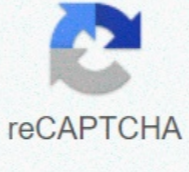




I'm not robot



**Continue**

## Ejercicios de arreglos en java resueltos pdf

```
1.- Empleando arreglos, mostrar n productos con sus respectivos precios. public class Ejercicio1 { public static void main(String[] args){ String utiles []={"pelota", "zapatilla","mochila","polo"}; double precios[]={35.5,89.60,45.99,25.70}; System.out.println("Lista de Productos y sus precios"); for (int u=0;u < utiles.length; u++){ System.out.println(utiles[u]+" "+precios[u]); } } }
public static void main(String[] args) { Scanner dato = new Scanner(System.in); int mayor = 0; int menor = 0; int[] numeros = new int[11]; for (int i = 0+1; i < 11; i++) { System.out.print("Ingrese el número " + (i) + " a evaluar :"); numeros[i] = dato.nextInt(); } for (int i = 1; i < numeros.length; i++) { if (mayor < numeros[i]) { mayor = numeros[i]; } } menor = mayor; for (int i = 1; i < numeros.length; i++) { if (menor > numeros[i]) { menor = numeros[i]; } } System.out.println(" el numero mayor es " + mayor + " el numero menor es : " + menor); } }
5.- Crear un arreglo, que nos pida ingresar una palabra, para mostrarlo letra por letra y la suma de sus caracteres. import java.util.*; public class Ejercicio5 { public static void main(String[] args) { Scanner dato=new Scanner(System.in); int c=0; System.out.println("Escriba una palabra"); String palabra=dato.nextLine(); char caracteres[]=new char[palabra.length()]; for(int i=0;i < palabra.length(); i++) { caracteres[i]=palabra.charAt(i); } for(int i=0;i < caracteres.length; i++) { System.out.println(caracteres[i]); } } }
10.- Ingresar por teclado n cantidad de números y ordenarlos en forma ascendente. import java.util.Scanner; public class Ejercicio10 { public static void main(String[] args) { Scanner dato = new Scanner(System.in); System.out.println("Ingrese la cantidad de numeros "); int elementos = dato.nextInt(); int[] n = new int[elementos]; for (int i = 0; i < n.length; i++) { n[i] = dato.nextInt(); } for (int i = 0; i < n.length; i++) { for (int j = i+1; j < n.length; j++) { if (n[i] > n[j]) { int temp = n[i]; n[i] = n[j]; n[j] = temp; } } } } }
//REGRESA EL NUMERO MINIMO DEL ARREGLO 95: public int minimo() 96: { 97: int minimo=100; 98: for(int x=0; x < n.length; x++) { 99: if(n[x] < minimo) minimo=n[x]; 100: } 101: return minimo; 102: }
//REGRESA EL NUMERO MAXIMO DEL ARREGLO 103: public int maximo() 104: { 105: int maximo=0; 106: for(int x=0; x < n.length; x++) { 107: if(n[x] > maximo) maximo=n[x]; 108: } 109: } 110: return maximo; 111: }
//REGRESA EL NUMERO DE ELEMENTOS DEL ARREGLO 112: public int cantidad() 113: { 114: return n.length; 115: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 116: public void mostrar() 117: { 118: for(int i=0; i < n.length; i++) { 119: System.out.println(n[i]); 120: } 121: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 122: public void mostrarArreglo() 123: { 124: for(int i=0; i < n.length; i++) { 125: System.out.println(n[i]); 126: } 127: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 128: public void mostrarArregloArreglo() 129: { 130: for(int i=0; i < n.length; i++) { 131: for(int j=0; j < n.length; j++) { 132: System.out.println(n[i][j]); 133: } 134: } 135: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 136: public void mostrarArregloArregloArreglo() 137: { 138: for(int i=0; i < n.length; i++) { 139: for(int j=0; j < n.length; j++) { 140: for(int k=0; k < n.length; k++) { 141: System.out.println(n[i][j][k]); 142: } 143: } 144: } 145: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 146: public void mostrarArregloArregloArregloArreglo() 147: { 148: for(int i=0; i < n.length; i++) { 149: for(int j=0; j < n.length; j++) { 150: for(int k=0; k < n.length; k++) { 151: for(int l=0; l < n.length; l++) { 152: System.out.println(n[i][j][k][l]); 153: } 154: } 155: } 156: } 157: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 158: public void mostrarArregloArregloArregloArregloArreglo() 159: { 160: for(int i=0; i < n.length; i++) { 161: for(int j=0; j < n.length; j++) { 162: for(int k=0; k < n.length; k++) { 163: for(int l=0; l < n.length; l++) { 164: for(int m=0; m < n.length; m++) { 165: System.out.println(n[i][j][k][l][m]); 166: } 167: } 168: } 169: } 170: } 171: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 172: public void mostrarArregloArregloArregloArregloArregloArreglo() 173: { 174: for(int i=0; i < n.length; i++) { 175: for(int j=0; j < n.length; j++) { 176: for(int k=0; k < n.length; k++) { 177: for(int l=0; l < n.length; l++) { 178: for(int m=0; m < n.length; m++) { 179: for(int n=0; n < n.length; n++) { 180: System.out.println(n[i][j][k][l][m][n]); 181: } 182: } 183: } 184: } 185: } 186: } 187: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 188: public void mostrarArregloArregloArregloArregloArregloArregloArreglo() 189: { 190: for(int i=0; i < n.length; i++) { 191: for(int j=0; j < n.length; j++) { 192: for(int k=0; k < n.length; k++) { 193: for(int l=0; l < n.length; l++) { 194: for(int m=0; m < n.length; m++) { 195: for(int n=0; n < n.length; n++) { 196: for(int o=0; o < n.length; o++) { 197: System.out.println(n[i][j][k][l][m][n][o]); 198: } 199: } 200: } 201: } 202: } 203: } 204: } 205: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 206: public void mostrarArregloArregloArregloArregloArregloArregloArregloArreglo() 207: { 208: for(int i=0; i < n.length; i++) { 209: for(int j=0; j < n.length; j++) { 210: for(int k=0; k < n.length; k++) { 211: for(int l=0; l < n.length; l++) { 212: for(int m=0; m < n.length; m++) { 213: for(int n=0; n < n.length; n++) { 214: for(int o=0; o < n.length; o++) { 215: for(int p=0; p < n.length; p++) { 216: System.out.println(n[i][j][k][l][m][n][o][p]); 217: } 218: } 219: } 220: } 221: } 222: } 223: } 224: } 225: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 226: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 227: { 228: for(int i=0; i < n.length; i++) { 229: for(int j=0; j < n.length; j++) { 230: for(int k=0; k < n.length; k++) { 231: for(int l=0; l < n.length; l++) { 232: for(int m=0; m < n.length; m++) { 233: for(int n=0; n < n.length; n++) { 234: for(int o=0; o < n.length; o++) { 235: for(int p=0; p < n.length; p++) { 236: for(int q=0; q < n.length; q++) { 237: System.out.println(n[i][j][k][l][m][n][o][p][q]); 238: } 239: } 240: } 241: } 242: } 243: } 244: } 245: } 246: } 247: } 248: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 249: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 250: { 251: for(int i=0; i < n.length; i++) { 252: for(int j=0; j < n.length; j++) { 253: for(int k=0; k < n.length; k++) { 254: for(int l=0; l < n.length; l++) { 255: for(int m=0; m < n.length; m++) { 256: for(int n=0; n < n.length; n++) { 257: for(int o=0; o < n.length; o++) { 258: for(int p=0; p < n.length; p++) { 259: for(int q=0; q < n.length; q++) { 260: for(int r=0; r < n.length; r++) { 261: System.out.println(n[i][j][k][l][m][n][o][p][q][r]); 262: } 263: } 264: } 265: } 266: } 267: } 268: } 269: } 270: } 271: } 272: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 273: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 274: { 275: for(int i=0; i < n.length; i++) { 276: for(int j=0; j < n.length; j++) { 277: for(int k=0; k < n.length; k++) { 278: for(int l=0; l < n.length; l++) { 279: for(int m=0; m < n.length; m++) { 280: for(int n=0; n < n.length; n++) { 281: for(int o=0; o < n.length; o++) { 282: for(int p=0; p < n.length; p++) { 283: for(int q=0; q < n.length; q++) { 284: for(int r=0; r < n.length; r++) { 285: for(int s=0; s < n.length; s++) { 286: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s]); 287: } 288: } 289: } 290: } 291: } 292: } 293: } 294: } 295: } 296: } 297: } 298: } 299: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 300: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 301: { 302: for(int i=0; i < n.length; i++) { 303: for(int j=0; j < n.length; j++) { 304: for(int k=0; k < n.length; k++) { 305: for(int l=0; l < n.length; l++) { 306: for(int m=0; m < n.length; m++) { 307: for(int n=0; n < n.length; n++) { 308: for(int o=0; o < n.length; o++) { 309: for(int p=0; p < n.length; p++) { 310: for(int q=0; q < n.length; q++) { 311: for(int r=0; r < n.length; r++) { 312: for(int s=0; s < n.length; s++) { 313: for(int t=0; t < n.length; t++) { 314: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t]); 315: } 316: } 317: } 318: } 319: } 320: } 321: } 322: } 323: } 324: } 325: } 326: } 327: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 328: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 329: { 330: for(int i=0; i < n.length; i++) { 331: for(int j=0; j < n.length; j++) { 332: for(int k=0; k < n.length; k++) { 333: for(int l=0; l < n.length; l++) { 334: for(int m=0; m < n.length; m++) { 335: for(int n=0; n < n.length; n++) { 336: for(int o=0; o < n.length; o++) { 337: for(int p=0; p < n.length; p++) { 338: for(int q=0; q < n.length; q++) { 339: for(int r=0; r < n.length; r++) { 340: for(int s=0; s < n.length; s++) { 341: for(int t=0; t < n.length; t++) { 342: for(int u=0; u < n.length; u++) { 343: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u]); 344: } 345: } 346: } 347: } 348: } 349: } 350: } 351: } 352: } 353: } 354: } 355: } 356: } 357: } 358: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 359: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 360: { 361: for(int i=0; i < n.length; i++) { 362: for(int j=0; j < n.length; j++) { 363: for(int k=0; k < n.length; k++) { 364: for(int l=0; l < n.length; l++) { 365: for(int m=0; m < n.length; m++) { 366: for(int n=0; n < n.length; n++) { 367: for(int o=0; o < n.length; o++) { 368: for(int p=0; p < n.length; p++) { 369: for(int q=0; q < n.length; q++) { 370: for(int r=0; r < n.length; r++) { 371: for(int s=0; s < n.length; s++) { 372: for(int t=0; t < n.length; t++) { 373: for(int u=0; u < n.length; u++) { 374: for(int v=0; v < n.length; v++) { 375: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v]); 376: } 377: } 378: } 379: } 380: } 381: } 382: } 383: } 384: } 385: } 386: } 387: } 388: } 389: } 390: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 391: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 392: { 393: for(int i=0; i < n.length; i++) { 394: for(int j=0; j < n.length; j++) { 395: for(int k=0; k < n.length; k++) { 396: for(int l=0; l < n.length; l++) { 397: for(int m=0; m < n.length; m++) { 398: for(int n=0; n < n.length; n++) { 399: for(int o=0; o < n.length; o++) { 400: for(int p=0; p < n.length; p++) { 401: for(int q=0; q < n.length; q++) { 402: for(int r=0; r < n.length; r++) { 403: for(int s=0; s < n.length; s++) { 404: for(int t=0; t < n.length; t++) { 405: for(int u=0; u < n.length; u++) { 406: for(int v=0; v < n.length; v++) { 407: for(int w=0; w < n.length; w++) { 408: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w]); 409: } 410: } 411: } 412: } 413: } 414: } 415: } 416: } 417: } 418: } 419: } 420: } 421: } 422: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 423: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 424: { 425: for(int i=0; i < n.length; i++) { 426: for(int j=0; j < n.length; j++) { 427: for(int k=0; k < n.length; k++) { 428: for(int l=0; l < n.length; l++) { 429: for(int m=0; m < n.length; m++) { 430: for(int n=0; n < n.length; n++) { 