 //cadena= leer.next();

 // System.out.println();

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

 ncadena = leer.readLine();

 // System.out.println("Nueva "+cadena );

 // cadena = borrarEspacio(ncadena);

 //datos[]= new int[m];

 datos = cargarValores(ncadena, m);

 for (int j = 0; j < m; j++) {

 for (int k = 0; k < m; k++) {

 MD[i][k] = datos[k];

 }// fin k

 }// fin J

 }//fin i

 /\*Recorrer matriz \*/

 System.out.print(" ");

 int a = 1;

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

 a = i + 1;

 System.out.print(a + completarCadena(0, 1));

 }

 // System.out.println();

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

 // cadena=leer.next();

 System.out.println();

 for (int j = 0; j < m; j++) {

 tc = MD[i][j];

 if (i - 1 >= 0) { // 1

 tc += MD[i - 1][j];

 }

 if (i == 0) { // caso espacial

 if ((i + 1 < n) && (j - 1) >= 0) {

 tc += MD[i + 1][j - 1];

 }

 if ((i + 1 < n) && (j + 1) < m) {

 tc += MD[i + 1][j + 1];

 }

 if ((j > 0) && (j + 1) < m - 2) {// vertical dcho

 tc += MD[i][j + 1];

 }

 }// fin si i== 0

 if ((j == m - 1) && (i > 0) && (i < n - 2)) {

 tc += MD[i][j];

 }

 if ((i == n - 1) && (j > 0) && (j < m - 2)) {

 tc += MD[i][j];

 }

 if ((i == n - 1) && (j == m - 1)) {

 tc += MD[i][j];

 }

 if ((i - 1 >= 0) && (j + 1) < m) { // 2

 tc += MD[i - 1][j + 1];

 }

 if ((j + 1 < m)) { //3

 tc += MD[i][j + 1];

 }

 if ((j + 1 == m - 1)) { //3.1

 tc += MD[i][j + 1];

 }

 if ((i + 1 < n) && (j + 1 < m)) {// 4

 tc += MD[i + 1][j + 1];

 }

 if ((i + 1 < n)) {// 5

 tc += MD[i + 1][j];

 }

 if ((i + 1 < n) && (j - 1 >= 0)) {// 6

 tc += MD[i + 1][j - 1];

 }

 if ((j - 1 >= 0)) {//7

 tc += MD[i][j - 1];

 }

 if ((i - 1 >= 0) && (j - 1 >= 0)) { // 8

 tc += MD[i - 1][j - 1];

 }

 if ((i - 1 == n - 1) && (j + 1 == m - 1)) {

 tc += MD[i - 1][j + 1];

 }

 if ((i + 1 == n - 1) && (j + 1 == m - 1)) {

 tc += MD[i - 1][j];

 tc += MD[i - 1][j + 1];

 tc += MD[i][j + 1];

 tc += MD[i + 1][j + 1];

 tc += MD[i + 1][j];

 tc += MD[i + 1][j - 1];

 tc += MD[i][j - 1];

 tc += MD[i - 1][j - 1];

 }

 if ((i + 2 == n - 1) && (j + 1 == m - 1)) {

 tc += MD[i - 1][j];

 tc += MD[i - 1][j + 1];

 tc += MD[i][j + 1];

 tc += MD[i + 1][j + 1];

 tc += MD[i + 1][j];

 tc += MD[i + 1][j - 1];

 tc += MD[i][j - 1];

 tc += MD[i - 1][j - 1];

 }

 if (j == 0) {

 int b = i + 1;

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

 }

 if ((tc / 5) > 10) {

 System.out.print("\*" + completarCadena(0, 1));

 // System.out.print("[\* ]" );

 } else {

 System.out.print(completarCadena(0, 2));

 // System.out.print(tc);

 }

 if (j == m - 1) {

 int c = i + 1;

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

 }

 }//

 }

 System.out.println();

 System.out.print(completarCadena(0, 1));

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

 System.out.print(i + completarCadena(0, 1));

 }

 }

 public static int[] cargarValores(String cad, int v) {

 int nvalores[] = new int[v];

 String nuevaCadena = "";

 int h = 0;

 String palabra = "";

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

 if (cad.charAt(i) == ' ') {

 nvalores[h] = Integer.parseInt(palabra.trim());

 h++;

 palabra = "";

 } else {

 palabra += cad.charAt(i);

 }

 }

 return nvalores;

