

```

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     int buscado = 0;
14     int numero_actual = 0;
15     int posicion_inicial = 0;
16     int posicion_final = 0;
17     int posicion = 0;
18     bool encontrado = false;
19
20     cout << "Introduce el buscado: ";
21     cin >> buscado;
22     cout << "Introduce secuencia: ";
23     cin >> numero_actual;
24
25     while(numero_actual != FIN_SECUENCIA){
26         posicion++;
27         if(numero_actual == buscado){
28             if(posicion_inicial == 0){
29                 posicion_inicial = posicion;
30                 encontrado = true;
31             }
32             posicion_final = posicion;
33         }
34         cin >> numero_actual;
35     }
36     if (encontrado){
37         cout << "Posición inicial: " << posicion_inicial
38             << " Posición final " << posicion_final;
39     }
40     else {
41         cout << "Número no encontrado" << endl;
42     }
43 }

```

```

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 TABULADOR_MES = 5;
14 const int TABULADOR_INICIAL = 10;
15 const int TABULADOR_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(TABULADOR_MES) << "Mes" << setw(TABULADOR_INICIAL) << "Inicial" << setw(TABULADOR_INICIAL)
37         << "Cuota" << setw(TABULADOR_CUOTA) << "Intereses" << setw(TABULADOR_CUOTA)
38         << "Amortizado" << setw(TABULADOR_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(TABULADOR_MES) << i << setw(TABULADOR_INICIAL) << capital << setw(TABULADOR_INICIAL)
48             << cuota << setw(TABULADOR_INICIAL) << interes << setw(TABULADOR_INICIAL)
49             << amortizado << setw(TABULADOR_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 int RETARDO = 60;
18 const int SERPIENTE_X_INICIAL = 20;
19 const int LIMITE_SUPERIOR = 2;
20 const int LIMITE_INFERIOR = 20;
21 const int LIMITE_IZQUIERDA = 2;
22 const int LIMITE_DERECHA = 78;
23 const int SERPIENTE_Y_INICIAL = 10;
24 const int MOVIMIENTO_X_DERECHA = 1;
25 const int MOVIMIENTO_Y_DESCENDENTE = 1;
26 const int MOVIMIENTO_X_IZQUIERDA = -1;
27 const int MOVIMIENTO_Y_ASCENDENTE = -1;
28 const string TITULO = "Juego de la serpiente ";
29 const string VERSION = "2.0";
30 const string SIGUIENTE_PANTALLA = "ESPACIO";
31
32 int main() {
33     int serpiente_x = SERPIENTE_X_INICIAL;
34     int serpiente_y = SERPIENTE_Y_INICIAL;
35     int movimiento_x = 0;
36     int movimiento_y = 0;
37     char tecla = '\0';
38
39     retardar(RETARDO);
40     hacer_cursor_visible(false);
41     habilitar_modo_crudo_terminal();
42
43     poner_cursor(1,1);
44     cout << " ***** " << endl;
45
46     poner_cursor(1,2);
47     cout << " * "<< TITULO << VERSION << " * " << endl;
48
49     poner_cursor(1,3);
50     cout << " ***** " << endl;
51
52     poner_cursor(1,6);
53     cout << "   _____ " << endl;
54
55     poner_cursor(1,7);
56     cout << " _/           \\" << endl;
57
58     poner_cursor(1,8);
59     cout << "  \\\_         \\" << endl;
60
61     poner_cursor(1,9);
62     cout << "           \\\   \\\_ " << endl;
63
64     poner_cursor(1,10);
65     cout << "           \\\           \\" << endl;
66

```

```

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

```

```

133
134 while (toupper(tecla) != TECLA_FIN &&
135         serpiente_x != LIMITE_IZQUIERDA && serpiente_x != LIMITE_DERECHA &&
136         serpiente_y != LIMITE_SUPERIOR && serpiente_y != LIMITE_INFERIOR) {
137
138     poner_cursor(serpiente_x,serpiente_y);
139     cout << SERPIENTE;
140
141     retardar(RETARDO);
142
143     poner_cursor(serpiente_x,serpiente_y);
144     cout << " ";
145
146     switch (toupper(tecla)) {
147         case ARRIBA:
148             movimiento_x = 0;
149             movimiento_y = MOVIMIENTO_Y_ASCENDENTE;
150             break;
151
152         case ABAJO:
153             movimiento_x = 0;
154             movimiento_y = MOVIMIENTO_Y_DESCENDENTE;
155             break;
156
157         case IZQUIERDA:
158             movimiento_x = MOVIMIENTO_X_IZQUIERDA;
159             movimiento_y = 0;
160             break;
161
162         case DERECHA:
163             movimiento_x = MOVIMIENTO_X_DERECHA;
164             movimiento_y = 0;
165             break;
166     }
167
168     serpiente_x = serpiente_x + movimiento_x;
169     serpiente_y = serpiente_y + movimiento_y;
170
171     tecla = leer_tecla();
172 }
173
174 deshabilitar_modos_crudo_terminal();
175 borrar_terminal();
176 }

```