431: for(int o=0; o < n.length; o++) { 432: for(int p=0; p < n.length; p++) { 433: for(int q=0; q < n.length; q++) { 434: for(int r=0; r < n.length; r++) { 435: for(int s=0; s < n.length; s++) { 436: for(int t=0; t < n.length; t++) { 437: for(int u=0; u < n.length; u++) { 438: for(int v=0; v < n.length; v++) { 439: for(int w=0; w < n.length; w++) { 440: for(int x=0; x < n.length; x++) { 441: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x]); 442: } 443: } 444: } 445: } 446: } 447: } 448: } 449: } 450: } 451: } 452: } 453: } 454: } 455: } 456: } 457: } 458: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 459: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 460: { 461: for(int i=0; i < n.length; i++) { 462: for(int j=0; j < n.length; j++) { 463: for(int k=0; k < n.length; k++) { 464: for(int l=0; l < n.length; l++) { 465: for(int m=0; m < n.length; m++) { 466: for(int n=0; n < n.length; n++) { 467: for(int o=0; o < n.length; o++) { 468: for(int p=0; p < n.length; p++) { 469: for(int q=0; q < n.length; q++) { 470: for(int r=0; r < n.length; r++) { 471: for(int s=0; s < n.length; s++) { 472: for(int t=0; t < n.length; t++) { 473: for(int u=0; u < n.length; u++) { 474: for(int v=0; v < n.length; v++) { 475: for(int w=0; w < n.length; w++) { 476: for(int x=0; x < n.length; x++) { 477: for(int y=0; y < n.length; y++) { 478: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x][y]); 479: } 480: } 481: } 482: } 483: } 484: } 485: } 486: } 487: } 488: } 489: } 490: } 491: } 492: } 493: } 494: } 495: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 496: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 497: { 498: for(int i=0; i < n.length; i++) { 499: for(int j=0; j < n.length; j++) { 500: for(int k=0; k < n.length; k++) { 501: for(int l=0; l < n.length; l++) { 502: for(int m=0; m < n.length; m++) { 503: for(int n=0; n < n.length; n++) { 504: for(int o=0; o < n.length; o++) { 505: for(int p=0; p < n.length; p++) { 506: for(int q=0; q < n.length; q++) { 507: for(int r=0; r < n.length; r++) { 508: for(int s=0; s < n.length; s++) { 509: for(int t=0; t < n.length; t++) { 510: for(int u=0; u < n.length; u++) { 511: for(int v=0; v < n.length; v++) { 512: for(int w=0; w < n.length; w++) { 513: for(int x=0; x < n.length; x++) { 514: for(int y=0; y < n.length; y++) { 515: for(int z=0; z < n.length; z++) { 516: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x][y][z]); 517: } 518: } 519: } 520: } 521: } 522: } 523: } 524: } 525: } 526: } 527: } 528: } 529: } 530: } 531: } 532: } 533: } 534: } 535: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 536: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 537: { 538: for(int i=0; i < n.length; i++) { 539: for(int j=0; j < n.length; j++) { 540: for(int k=0; k < n.length; k++) { 541: for(int l=0; l < n.length; l++) { 542: for(int m=0; m < n.length; m++) { 543: for(int n=0; n < n.length; n++) { 544: for(int o=0; o < n.length; o++) { 545: for(int p=0; p < n.length; p++) { 546: for(int q=0; q < n.length; q++) { 547: for(int r=0; r < n.length; r++) { 548: for(int s=0; s < n.length; s++) { 549: for(int t=0; t < n.length; t++) { 550: for(int u=0; u < n.length; u++) { 551: for(int v=0; v < n.length; v++) { 552: for(int w=0; w < n.length; w++) { 553: for(int x=0; x < n.length; x++) { 554: for(int y=0; y < n.length; y++) { 555: for(int z=0; z < n.length; z++) { 556: for(int aa=0; aa < n.length; aa++) { 557: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x][y][z][aa]); 558: } 559: } 560: } 561: } 562: } 563: } 564: } 565: } 566: } 567: } 568: } 569: } 570: } 571: } 572: } 573: } 574: } 575: } 576: } 577: } 578: } 579: } 580: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 581: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 582: { 583: for(int i=0; i < n.length; i++) { 584: for(int j=0; j < n.length; j++) { 585: for(int k=0; k < n.length; k++) { 586: for(int l=0; l < n.length; l++) { 587: for(int m=0; m < n.length; m++) { 588: for(int n=0; n < n.length; n++) { 589: for(int o=0; o < n.length; o++) { 590: for(int p=0; p < n.length; p++) { 591: for(int q=0; q < n.length; q++) { 592: for(int r=0; r < n.length; r++) { 593: for(int s=0; s < n.length; s++) { 594: for(int t=0; t < n.length; t++) { 595: for(int u=0; u < n.length; u++) { 596: for(int v=0; v < n.length; v++) { 597: for(int w=0; w < n.length; w++) { 598: for(int x=0; x < n.length; x++) { 599: for(int y=0; y < n.length; y++) { 600: for(int z=0; z < n.length; z++) { 601: for(int aa=0; aa < n.length; aa++) { 602: for(int bb=0; bb < n.length; bb++) { 603: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x][y][z][aa][bb]); 604: } 605: } 606: } 607: } 608: } 609: } 610: } 611: } 612: } 613: } 614: } 615: } 616: } 617: } 618: } 619: } 620: } 621: } 622: } 623: } 624: } 625: } 626: } 627: } 628: } 629: } 630: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 631: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 632: { 633: for(int i=0; i < n.length; i++) { 634: for(int j=0; j < n.length; j++) { 635: for(int k=0; k < n.length; k++) { 636: for(int l=0; l < n.length; l++) { 637: for(int m=0; m < n.length; m++) { 638: for(int n=0; n < n.length; n++) { 639: for(int o=0; o < n.length; o++) { 640: for(int p=0; p < n.length; p++) { 641: for(int q=0; q < n.length; q++) { 642: for(int r=0; r < n.length; r++) { 643: for(int s=0; s < n.length; s++) { 644: for(int t=0; t < n.length; t++) { 645: for(int u=0; u < n.length; u++) { 646: for(int v=0; v < n.length; v++) { 647: for(int w=0; w < n.length; w++) { 648: for(int x=0; x < n.length; x++) { 649: for(int y=0; y < n.length; y++) { 650: for(int z=0; z < n.length; z++) { 651: for(int aa=0; aa < n.length; aa++) { 652: for(int bb=0; bb < n.length; bb++) { 653: for(int cc=0; cc < n.length; cc++) { 654: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x][y][z][aa][bb][cc]); 655: } 656: } 657: } 658: } 659: } 660: } 661: } 662: } 663: } 664: } 665: } 666: } 667: } 668: } 669: } 670: } 671: } 672: } 673: } 674: } 675: } 676: } 677: } 678: } 679: } 680: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 681: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 682: { 683: for(int i=0; i < n.length; i++) { 684: for(int j=0; j < n.length; j++) { 685: for(int k=0; k < n.length; k++) { 686: for(int l=0; l < n.length; l++) { 687: for(int m=0; m < n.length; m++) { 688: for(int n=0; n < n.length; n++) { 689: for(int o=0; o < n.length; o++) { 690: for(int p=0; p < n.length; p++) { 691: for(int q=0; q < n.length; q++) { 692: for(int r=0; r < n.length; r++) { 693: for(int s=0; s < n.length; s++) { 694: for(int t=0; t < n.length; t++) { 695: for(int u=0; u < n.length; u++) { 696: for(int v=0; v < n.length; v++) { 697: for(int w=0; w < n.length; w++) { 698: for(int x=0; x < n.length; x++) { 699: for(int y=0; y < n.length; y++) { 700: for(int z=0; z < n.length; z++) { 701: for(int aa=0; aa < n.length; aa++) { 702: for(int bb=0; bb < n.length; bb++) { 703: for(int cc=0; cc < n.length; cc++) { 704: for(int dd=0; dd < n.length; dd++) { 705: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x][y][z][aa][bb][cc][dd]); 706: } 707: } 708: } 709: } 710: } 711: } 712: } 713: } 714: } 715: } 716: } 717: } 718: } 719: } 720: } 721: } 722: } 723: } 724: } 725: } 726: } 727: } 728: } 729: } 730: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 731: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 732: { 733: for(int i=0; i < n.length; i++) { 734: for(int j=0; j < n.length; j++) { 735: for(int k=0; k < n.length; k++) { 736: for(int l=0; l < n.length; l++) { 