

```

1: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // Departamento de Ciencias de la Computación e Inteligencia Artificial
7: // Autor: Don Oreo
8: //
9: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
10:
11: // Mapa de distancias entre ciudades
12:
13: #include <iostream>
14: #include <cmath>
15: using namespace std;
16:
17: int main() {
18:     const int MAX_NUM_CIUDADES = 50;
19:     double mapa[MAX_NUM_CIUDADES][MAX_NUM_CIUDADES];
20:     int num_ciudades;
21:     const int CENTINELA = 0;
22:
23:     cout << "Mapa de distancias"
24:         << "\n\nIntroduzca los datos en el siguiente orden:"
25:         << "\n- Número de ciudades"
26:         << "\n- Distancias entre ellas en forma de matriz diagonal superior"
27:         << "\n\n";
28:
29:     cin >> num_ciudades;
30:
31:     for (int i = 0; i < num_ciudades; i++)
32:         for (int j = 0; j < num_ciudades; j++)
33:             mapa[i][j] = 0;
34:
35:     for (int i = 0; i < num_ciudades - 1; i++)
36:         for (int j = i+1; j < num_ciudades; j++){
37:             double dist;
38:
39:             cin >> dist;
40:             mapa[i][j] = mapa[j][i] = dist;
41:         }
42:
43:     // -----
44:
45:     int ciudad_mas_conectada;
46:     int max_conex = -1, num_conex;
47:
48:     // COMPUTO DE DATOS
49:
50:     for (int i = 0; i < num_ciudades; i++){
51:         num_conex = 0;
52:         for (int j = 0; j < num_ciudades; j++){
53:             if(mapa[i][j] != CENTINELA){
54:                 num_conex++;
55:             }
56:         }
57:         if(num_conex > max_conex){
58:             max_conex = num_conex;
59:             ciudad_mas_conectada = i;
60:         }
61:     }
62:
63:
64:     cout << "\nCiudad más conectada: " << ciudad_mas_conectada
65:         << " con un total de " << max_conex
66:         << " conexiones";
67: }
68: /*
69:     5
70:    50 100  0  150
71:      70  0   0
72:        60  80
73:         90
74: */
75: /*
76: Salida:
77:
78: Ciudad más conectada: 2 con un total de 4 conexiones
79: */

```

```
1: //////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////
9:
10: //DoubleToString
11:
12: #include <iostream>
13: #include <cmath>
14: using namespace std;
15:
16: string EliminaUltimo(string cadena){
17:     int ultimo;
18:     ultimo = cadena.size() - 1;
19:     while(cadena[ultimo] == '0' || cadena[ultimo] == '.'){
20:         cadena.erase(ultimo);
21:         ultimo = cadena.size() - 1;
22:     }
23:     return cadena;
24: }
25:
26: double Redondeado(double real, int decimales){
27:     real = round(real * pow(10,decimales));
28:     real = real / pow(10,decimales);
29:
30:     return real;
31: }
32: string DoubleToString(double real, int decimales){
33:     string cadena;
34:     real = Redondeado(real,decimales);
35:     cadena = to_string(real);
36:     cadena = EliminaUltimo(cadena);
37:     return cadena;
38: }
39:
40: int main(){
41:     const char TERMINADOR = '#';
42:     double real;
43:     int decimales;
44:     string cadena;
45:
46:     cout << "Convertir a string\n\n"
47:         << "Introduzca el real a convertir con las decimales a redondear" << TERMINADOR << endl;
48:
49:     cin >> real >> decimales;
50:
51:     cadena = DoubleToString(real,decimales);
52:
53:     cout << cadena << endl;
54: }
55:
```

```

1: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // Departamento de Ciencias de la Computación e Inteligencia Artificial
7: // Autor: Don Oreo
8: //
9: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
10:
11: // Elimina Ultimos
12:
13: #include <iostream>
14: #include <string>
15:
16: using namespace std;
17:
18: string LeeString(char terminador){
19:     string cadena;
20:     char caracter;
21:
22:     caracter = cin.get();
23:
24:     while (caracter != terminador){
25:         cadena.push_back(caracter);
26:         caracter = cin.get();
27:     }
28:
29:     return cadena;
30: }
31:
32: string EliminaUltimo(string cadena, char a_borrar){
33:     int ultimo;
34:
35:     ultimo = cadena.size() - 1;
36:     while(cadena[ultimo] == a_borrar){
37:         cadena.erase(ultimo);
38:         ultimo = cadena.size() - 1;
39:     }
40:     return cadena;
41: }
42:
43: /* Metodo 2
44: string EliminaUltimo(string cadena, char a_borrar){
45:     while( cadena.back() == a_borrar)
46:         cadena.pop_back();
47:     return cadena;
48: }
49: */
50: int main(){
51:     const char TERMINADOR = '#';
52:     char a_borrar;
53:     string cadena;
54:
55:     // Lectura
56:
57:     cout << "Lee string\n\n"
58:     << "Introduzca caracteres con terminador " << TERMINADOR
59:     << ". Y despues escribe el caracter ultimo a borrar." << endl;
60:
61:     cadena = LeeString(TERMINADOR);
62:
63:     a_borrar = cin.get();
64:
65:     cadena = EliminaUltimo(cadena, a_borrar);
66:
67:     cout << cadena;
68: }

