

```

1  /*
2  * Secuencia inicial y final
3  * Pablo Villa 18/10/23
4  *
5  */
6  #include <iostream>
7
8  using namespace std;
9  const int FIN_SECUENCIA = 0;
10
11 int main()
12 {
13
14     int buscado = 0;
15     int numero_actual = 0;
16     int posicion_inicial = 0;
17     int posicion_final = 0;
18     int posicion = 0;
19     bool encontrado = false;
20
21     cout << "Introduce el buscado: ";
22     cin >> buscado;
23
24     cout << "Introduce secuencia: ";
25     cin >> numero_actual;
26
27     if(numero_actual == 0){
28         cout << "Secuencia vacia" << endl;
29     }
30     while(numero_actual != FIN_SECUENCIA){
31         posicion++;
32         if(numero_actual == buscado){
33             if(posicion_inicial == 0){
34                 posicion_inicial = posicion;
35                 encontrado = true;
36             }
37             posicion_final = posicion;
38         }
39         cin >> numero_actual;
40     }
41     if (encontrado){
42         cout << "Posicion inicial: " << posicion_inicial
43             << " Posicion final " << posicion_final;
44     }
45 }
46 else {
47     cout << "Numero no encontrado" << endl;
48 }
49 }
50 }

```

```

1  /*
2  *
3  *EJ2_p4 Pablo Villa
4  * 18/10/23
5  */
6
7  #include <iostream>
8  #include <iomanip>
9  #include <cmath>
10
11 using namespace std;
12 const int NUM_DECIMALES = 3;
13 const int ANCHO_MES = 5;
14 const int ANCHO_INICIAL = 10;
15 const int ANCHO_CUOTA = 12;
16
17 int main() {
18     double capital = 0.0;
19     double interes_anual = 0.0;
20     double meses = 0.0;
21     double interes_mensual = 0.0;
22     double cuota = 0.0;
23     double saldo_restante = 0.0;
24     double interes = 0.0;
25     double amortizado = 0.0;
26
27     cout << setprecision(NUM_DECIMALES) << fixed;
28
29     cout << "Introduce capital, interes anual en tanto por ciento y meses: ";
30     cin >> capital >> interes_anual >> meses;
31
32     interes_mensual = (interes_anual / 12.0) / 100.0;
33
34     cuota = (capital * interes_mensual) / (1 - pow(1 + interes_mensual, - meses));
35
36     cout << setw(ANCHO_MES) << "Mes" << setw(ANCHO_INICIAL) << "Inicial" << setw(ANCHO_INICIAL)
37         << "Cuota" << setw(ANCHO_CUOTA) << "Intereses" << setw(ANCHO_CUOTA)
38         << "Amortizado" << setw(ANCHO_INICIAL) << "Restante" << endl;
39
40     saldo_restante = capital;
41
42     for (int i = 1; i <= meses; ++i) {
43         interes = saldo_restante * interes_mensual;
44         amortizado = cuota - interes;
45         saldo_restante = saldo_restante - amortizado;
46
47         cout << setw(ANCHO_MES) << i << setw(ANCHO_INICIAL) << capital << setw(ANCHO_INICIAL)
48             << cuota << setw(ANCHO_INICIAL) << interes << setw(ANCHO_INICIAL)
49             << amortizado << setw(ANCHO_INICIAL) << saldo_restante << endl;
50
51         capital = saldo_restante;
52     }
53
54 }

```

```

1  /*
2  * Juego de la Serpiente v1
3  * Pablo_Villa 28/09/2023
4  */
5  #include <iostream>
6  #include "terminal.h"
7
8  using namespace std;
9
10 const char TECLA_SIGUIENTE = ' ';
11 const char TECLA_FIN = 'F';
12 const char SERPIENTE = '@';
13 const char ARRIBA = 'W';
14 const char ABAJO = 'S';
15 const char IZQUIERDA = 'A';
16 const char DERECHA = 'D';
17 const char BORRAR_SERPIENTE = ' ';
18 const int RETARDO = 60;
19 const int SERPIENTE_X_INICIAL = 20;
20 const int LIMITE_SUPERIOR = 3;
21 const int LIMITE_INFERIOR = 19;
22 const int LIMITE_IZQUIERDA = 2;
23 const int LIMITE_DERECHA = 78;
24 const int SERPIENTEX_INICIAL = 10;
25 const int MOVIMIENTO_X_INICIAL = 0;
26 const int MOVIMIENTO_Y_INICIAL = 0;
27 const int MOVIMIENTO_X_DERECHA = 1;
28 const int MOVIMIENTO_Y_DESCENDENTE = 1;
29 const int MOVIMIENTO_X_IZQUIERDA = -1;
30 const int MOVIMIENTO_Y_ASCENDENTE = -1;
31
32 const string TITULO = "Juego de la serpiente ";
33 const string VERSION = "2.0";
34 const string TECLA_CONTINUAR = "ESPACIO";
35
36 int main() {
37     int serpiente_x = 0;
38     int serpiente_y = 0;
39     int movimiento_x = 0;
40     int movimiento_y = 0;
41
42     serpiente_x = SERPIENTE_X_INICIAL;
43     serpiente_y = SERPIENTEX_INICIAL;
44     movimiento_x = MOVIMIENTO_X_INICIAL;
45     movimiento_y = MOVIMIENTO_Y_INICIAL;
46
47     char tecla = '\0';
48     retardar(RETARDO);
49     hacer_cursor_visible(false);
50     poner_cursor(1,1);
51     cout << " ***** " << endl;
52
53     poner_cursor(1,2);
54     cout << " * " << TITULO << VERSION << " * " << endl;
55
56     poner_cursor(1,3);
57     cout << " ***** " << endl;
58
59     poner_cursor(1,6);
60     cout << "   _____   " << endl;
61
62     poner_cursor(1,7);
63     cout << " _/           \\ " << endl;
64
65     poner_cursor(1,8);
66     cout << "  \\_____  \\ " << endl;

