

```

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     while(numero_actual!= FIN_SECUENCIA){
28         posicion++;
29         if(numero_actual == buscado){
30             if(posicion_inicial == 0){
31                 posicion_inicial = posicion;
32                 encontrado = true;
33             }
34             posicion_final = posicion;
35         }
36         cin >> numero_actual;
37     }
38     if (encontrado){
39         cout << "Posición inicial: " << posicion_inicial
40             << " Posición final " << posicion_final;
41     }
42 }
43 else {
44     cout << "Número no encontrado" << endl;
45 }
46 }
47 }

```

```

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 SEPARADOR_MES = 5;
14 const int SEPARADOR_INICIAL = 10;
15 const int SEPARADOR_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(SEPARADOR_MES) << "Mes" << setw(SEPARADOR_INICIAL) << "Inicial" << setw(SEPARADOR_INICIAL)
37         << "Cuota" << setw(SEPARADOR_CUOTA) << "Intereses" << setw(SEPARADOR_CUOTA)
38         << "Amortizado" << setw(SEPARADOR_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(SEPARADOR_MES) << i << setw(SEPARADOR_INICIAL) << capital << setw(SEPARADOR_INICIAL)
48             << cuota << setw(SEPARADOR_INICIAL) << interes << setw(SEPARADOR_INICIAL)
49             << amortizado << setw(SEPARADOR_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 = 2;
21 const int LIMITE_INFERIOR = 20;
22 const int LIMITE_IZQUIERDA = 2;
23 const int LIMITE_DERECHA = 78;
24 const int SERPIENTE_Y_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_SIGUIENTE = "ESPACIO";
35
36 int main() {
37     int serpiente_x = SERPIENTE_X_INICIAL;
38     int serpiente_y = SERPIENTE_Y_INICIAL;
39     int movimiento_x = MOVIMIENTO_X_INICIAL;
40     int movimiento_y = MOVIMIENTO_Y_INICIAL;
41     char tecla = '\0';
42
43     retardar(RETARDO);
44     hacer_cursor_visible(false);
45     poner_cursor(1,1);
46     cout << " ***** " << endl;
47
48     poner_cursor(1,2);
49     cout << " * "<< TITULO << VERSION << " * " << endl;
50
51     poner_cursor(1,3);
52     cout << " ***** " << endl;
53
54     poner_cursor(1,6);
55     cout << "   _____" << endl;
56
57     poner_cursor(1,7);
58     cout << " _/           \\" << endl;
59
60     poner_cursor(1,8);
61     cout << "  \\\_         \\" << endl;
62
63     poner_cursor(1,9);
64     cout << "           \\\  \\\_ " << endl;
65
66     poner_cursor(1,10);

```

```

67     cout << "          \\\          \\\ " << endl;
68
69     poner_cursor(1,11);
70     cout << "          \\\_          \\\_          _|_ " << endl;
71
72     poner_cursor(1,12);
73     cout << "          \\\          0 \\\_ / \\\ " << endl;
74
75     poner_cursor(1,13);
76     cout << "          \\\_/_ / \\\ \\\_/" << endl;
77
78     poner_cursor(1,17);
79     cout << "Pulsa la tecla de " << TECLA_SIGUIENTE << " para continuar" << endl;
80
81     while(Leer_tecla() != TECLA_SIGUIENTE){
82
83         retardar(RETARDO);
84     }
85
86     deshabilitar_modos_crudo_terminal();
87     borrar_terminal();
88
89     poner_cursor(2,1);
90     cout << " +-----+ " << TITULO
91         << VERSION << " -----+ " << endl;
92     poner_cursor(2,2);
93     cout << "| " << endl;
94     poner_cursor(2,3);
95     cout << "| " << endl;
96     poner_cursor(2,4);
97     cout << "| " << endl;
98     poner_cursor(2,5);
99     cout << "| " << endl;
100    poner_cursor(2,6);
101    cout << "| " << endl;
102    poner_cursor(2,7);
103    cout << "| " << endl;
104    poner_cursor(2,8);
105    cout << "| " << endl;
106    poner_cursor(2,9);
107    cout << "| " << endl;
108    poner_cursor(2,10);
109    cout << "| " << endl;
110    poner_cursor(2,11);
111    cout << "| " << endl;
112    poner_cursor(2,12);
113    cout << "| " << endl;
114    poner_cursor(2,13);
115    cout << "| " << endl;
116    poner_cursor(2,14);
117    cout << "| " << endl;
118    poner_cursor(2,15);
119    cout << "| " << endl;
120    poner_cursor(2,16);
121    cout << "| " << endl;
122    poner_cursor(2,17);
123    cout << "| " << endl;
124    poner_cursor(2,18);
125    cout << "| " << endl;
126    poner_cursor(2,19);
127    cout << "| " << endl;
128    poner_cursor(2,20);
129    cout << " +-----+ " << endl;
130    poner_cursor(2,21);
131    cout << ARRIBA << "-> Subir " << ABAJO << "-> Bajar " << IZQUIERDA
132        << "-> Izda " << DERECHA << "-> Dcha " << TECLA_FIN << "-> Fin" << endl;

```

```

133
134     habilitar_modocruerto_terminal();
135     hacer_cursor_visible(false);
136
137     while (tecla != TECLA_FIN && serpiente_x != LIMITE_IZQUIERDA && serpiente_x != LIMITE_DERECHA &&
138           serpiente_y != LIMITE_SUPERIOR && serpiente_y != LIMITE_INFERIOR) {
139
140
141         poner_cursor(serpiente_x,serpiente_y);
142         cout << SERPIENTE;
143         retardar(RETARDO);
144         poner_cursor(serpiente_x,serpiente_y);
145         cout << BORRAR_SERPIENTE;
146
147         switch (toupper(tecla)) {
148             case ARRIBA:
149                 movimiento_x = MOVIMIENTO_X_INICIAL;
150                 movimiento_y = MOVIMIENTO_Y_ASCENDENTE;
151                 break;
152
153             case ABAJO:
154                 movimiento_x = MOVIMIENTO_X_INICIAL;
155                 movimiento_y = MOVIMIENTO_Y_DESCENDENTE;
156                 break;
157
158             case IZQUIERDA:
159                 movimiento_x = MOVIMIENTO_X_IZQUIERDA;
160                 movimiento_y = MOVIMIENTO_Y_INICIAL;
161                 break;
162
163             case DERECHA:
164                 movimiento_x = MOVIMIENTO_X_DERECHA;
165                 movimiento_y = MOVIMIENTO_Y_INICIAL;
166                 break;
167
168         }
169
170         serpiente_x = serpiente_x + movimiento_x;
171         serpiente_y = serpiente_y + movimiento_y;
172         tecla = leer_tecla();
173
174     }
175     deshabilitar_modocruerto_terminal();
176     borrar_terminal();
177 }

```