

```

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:
12: using namespace std;
13:
14: int main(){
15:     const int MAX_PEDIDOS = 100;
16:     const int MAX_TECNICOS = 100;
17:     int tarifa[MAX_TECNICOS][MAX_PEDIDOS];
18:     bool A[MAX_TECNICOS][MAX_PEDIDOS];
19:     bool cogidos[MAX_PEDIDOS];
20:     int tecnicos,pedidos;
21:     int menor;
22:     int pos_menor = 100;    //Lo inicializamos a cualquier número que no se vaya a utilizar como posici
on en la matriz tarifa(como si pongo 123433124)
23:
24:     //Entrada de Datos
25:
26:     cout << "Introduce el numero de técnicos: ";
27:     cin >> tecnicos;
28:
29:     pedidos = tecnicos;
30:
31:     cout << "\nIntroduce la matriz tarifa: " << endl;
32:
33:     for(int i = 0; i < tecnicos; i++){
34:         for(int j = 0; j < pedidos; j++){
35:             cin >> tarifa[i][j];
36:
37:             //Cómputo
38:
39:             int precio_total = 0;
40:
41:             for(int i = 0; i < tecnicos; i++){
42:                 menor = 100;
43:                 pos_menor = 100;
44:                 for(int j = 0; j < pedidos; j++){
45:                     if(tarifa[i][j] < menor && cogidos[j] == false){
46:                         menor = tarifa[i][j];
47:                         pos_menor = j;
48:                     }
49:                 }
50:                 cogidos[pos_menor] = true;
51:                 A[i][pos_menor] = true;
52:                 precio_total += menor;
53:             }
54:
55:             //Salida de datos
56:
57:             /*Bool*/
58:             cout << "\n\n" << "Asignacion de pedidos:" << endl;
59:
60:             for(int i=0; i<tecnicos; i++){
61:                 cout << "\n";
62:
63:                 for(int j=0; j<pedidos; j++){
64:                     cout << A[i][j] << '\t';
65:                 }
66:             }
67:
68:             cout << endl << "-----";
69:             /*Asignacion de pedidos*/
70:
71:             for(int i = 0; i < tecnicos; i++){
72:                 for(int j = 0; j < pedidos; j++){
73:                     if(A[i][j] == true)
74:                         cout << "\nTécnico " << i << " --> Pedido " << j;
75:                 }
76:             }
77:
78:             cout <<endl << "-----";
79:             cout << "\nEl coste total es de: " << precio_total << endl;
80:
81:
82:             return 0;
83: }
84:
85:

```

```

1: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
9:
10:
11: // Sustituir carácter por vector
12: #include <iostream>
13: using namespace std;
14:
15:
16: int main(){
17:     const char TERMINADOR = '#';
18:     const int MAX_NUM_CARACT = 3e6; // Compile con la opción -Wl,--stack,21000000
19:     char v[MAX_NUM_CARACT],
20:         nuevo[MAX_NUM_CARACT];
21:     char a_borrar;
22:     char car;
23:     int i, util_v, util_nuevo;
24:
25:     // Lectura
26:
27:     car = cin.get();
28:     i = 0;
29:
30:     while (car != TERMINADOR){
31:         v[i] = car;
32:         car = cin.get();
33:         i++;
34:     }
35:
36:     util_v = i;
37:
38:     car = cin.get();
39:     i = 0;
40:
41:     while (car != TERMINADOR){
42:         nuevo[i] = car;
43:         car = cin.get();
44:         i++;
45:     }
46:
47:     util_nuevo = i-1;
48:
49:     a_borrar = cin.get();
50:
51:
52:     //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
53:     int contador=0;
54:
55:     /*Contamos el numero de ocurrencias a borrar*/
56:     for(int i=0; i<util_v;i++){
57:         if(v[i]==a_borrar){
58:             contador++;
59:         }
60:     }
61:
62:     int util_resultado=util_v;
63:     util_resultado+=contador*util_nuevo; //Obtenemos el util final
64:
65:     /*Inicializamos las variables de lectura/escritura*/
66:     int lec=util_v;
67:     int esc=util_resultado;
68:
69:     /** Comprueba si los caracteres de v coinciden con a_borrar
70:     *si coinciden, imprime el vector nuevo de derecha a izquierda
71:     *si no coincide, imprime el caracter de v a la izquierda del caracter de más a la derecha
72:     */
73:     //Nota:Los vectores leen de derecha a izquierda
74:
75:     while(lec>=0){
76:         if(v[lec]==a_borrar){
77:             for(int j=util_nuevo; j>=0; j--){
78:                 v[esc]=nuevo[j];
79:                 esc--;
80:             }
81:         }
82:         else{
83:             v[esc]=v[lec];
84:             esc--;
85:         }
86:         lec--;