```

```

67
68 poner_cursor(1,9);
69 cout << "      \\      \\_____" << endl;
70
71 poner_cursor(1,10);
72 cout << "      \\      \\ " << endl;
73
74 poner_cursor(1,11);
75 cout << "      \\____      \\____      _|_ " << endl;
76
77 poner_cursor(1,12);
78 cout << "      \\      0 \\_ /  /  \\ " << endl;
79
80 poner_cursor(1,13);
81 cout << "      \\_____/ \\  \\_/" << endl;
82
83 poner_cursor(1,17);
84 cout << "Pulsa la tecla de " <<  TECLA_CONTINUAR << " para continuar" << endl;
85
86 while(Leer_tecla() != TECLA_SIGUIENTE){
87
88     retardar(RETARDO);
89 }
90
91 deshabilitar_modos_crudo_terminal();
92 borrar_terminal();
93
94 poner_cursor(2,1);
95 cout << " +-----" << TITULO
96     << VERSION << " -----+ " << endl;
97
98 poner_cursor(2,2);
99 cout << " | " << endl;
100 poner_cursor(2,3);
101 cout << " | " << endl;
102 poner_cursor(2,4);
103 cout << " | " << endl;
104 poner_cursor(2,5);
105 cout << " | " << endl;
106 poner_cursor(2,6);
107 cout << " | " << endl;
108 poner_cursor(2,7);
109 cout << " | " << endl;
110 poner_cursor(2,8);
111 cout << " | " << endl;
112 poner_cursor(2,9);
113 cout << " | " << endl;
114 poner_cursor(2,10);
115 cout << " | " << endl;
116 poner_cursor(2,11);
117 cout << " | " << endl;
118 poner_cursor(2,12);
119 cout << " | " << endl;
120 poner_cursor(2,13);
121 cout << " | " << endl;
122 poner_cursor (2,14);
123 cout << " | " << endl;
124 poner_cursor(2,15);
125 cout << " | " << endl;
126 poner_cursor(2,16);
127 cout << " | " << endl;
128 poner_cursor(2,17);
129 cout << " | " << endl;
130 poner_cursor(2,18);
131 cout << " | " << endl;
132 poner_cursor(2,19);
133 cout << " | " << endl;

```

```

133     poner_cursor(2,20);
134     cout << "+-----+ " << endl;
135     poner_cursor(2, 21);
136     cout << ARRIBA << "-> Subir " << ABAJO << "-> Bajar " << IZQUIERDA
137         << "-> Izda " << DERECHA << "-> Dcha " << TECLA_FIN << "-> Fin" << endl;
138
139     habilitar_modocru_terminal();
140     hacer_cursor_visible(false);
141     tecla = leer_tecla();
142
143     while (tecla != TECLA_FIN) {
144
145         poner_cursor(serpiente_x,serpiente_y);
146         cout << SERPIENTE;
147         retardar(RETARDO);
148         poner_cursor(serpiente_x,serpiente_y);
149         cout << BORRAR_SERPIENTE;
150
151         if (serpiente_x == LIMITE_DERECHA || serpiente_x == LIMITE_IZQUIERDA ||
152             serpiente_y == LIMITE_INFERIOR || serpiente_y == LIMITE_SUPERIOR) {
153
154             break;
155         }
156         switch (toupper(tecla)) {
157             case ARRIBA:
158                 movimiento_x = MOVIMIENTO_X_INICIAL;
159                 movimiento_y = MOVIMIENTO_Y_ASCENDENTE;
160                 break;
161
162             case ABAJO:
163                 movimiento_x = MOVIMIENTO_X_INICIAL;
164                 movimiento_y = MOVIMIENTO_Y_DESCENDENTE;
165                 break;
166
167             case IZQUIERDA:
168                 movimiento_x = MOVIMIENTO_X_IZQUIERDA;
169                 movimiento_y = MOVIMIENTO_Y_INICIAL;
170                 break;
171
172             case DERECHA:
173                 movimiento_x = MOVIMIENTO_X_DERECHA;
174                 movimiento_y = MOVIMIENTO_Y_INICIAL;
175                 break;
176
177         }
178
179         serpiente_x = serpiente_x + movimiento_x;
180         serpiente_y = serpiente_y + movimiento_y;
181         tecla = leer_tecla();
182
183     }
184     deshabilitar_modocru_terminal();
185     borrar_terminal();
186 }

```