 }

 public static String completarCadena(int longitudCadena, int longitudCampo) {

 String cadena = "";

 int diferencia = 0;

 String espacios = "";

 if (longitudCadena < longitudCampo) {

 diferencia = longitudCampo - longitudCadena;

 espacios = "%1$" + "-" + diferencia + "s";

//cadena = String.format("%1$-10s",cadena);

 /\*Clase String.format(java 1.5 en adelante)

 Cadenas al final\*/

 cadena = String.format(espacios, cadena);

 }

 return cadena;

 }

 public static void main(String[] args) throws IOException {

 cadena = leer.readLine();

 cadena = cadena.replace(" ", "");

 n = Integer.parseInt(String.valueOf(cadena.charAt(0)));

 m = Integer.parseInt(String.valueOf(cadena.charAt(1)));

 buscar(n, m);

 }

}

3. Saber si una palabra es palindroma o no

import java.util.Scanner;

//ejemplo para el leghtn

public class Ej3 {

public static void main(String[] args) {

 Scanner sc=new Scanner(System.in);

 System.out.println("Ingrese la palabra: ");

 String str=sc.nextLine();

 int cont=1;

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

 if(str.charAt(i) != str.charAt(str.length()-1-i)){

 cont=0;

 break;

 }

 }

 System.out.println( cont==1 ? "Es palindromo": "No es palindromo");

 }

}

4.

**CUATRO EN LÍNEA**

**Nombre de archivo de fuente:** (Java: Cuatro.java, C++: Cuatro.cpp, C#: Cuatro.cs)

Un popular juego de mesa de los 80 era cuatro en línea. Este consistía en una matriz de (7 columnas por 6 filas), las fichas eran colocadas desde la parte superior (por las columnas) y estas caían hasta encontrar el primer espacio libre dentro de las la línea correspondiente. Los jugadores iban alternando colocando fichas (blancas o negras) por la columnas hasta que uno de los jugadores lograba obtener cuatro fichas en línea (horizontal, vertical o diagonal), o hasta que todas las fichas fueran colocadas.

Vamos a modelar una versión más simple del juego donde solamente vamos a identificar cuál de los jugadores ganan el juego al hacer primer una línea vertical de cuatro

El juego consistir en una serie de tableros: cada uno de 42 valores numéricos de 1 al 7, distribuidos en seis líneas de 7 números (del 1 al 7 puestos de forma aleatoria) que describen un juego en particular. Cada número indica por que columna están entrando las fichas. El jugador A (ficha blanca) siempre comienza. El programa debe identificar qué jugador ganó (Winner\_A ó Winner\_B) o si hay empate (Tie).

La primera línea de la entrada identifica el número total de tableros:

3

1234567

1234567

1234567

1234567

1234567

1234567

1212121

2345673

4567345

6734567

3456712

1323435

3671245

6712456

7124567

1245671

2456733

La salida será:

Tie

Winner\_A

Winner\_B

5. **Espejo.cpp ó Espejo.java**

Supongamos que se nos da una matriz **b** de la medida **x** \* **y**, vamos a determinar el funcionamiento para crear el espejo de la matriz **b**. La matriz resultante de la creación del espejo de la matriz **b** es una matriz **c** de **2x** \* **y**, que tiene las siguientes propiedades:

* La mitad superior de la matriz **c** (filas con números del 1 al **x** ) coincide exactamente con **b**;
* La mitad inferior de la matriz **c** (filas con números de **x** + 1 a **2x** ) es simétrica a la superior;
* En este caso la línea de simetría será la línea que separa las dos mitades (la línea que va en el medio, entre las filas de **x** y **x + 1** ) .

Si se tiene la matriz ***a*** de **m \* n** y se quiere encontrar la matriz **b**, que se puede transformar en la matriz ***a***, si vamos a realizar sobre ella varios (posiblemente cero) reflejos. ¿El mínimo número de filas que puede contener tal matriz es?

ENTRADA

* La primera línea contiene dos enteros, n y m ( 1 ≤ n , m ≤ 100 ).
* Cada una de las siguientes n líneas contiene m enteros - Los elementos de la matriz ***a.***
* La línea de orden ***i*** contiene enteros ***ai*1, *ai*2, ..., *aim*** (0 ≤ ***aij***≤ 1) — La *i*- esima fila de la matriz ***a***.

SALIDA

* En la única línea imprimir la respuesta al problema - el número mínimo de filas de la matriz b.