```

```
1:
2: #include <iostream>
3: #include <string>
4: using namespace std;
5:
6: /**
7:  * 3. [Errores en funciones void] Encuentre los errores, si los hubiese, en las siguientes
8:  * funciones void:
9:  */
10:
11: /*
12: void EliminaUltimo(string cadena){
13:     cadena.pop_back();
14: }
15: */
16: //Una funcion void no puede modificar una variable ya que no la puede devolver
17:
18: /*
19: void Imprime(double valor){
20:     double valor;
21:     cout << valor;
22: }
23: */
24: //Se redeclara la variable valor
25:
26: /*
27: void Cuadrado (int entero){
28:     return entero*entero;
29: }
30: */
31: //Un void no puede devolver una variable mediante un return
32:
```

```

1: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
9:
10: #include <iostream>
11: using namespace std;
12:
13: int LeeIntMayorIgual(int min) {
14:     int a_leer;
15:     do
16:         cin >> a_leer;
17:     while (min > a_leer);
18:
19:     return a_leer;
20: }
21:
22: int LeeIntRango(int min,int max){
23:     int dato = min -1;    //Para que siempre esté fuera del intervalo y entre en el bucle
24:
25:     while(min > dato || max < dato)
26:         cin >> dato;
27:
28:     return dato;
29:
30: }
31:
32: int main(){
33:
34:     long min, max, dato;
35:
36:     cout << "\nIntroduzca el valor mínimo y el máximo"
37:         << "\nA continuación introduzca enteros en el rango anterior\n";
38:
39:     cin >> min;
40:
41:     max = LeeIntMayorIgual(min);
42:
43:     dato = LeeIntRango(min,max);
44:
45:     cout << dato << " está en el intervalo [" << min << "," << max << "]" << endl;
46: }

```

```
1: //////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////
9:
10: #include <iostream>
11: using namespace std;
12:
13: string LeeString(char terminador){
14:     string cadena;
15:     char car;
16:
17:     car = cin.get();
18:
19:     while (car != terminador){
20:         cadena.push_back(car);
21:         car = cin.get();
22:     }
23:
24:     return cadena;
25: }
26:
27: void ImprimeMarco (char car, int repeticiones){
28:     for(int i = 0; i < repeticiones; i++){
29:         cout << " * ";
30:     }
31: }
32:
33: void Enmarca(string mensaje, char caracter){
34:     int util;
35:     string marco;
36:
37:     util = mensaje.size();
38:     ImprimeMarco(caracter,util);
39:     cout << "\n" << mensaje << "\n";
40:     ImprimeMarco(caracter,util);
41:
42: }
43: int main(){
44:     const char TERMINADOR = '@';
45:     char car;
46:     string msj;
47:
48:     // Lectura
49:
50:     cout << "Presentacion\n\n"
51:           << "Introduzca caracteres con terminador " << TERMINADOR << "\n";
52:
53:     msj = LeeString(TERMINADOR);
54:     Enmarca(msj, car);
55:
56: }
```