```

```

87:     }
88:     //////////////////////////////////////
89:
90:     cout << "<";
91:
92:     for (int i = 0; i < util_resultado; i++)
93:         cout << v[i] ;
94:
95:     cout << ">";
96:
97: // Ejemplo de entrada:
98: // unoadosaa#TTU#a
99:
100: // Salida:
101: // unoTTUdosTTUTTU
102: }

```

```

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: #include <cmath>
12: using namespace std;
13:
14: double Redondeado(double numero,int cifra){
15:     numero = round(numero * pow(10,cifra));
16:     numero = numero / pow(10,cifra);
17:     return numero;
18: }
19:
20:
21: int main(){
22:     double numero;
23:     int cifra;
24:
25:     cin >> numero >> cifra;
26:     numero = Redondeado(numero,cifra);
27:
28:     cout << "El numero redondeado es: " << numero << endl;
29:
30:     return 0;
31:
32: }
33:
34:

```

```

1: //////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////
9:
10: // Elasticidad Precio-Demanda
11:
12:
13: #include <iostream>
14: #include <cmath>
15: using namespace std;
16:
17: double Variacion_porcentual(double valor_ini,double valor_fin){
18:     double resultado;
19:     resultado = abs(100*((valor_ini - valor_fin)/valor_ini));
20:     return resultado;
21: }
22:
23: double Elasticidad(double precio_ini,double precio_fin, double demanda_ini,double demanda_fin){
24:     double vp_demanda, vp_precio, elasticidad;
25:     vp_precio = Variacion_porcentual(precio_ini,precio_fin);
26:     vp_demanda = Variacion_porcentual(demanda_ini,demanda_fin);
27:     elasticidad = vp_demanda / vp_precio;
28:     return elasticidad;
29: }
30: //////////////////////////////////////
31:
32:
33: int main(){
34:     double precio_ini, precio_fin;
35:     double demanda_ini, demanda_fin;
36:     double elast_pd;
37:
38:     cout << "Cálculo de la Elasticidad Precio-Demanda.\n\n"
39:     << "Introduzca cuaternas de valores:"
40:     << "El precio inicial, el precio final, la demanda inicial y la demanda final.\n"
41:     << "Introduzca cualquier negativo en el precio inicial para terminar\n\n";
42:
43:     cin >> precio_ini;
44:
45:     while (precio_ini >= 0){
46:         cin >> precio_fin;
47:         cin >> demanda_ini;
48:         cin >> demanda_fin;
49:
50:         elast_pd = Elasticidad(precio_ini,precio_fin,demanda_ini,demanda_fin);
51:
52:         cout << "Elasticidad Precio-Demanda: " << elast_pd << "\n";
53:         cin >> precio_ini;
54:     }
55: }

```

```
1: //////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////
9: /*
10: int ValorAbsoluto(int entero){
11:     if (entero < 0)
12:         entero = -entero;
13:     else
14:         return entero;
15: }
16: En esta primera función el error está en poner el
17: return entero dentro de un else ya que si el entero
18: es negativo, no va a devolver nada
19:
20: bool EsPositivo(int valor){
21:     if (valor > 0)
22:         return true;
23: }
24: Le estamos dando un valor int, nos debe devolver un
25: valor int, no un true/false como si fuera un boolean
26:
27: long ParteEntera(double real){
28:     int parte_entera;
29:
30:     parte_entera = trunc(real);
31:     return parte_entera;
32: }
33:
34: El int parte_entera es totalmente innecesario ya que es
35: justamente lo que hace el trunc. O quitas el trunc y haces
36: el casting automatico double a int o modificas el double real
37: */
```

```

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: #include <cmath>
12:
13: using namespace std;
14:
15: int Max2(int uno,int otro){
16:     if(uno > otro)
17:         return uno;
18:     else
19:         return otro;
20: }
21:
22: int Max3(int uno,int otro,int otro_mas){
23:     int maximo = Max2(uno,otro);
24:     if( maximo > otro_mas)
25:         return maximo;
26:     else
27:         return otro_mas;
28: }
29:
30: int main(){
31:     int entero1,entero2,entero3;
32:     int maximo;
33:
34:     cin >> entero1 >> entero2 >> entero3;
35:
36:     maximo = Max2(entero1,entero2);
37:     maximo = Max3(entero1,entero2,entero3);
38:
39:     cout << maximo << endl;
40: }
41:

```

```

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 main(){
14:     const int MIN_TEMP = -90, MAX_TEMP = 60;
15:     double anterior, actual;
16:     int secuencia = 1;           //Se va a repetir al menos 1 vez
17:     double mayor_secuencia = 0;
18:     double pos_secuencia = 1;
19:     int pos_mayor_secuencia = 1;
20:     int contador = 1;           //Empezamos desde 1 para que sea "lógico" y no empezar desde el 0 como
los vectores
21:     bool final_entrada_datos=false;
22:
23:     cin >> anterior;
24:     final_entrada_datos=anterior < MIN_TEMP
25:         ||
26:         anterior > MAX_TEMP;
27:
28:     while(!final_entrada_datos){
29:         cin >> actual;
30:         if(actual > anterior){
31:             secuencia++;
32:         }
33:         else if(secuencia > mayor_secuencia){
34:             mayor_secuencia = secuencia;
35:             pos_mayor_secuencia = pos_secuencia;
36:             pos_secuencia = contador +1;       //Contador +1 ya que contamos apartir del siguiente
37:             secuencia = 1;
38:         }
39:         if(actual < MIN_TEMP || actual > MAX_TEMP){
40:             if(secuencia > mayor_secuencia){
41:                 mayor_secuencia = secuencia -1;    // -1 ya que no contamos al "terminador"
42:                 pos_mayor_secuencia = pos_secuencia;
43:             }
44:             final_entrada_datos=true;
45:         }
46:         anterior = actual;
47:         contador++;
48:     }
49:
50:
51:     cout << "Inicio: " << pos_mayor_secuencia << " Longitud: " << mayor_secuencia << endl;
52:
53:     return 0;
54:
55: }
56:
57:

```



```

1: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2: //
3: // Fundamentos de Programación
4: // ETS Informática y Telecomunicaciones
5: // Universidad de Granada
6: // // Don Oreó
7: //
8: //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
9:
10:
11: // Top K (versión eficiente)
12:
13: #include <iostream>
14: using namespace std;
15:
16: int main () {
17:     const int TERMINADOR = -1;
18:     int dato;
19:     const int TAMANIO = 1e+6;
20:     int vector[TAMANIO], topk[TAMANIO];
21:     int utilizados_vector, maximo, posicion_maximo, guardar, izda, i, k;
22:
23:     cout << "Topk.\n\n"
24:         << "Introduzca enteros con terminador "
25:         << TERMINADOR << "\n"
26:         << "Luego introduzca el valor de k.\n\n";
27:
28:     //Entrada de Datos
29:
30:     utilizados_vector = 0;
31:     cin >> dato;
32:
33:     while (dato != TERMINADOR && utilizados_vector < TAMANIO) {
34:         vector[utilizados_vector] = dato;
35:         utilizados_vector++;
36:         cin >> dato;
37:     }
38:
39:     cin >> k;
40:
41:     //computo
42:
43:     for (i = 0; i < utilizados_vector; i++)
44:         topk[i] = vector[i];
45:
46:     for (izda = 0 ; izda < k ; izda++){
47:         maximo = topk[izda];
48:         for (i = izda + 1 ; i < utilizados_vector ; i++){
49:             if (topk[i] > maximo){
50:                 maximo = topk[i];
51:                 posicion_maximo = i;
52:             }
53:         }
54:
55:         guardar = topk[izda];
56:         topk[izda] = maximo;
57:         topk[posicion_maximo] = guardar;
58:     }
59:
60:     //Salida de Datos
61:
62:     for (int i = 0; i < k; i++){
63:         cout << topk[i] << " ";
64:     }
65: }

```