737: for(int m=0; m < n.length; m++) { 738: for(int n=0; n < n.length; n++) { 739: for(int o=0; o < n.length; o++) { 740: for(int p=0; p < n.length; p++) { 741: for(int q=0; q < n.length; q++) { 742: for(int r=0; r < n.length; r++) { 743: for(int s=0; s < n.length; s++) { 744: for(int t=0; t < n.length; t++) { 745: for(int u=0; u < n.length; u++) { 746: for(int v=0; v < n.length; v++) { 747: for(int w=0; w < n.length; w++) { 748: for(int x=0; x < n.length; x++) { 749: for(int y=0; y < n.length; y++) { 750: for(int z=0; z < n.length; z++) { 751: for(int aa=0; aa < n.length; aa++) { 752: for(int bb=0; bb < n.length; bb++) { 753: for(int cc=0; cc < n.length; cc++) { 754: for(int dd=0; dd < n.length; dd++) { 755: for(int ee=0; ee < n.length; ee++) { 756: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x][y][z][aa][bb][cc][dd][ee]); 757: } 758: } 759: } 760: } 761: } 762: } 763: } 764: } 765: } 766: } 767: } 768: } 769: } 770: } 771: } 772: } 773: } 774: } 775: } 776: } 777: } 778: } 779: } 780: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 781: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 782: { 783: for(int i=0; i < n.length; i++) { 784: for(int j=0; j < n.length; j++) { 785: for(int k=0; k < n.length; k++) { 786: for(int l=0; l < n.length; l++) { 787: for(int m=0; m < n.length; m++) { 788: for(int n=0; n < n.length; n++) { 789: for(int o=0; o < n.length; o++) { 790: for(int p=0; p < n.length; p++) { 791: for(int q=0; q < n.length; q++) { 792: for(int r=0; r < n.length; r++) { 793: for(int s=0; s < n.length; s++) { 794: for(int t=0; t < n.length; t++) { 795: for(int u=0; u < n.length; u++) { 796: for(int v=0; v < n.length; v++) { 797: for(int w=0; w < n.length; w++) { 798: for(int x=0; x < n.length; x++) { 799: for(int y=0; y < n.length; y++) { 800: for(int z=0; z < n.length; z++) { 801: for(int aa=0; aa < n.length; aa++) { 802: for(int bb=0; bb < n.length; bb++) { 803: for(int cc=0; cc < n.length; cc++) { 804: for(int dd=0; dd < n.length; dd++) { 805: for(int ee=0; ee < n.length; ee++) { 806: for(int ff=0; ff < n.length; ff++) { 807: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x][y][z][aa][bb][cc][dd][ee][ff]); 808: } 809: } 810: } 811: } 812: } 813: } 814: } 815: } 816: } 817: } 818: } 819: } 820: } 821: } 822: } 823: } 824: } 825: } 826: } 827: } 828: } 829: } 830: }
//REGRESA EL NUMERO DE ELEMENTOS QUE HAY EN EL ARREGLO 831: public void mostrarArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArregloArreglo() 832: { 833: for(int i=0; i < n.length; i++) { 834: for(int j=0; j < n.length; j++) { 835: for(int k=0; k < n.length; k++) { 836: for(int l=0; l < n.length; l++) { 837: for(int m=0; m < n.length; m++) { 838: for(int n=0; n < n.length; n++) { 839: for(int o=0; o < n.length; o++) { 840: for(int p=0; p < n.length; p++) { 841: for(int q=0; q < n.length; q++) { 842: for(int r=0; r < n.length; r++) { 843: for(int s=0; s < n.length; s++) { 844: for(int t=0; t < n.length; t++) { 845: for(int u=0; u < n.length; u++) { 846: for(int v=0; v < n.length; v++) { 847: for(int w=0; w < n.length; w++) { 848: for(int x=0; x < n.length; x++) { 849: for(int y=0; y < n.length; y++) { 850: for(int z=0; z < n.length; z++) { 851: for(int aa=0; aa < n.length; aa++) { 852: for(int bb=0; bb < n.length; bb++) { 853: for(int cc=0; cc < n.length; cc++) { 854: for(int dd=0; dd < n.length; dd++) { 855: for(int ee=0; ee < n.length; ee++) { 856: for(int ff=0; ff < n.length; ff++) { 857: for(int gg=0; gg < n.length; gg++) { 858: System.out.println(n[i][j][k][l][m][n][o][p][q][r][s][t][u][v][w][x][y][z][aa][bb][cc][dd][ee][ff][gg]); 859: } 860: } 861: } 862: } 863: } 864: } 865: } 866: } 867: } 868: } 869: } 870: } 871: } 872: } 873: } 874: } 875: } 876: } 877: } 878: } 879: } 880: }
//REGRESA EL
```