EJEMPLOS

ENTRADA

4 3
0 0 1
1 1 0
1 1 0
0 0 1

SALIDA

2

ENTRADA

3 3
0 0 0
0 0 0
0 0 0

SALIDA

3

ENTRADA

8 1
0
1
1
0
0
1
1
0

SALIDA

2

#include <iostream>

using namespace std;

bool a[1000][1000];

int n,m;

int ism()

{

 if (n%2)

 return false;

 int k=(n/2)+1;

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

 for (int j=1;j<=m;j++)

 if (a[i][j]!=a[n-i+1][j]){

 return false;

 }

 return true;

}

int main()

{

 cin>>n>>m;

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

 for (int j=1;j<=m;j++)

 cin>>a[i][j];

 while (ism())

 n= n/2;

 cout<<n;

}

**6.ESCRIBIENDO UN SMS**

**Nombre Archivo Fuente:** validarmensaje.java

Los celulares se han convertido en una parte esencial de la vida moderna. Además de llamadas de voz, los celulares pueden ser usados para enviar mensajes de texto, los cuales son conocidos como SMS en su forma corta. A diferencia de los teclados de ordenador, muchos teléfonos celulares han limitado el número de teclas. Para acomodar todo el alfabeto, las letras son compactadas dentro de una sola tecla. Por eso, para escribir algunos caracteres, es necesario que la tecla sea presionada en repetidas ocasiones hasta que el carácter sea mostrado en la pantalla.

En este problema, estamos interesados en saber el número de veces que se presionan las teclas para un mensaje en particular.

El teclado de nuestro celular tiene la siguiente distribución:



En la grilla de arriba, cada celda representa una tecla. SP significa un espacio. Si se desea escribir la letra “a”, tenemos que presionar una vez, para tipear la “b” la misma tecla debe ser presionada dos veces y para “c” tres veces. De igual forma, para escribir “d” es necesario presionar la tecla donde aparece una vez, para “e” dos veces la misma tecla y para “f” tres veces. Esto es también aplicable al resto de las teclas y letras. Note que para escribir un espacio solo es necesario oprimir la tecla donde aparece una sola vez.

package ejercicio;

import java.util.Scanner;

import java.io.\*;

//ejercicicio Hector Dario Cardona Arenas

public class validarmensaje

{

 public static void main(String[] args)

 {

 int conta=0,n=0;

 int vect[];

 vect=new int[n];

 try

 {

 InputStreamReader ncas = new InputStreamReader(System.in);

 BufferedReader flujoN = new BufferedReader(ncas);

 n=Integer.parseInt(flujoN.readLine());

 vect=new int[n];

 }

 catch(IOException e)

 {

 System.err.println("Error: " + e.getMessage());

 }

 String cadena="",sdato="";

 for(int j=0;j<n;j++){

 System.out.print(" ");

 Scanner leer=new Scanner(System.in);

 cadena=leer.nextLine();

 /\* try

 {

 InputStreamReader isr = new InputStreamReader(System.in);

 BufferedReader flujoE = new BufferedReader(isr);

 cadena = flujoE.readLine(

 }

 catch(IOException e)

 {

 System.err.println("Error: " + e.getMessage());

 }\*/

 char[] arrayChar = cadena.toCharArray();

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

 {

 if( arrayChar[i] == ' ')

 {

 conta=conta+1;

 }

 if( arrayChar[i] == 'a')

 {

 conta=conta+1;

 }

 if( arrayChar[i] == 'b')

 {

 conta=conta+2;

 }

 if( arrayChar[i] == 'c')

 {

 conta=conta+3;

 }

 if( arrayChar[i] == 'd')

 {

 conta=conta+1;

 }

 if( arrayChar[i] == 'e')

 {

 conta=conta+2;

 }

 if( arrayChar[i] == 'f')

 {

 conta=conta+3;

 }

 if( arrayChar[i] == 'g')

 {

 conta=conta+1;

 }

 if( arrayChar[i] == 'h')

 {

 conta=conta+2;

 }

 if( arrayChar[i] == 'i')

 {

 conta=conta+3;

 }

 if( arrayChar[i] == 'j')

 {

 conta=conta+1;

 }

 if( arrayChar[i] == 'k')

 {

 conta=conta+2;

 }

 if( arrayChar[i] == 'l')

 {

 conta=conta+3;

 }

 if( arrayChar[i] == 'm')

 {

 conta=conta+1;

