

```

1:  /*
2:     INSTITUTO POLITÉCNICO NACIONAL
3:     ESCOM
4:     Alumno...
5:  */
6:  #include <stdio.h>
7:  #include <stdlib.h>
8:  #include <time.h>
9:  #define TAM 1000
10:
11: void principal();
12: void generaArreglo(int *A, int n);
13: void imprimeArreglo(int *A, int n);
14: int busquedaSecuencial(int *A, int n, int x, int *ct);
15: int sumaArreglo(int *A, int n, int *ct);
16: void insertionSort(int *A, int n, int *ct);
17:
18: main()
19: {
20:     principal();
21:     return 0;
22: }
23:
24: void principal()
25: {
26:     FILE *pf=fopen("busqSec.csv","at");
27:
28:     int Arr[TAM];
29:     int tam=10;
30:     int ct=0;
31:
32:     int i=0;
33:     int pts=20;
34:
35:     for(i=0; i<pts; i++)
36:     {
37:         ct=0;
38:         generaArreglo(Arr,tam);
39:         imprimeArreglo(Arr,tam);
40:         insertionSort(Arr,tam,&ct);
41:         printf("\n");
42:         imprimeArreglo(Arr,tam);
43:
44:         printf("\n-----\n");
45:
46:         // srand(time(NULL));
47:         // int s=sumaArreglo(Arr,tam,&ct);
48:         // int ind=rand()%tam;
49:         // int x=Arr[ind];
50:         // int b=busquedaSecuencial(Arr,tam,x,&ct);
51:         // printf("\n--> %d <--", b);
52:         printf("\n Num. Pasos: %d",ct);
53:         fprintf(pf,"%d,%d\n",tam,ct);
54:         tam+=10;
55:     }
56:
57:
58:
59:
60:
61:     fclose(pf);
62: }
63:
64: void generaArreglo(int *A, int n)
65: {
66:     srand(time(NULL));
67:     int i;
68:     for(i=0; i<n; i++)
69:         A[i]=rand()%(n+10);

```

```

70: }
71:
72: void imprimeArreglo(int *A, int n)
73: {
74:     int i;
75:     for(i=0; i<n; i++)
76:         printf("[ %d ]", A[i]);
77: }
78:
79: int busquedaSecuencial(int *A, int n, int x, int *ct)
80: {
81:     int i; (*ct)++;
82:
83:     (*ct)++;
84:     for(i=0; i<n; i++)
85:     {
86:         (*ct)++;
87:         if(A[i]==x)
88:         {
89:             (*ct)++;
90:             (*ct)++;
91:             return i;
92:         }
93:         (*ct)++;
94:         (*ct)++;
95:     }
96:     (*ct)++;
97:     (*ct)++;
98:     return -1;
99: }
100:
101: int sumaArreglo(int *A, int n, int *ct)
102: {
103:     int s=0; (*ct)++;
104:     int i; (*ct)++;
105:
106:     (*ct)++;
107:     for(i=0; i<n; i++)
108:     {
109:         (*ct)++;
110:         s+=A[i]; (*ct)++;
111:         (*ct)++;
112:     }
113:     (*ct)++;
114:     (*ct)++;
115:     return s;
116: }
117:
118:
119: void insertionSort(int *A, int n, int *ct)
120: {
121:     int i; (*ct)++;
122:     int j; (*ct)++;
123:     int key; (*ct)++;
124:
125:     (*ct)++;
126:     for(i=1; i<n; i++)
127:     {
128:         (*ct)++;
129:         key=A[i]; (*ct)++;
130:         j=i-1; (*ct)++;
131:
132:         while(j>=0 && A[j]>key)
133:         {
134:             (*ct)++;
135:             (*ct)++;
136:             (*ct)++;
137:             A[j+1]=A[j]; (*ct)++;
138:             j--; (*ct)++;

```

```
139:         }
140:         (*ct)++;
141:         //      (*ct)++;
142:         /*      if(j<0)
143:             (*ct)++;
144:         else
145:             {
146:                 (*ct)++;
147:                 (*ct)++;
148:             }
149:         */
150:
151:
152:         A[j+1]=key; (*ct)++;
153:     }
154: }
```