 }

 if( arrayChar[i] == 'n')

 {

 conta=conta+2;

 }

 if( arrayChar[i] == 'o')

 {

 conta=conta+3;

 }

 if( arrayChar[i] == 'p')

 {

 conta=conta+1;

 }

 if( arrayChar[i] == 'q')

 {

 conta=conta+2;

 }

 if( arrayChar[i] == 'r')

 {

 conta=conta+3;

 }

 if( arrayChar[i] == 's')

 {

 conta=conta+4;

 }

 if( arrayChar[i] == 't')

 {

 conta=conta+1;

 }

 if( arrayChar[i] == 'u')

 {

 conta=conta+2;

 }

 if( arrayChar[i] == 'v')

 {

 conta=conta+3;

 }

 if( arrayChar[i] == 'w')

 {

 conta=conta+1;

 }

 if( arrayChar[i] == 'x')

 {

 conta=conta+2;

 }

 if( arrayChar[i] == 'y')

 {

 conta=conta+3;

 }

 if( arrayChar[i] == 'z')

 {

 conta=conta+4;

 }

 }

 vect[j]=conta;

 conta=0;

 }

// System.out.println("El numero de veces que oprimio tecla fue \n");

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

 System.out.print("caso #"+(i+1)+":"+vect[i]+"\n");

 }

}

}

**DESCOMPOSICION EN FACTORES**

**Grado 11**

**Nombre del archivo fuente:** Factores.C, Factores.CPP ó Factores.Java

Hacer un programa que descomponga un número en sus factores. Es decir:

   

**Ejemplo**: al ejecutar el programa, queda esperando un número, si el usuario presiona 72 <Enter> el programa debe tener la siguiente salida:



**Nota**: Su programa debe tener una salida exacta a la que muestra el ejemplo, no coloque mensajes ni información adicional.

import java.util.Scanner;

public class Factores {

 public static void main(String[] args) {

 Scanner sc = new Scanner(System.in);

 int n = sc.nextInt();

 int m[][] = new int[n][2];

 int t = 0;

 m[t][0] = n;

 while (!esPrimo(m[t][0])) {

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

 if (m[t][0] % i == 0) {

 m[t][1] = i;

 break;

 }

 }

 m[t + 1][0] = m[t][0] / m[t][1];

 t++;

 }

 m[t][1] = m[t][0];

 t++;

 String aux = n + " = ";

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

 aux = aux + m[i][1] + " x ";

 }

 n = aux.length();

 aux = aux.substring(0, n - 3);

 System.out.println(aux);

 }

 public static boolean esPrimo(int n) {

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

 if (n % i == 0) {

 return false;

 }

 }

 return true;

 }

}

**Ejercicio de Agencia de viajes**

import java.io.BufferedReader;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.InputStreamReader;

import java.text.DateFormatSymbols;

import java.util.LinkedHashMap;

import java.util.LinkedList;

import java.util.List;

import java.util.Locale;

import java.util.Map;

public class Agencias {

 static final String[] agencias = {"Lomas", "Vallejo", "Perisur", "Delvalle", "Oriente"};

 public static void main(String[] args) throws FileNotFoundException, IOException {

 BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

 int mes = 0, totalLomas = 0, sumaDiciembre = 0, mayoresVentasMayo = 0;

 String linea, agenciaMayorVentasMayo = agencias[0];

 DateFormatSymbols dfs = new DateFormatSymbols(new Locale("es", "ES"));

 String[] meses = dfs.getMonths();

 for (int i = 0; i < meses.length - 1; i++) {

 String tmp = meses[i];

 meses[i] = String.valueOf(tmp.charAt(0)).toUpperCase() + tmp.substring(1);

 }

 int[] menoresValores = new int[agencias.length];

 Map<String, List<String>> menoresVentas = new LinkedHashMap<String, List<String>>() {

 {

 put(agencias[0], new LinkedList<String>());

 put(agencias[1], new LinkedList<String>());

 put(agencias[2], new LinkedList<String>());

 put(agencias[3], new LinkedList<String>());

 put(agencias[4], new LinkedList<String>());

 }

 };

 int c=0;

 while (c<12) {

 linea = br.readLine();

 String[] valores = linea.split(" ");

 int indAg = 0;

 for (String valor : valores) {

 int valorVentas = Integer.parseInt(valor);

 if (indAg == 0) {

 totalLomas += valorVentas;

 }

 if (mes == 0) {

 menoresValores[indAg] = valorVentas;

 menoresVentas.get(agencias[indAg]).add(meses[mes]);

 } else {

 //System.out.println("valores" + valorVentas +"- "+menoresValores[indAg]);

 if (valorVentas < menoresValores[indAg]) {

 menoresValores[indAg] = valorVentas;

 List<String> mesesMenoresVentas = new LinkedList<String>();

 mesesMenoresVentas.add(meses[mes]);

 menoresVentas.put(agencias[indAg], mesesMenoresVentas);

 } else if (valorVentas == menoresValores[indAg]) {

 menoresVentas.get(agencias[indAg]).add(meses[mes]);

 }

 if (mes == 11) {

 sumaDiciembre += valorVentas;

 } else if (mes == 4) {

 if (indAg == 0) {

 mayoresVentasMayo = valorVentas;

 } else if (valorVentas > mayoresVentasMayo) {

 mayoresVentasMayo = valorVentas;

 agenciaMayorVentasMayo = agencias[indAg];

 }

 }

 }

 indAg++;

 }

 mes++;

 c++;

 }

 System.out.println("Total Lomas: " + totalLomas);

 System.out.println("Promedio Diciembre: " + (sumaDiciembre / agencias.length));

 System.out.println("Agencia con mayor venta â€“ Mes 05: " + agenciaMayorVentasMayo + " con " + mayoresVentasMayo);

 System.out.println("Mes que registrÃ³ menor venta");

 int indAg = 0;

 for (String agencia : agencias) {

 System.out.println(agencia + ": " + menoresValores[indAg++]);

 for (String string : menoresVentas.get(agencia)) {

 System.out.println(string);

 }

 }

 }

}

***Reto1 Categoria 1***

***Una fórmula para la elaboración de un Producto x tiene un costo y biene determinado por la siguiente formula 3x + 2y - 3z***

***Existe los valores que se toman como referencias estan almacendos en 3 arreglos de igual tamañ0 N.***

***x, y, z***

|  |  |  |  |
| --- | --- | --- | --- |
| ***x*** | ***y*** | ***z*** | ***resultado*** |
| ***788*** | ***200*** | ***300*** | ***-11064*** |
| ***500*** | ***450*** | ***780*** | ***......*** |
| ***700*** | ***780*** | ***900*** | ***........*** |
| ***56000*** | ***6890*** | ***2300*** | ***.......*** |
| ***n*** | ***n*** | ***n*** | ***......*** |

***Realice un programa en java que permita evaluar el resultado:***

***datos de prueba de entrada***

|  |  |
| --- | --- |
| ***Datos de entrada*** | ***datos de salida*** |
|  |  |

import java.util.Scanner;

public class reto1\_10 {

public static void main(String args[])

{

Scanner leer= new Scanner(System.in);

int val= leer.nextInt();

int z[]=new int[val];

int x[]=new int[val];

int y[]=new int[val];

int vz,vx,vy;

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

{

z[i]=leer.nextInt();

x[i]=leer.nextInt();

y[i]=leer.nextInt();

}

System.out.println();

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

{

 int res = 3\*x[i] + 2\*y[i]- 3\*z[i];

 //System.out.println(x[i]);

 //System.out.println(y[i]);

 // System.out.println(z[i]);

 System.out.println(res);

}

}

}

***Reto 2 Gráfica estadística Categoría 1(10)***

***Una empresa desea ver de una forma gráfica los resultados de las cantidades vendidas de un producto en una semana.***

***La empresa tiene su tienda abierta de Lunes a Domingo***

***como se vé en el siguiente arreglo:***

***Cantidades vendidas del producto x por día:***

|  |  |
| --- | --- |
| ***Dia*** | ***Cantidades*** |
| ***Lunes*** | ***2*** |
| ***Martes*** | ***3*** |
| ***Mierco*** | ***n..*** |
| ***Jueves*** | ***n...*** |
| ***Viernes*** | ***n..*** |
| ***Sábado*** | ***n..*** |
| ***Domingo*** | ***n..*** |

***el resultado de salida***

|  |  |
| --- | --- |
| ***Datos de entrada*** | ***Datos de salida*** |
|  |  |

import java.util.\*;

public class reto2\_10 {

 public static void main( String args[] )

 {

 int arreglo[] = new int[7];

 Scanner leer = new Scanner(System.in);

 String dia[] = { "Lunes", "Martes", "Mierc", "Jueves", "Viernes", "Sábado", "Domingo"};

 String salida="";

 for ( int contador = 0; contador < arreglo.length; contador++ ) {

 arreglo[contador] =leer.nextInt();

 salida += "\n" + dia[contador] +"\t "+ arreglo[ contador ] + "\t";

 for ( int estrellas = 0; estrellas < arreglo[ contador ]; estrellas++ )

 salida += "\*";

 }

 System.out.println(salida);

 System.exit( 0 );

 }

}

***Reto2 Categoria 2 (11)***

***Los costos de produccción fijos de 4 productos de una empresa x bienen determinados por la siguiente expresión algebraica 2X + 3w+2t***

***Reealice una programa que permita simular el costo capturando cada uno de los valores***

***o sea x, w ,t Ejemplo :***

***x=3;***

 ***w=4;***

***t= 8***

***El resultado sería reemplado en la formula dada en este caso seria igual a 34***

***La empresa desea un programa que permita simular las costos para dichos productos***

***mas eficiente.***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Producto*** | ***x*** | ***w*** | ***t*** | ***Costo T*** |
| ***A*** | ***3*** | ***4*** | ***8*** | ***34*** |
| ***B*** |  |  |  |  |
| ***C*** |  |  |  |  |
| ***D*** |  |  |  |  |

|  |  |
| --- | --- |
|  |  |
| Datos entrada | Datos salida |

***import java.util.Random;***

***import java.util.Scanner;***

***public class Ejemplo {***

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

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

***int n=0;// para capturar n datos***

***int r=0;***

***String producto []={"A","B","C","D"};***

***String v[]={"X","W","T"};***

***int valores[]= new int[3];***

***int matriz[ ][ ] = new int [4][3];***

***int x,w,t=0;***

***//abrir el ciclo para recorrer las filas***

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

 ***System.out.println(producto[i]);***

***for (int j = 0; j < matriz[i].length; j++) {***

 ***System.out.print(v[j] +"\t");***

 ***n= leer.nextInt();***

 ***matriz[i][j]=n;***

 ***}//fin para j***

 ***}// fin para i***

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

***for (int j = 0; j <matriz[i].length; j++) {***

***}// cierro para j***

 ***x= matriz[i][0];***

 ***w= matriz[i][1];***

 ***t= matriz[i][2];***

 ***r = 2\* x + 3\* w + 2\*t;***

 ***System.out.println(" X: " +x +" W :" + w +" T:" + t + ":" + r );***

***}// cierro para i***

 ***}***

***}***

***Reto 1 Categoría 2 11***

***En un arreglo de dos dimensiones(Matriz) , de igual número filas por columnas(desconocido)***

***se deben agregar una serie de valores numéricos, al mostrar los datos en pantalla***

***debe aparecer debujada una x con (\*) asterizcos***

|  |  |
| --- | --- |
| ***Datos de Entrada*** | ***Datos de salida*** |
|  |  |

***import java.util.Scanner;***

***public class Ejemplo{***

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

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

***int dato = 0;// para la longitud***

***String n="";// para capturar n datos***

***//System.out.print("Por favor digite el tamaño de su matriz \t");***

***dato = leer.nextInt();***

 ***// declarar matriz de tipo entero***

***String matriz[ ][ ] = new String [dato][dato];***

***int sumaColumnas[ ]= new int[dato];***

***//abrir el ciclo para recorrer las filas***

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

***for (int j = 0; j < matriz[i].length; j++) {***

 ***// System.out.print("digite dato : "+"["+i+"]"+"["+j+"]"+"\t");***

 ***n= leer.next();***

 ***matriz[i][j]=n;***

 ***}//fin para j***

 ***}// fin para i***

***//System.out.println("\n Matriz de "+dato+" \* "+dato);***

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

***for (int j = 0; j <matriz[i].length; j++) {***

***if(i==j){***

 ***matriz[i][j]="\*";***

***}***

***if(j==(dato-(i+1))){***

 ***matriz[i][dato-(i+1)]="\*";***

***}***

***System.out.print(matriz[i][j] +"\t");// imprimo la matriz***

***}// cierro para j***

***//System.out.println("suma fila "+i+":"+sumaf);***

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

***}// cierro para i***

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

***matriz[i][dato-(i+1)]="+";***

 ***matriz[i][i]="+";***

***}// fin para***

***}// cierro main***

***}// cierro la clase matriz1***