

```

1  public class Main {
2      public static void main(String[] args) {
3          Array Elenco = new Array();
4          final int max = 10;
5          int[] vett = new int[max];
6          int Trovato;
7
8          System.out.println("attraversa Array VUOTO -----");
9          Elenco.attraversaArray(vett);
10         attraversa Array VUOTO -----
11             0 0
12             1 0
13             2 0
14             3 0
15             4 0
16             5 0
17             6 0
18             7 0
19             8 0
20             9 0
21
22         System.out.println("popola Array e visualizza -----");
23         Elenco.popolaArray(vett,5,10);
24         Elenco.attraversaArray(vett);
25         popola Array e visualizza -----
26             0 10
27             1 20
28             2 30
29             3 40
30             4 50
31             5 0
32             6 0
33             7 0
34             8 0
35             9 0
36
37         System.out.println("inserisce Item 22 in 3 in Array e visualizza-----");
38         Elenco.inserisceItemArray(vett,3,22);
39         Elenco.attraversaArray(vett);
40         inserisce Item 22 in 3 in Array e visualizza-----
41         Prima dell'inserimento il primo elemento LIBERO si trovava in posizione: 5
42             0 10
43             1 20
44             2 30
45             3 22
46             4 40
47             5 50
48             6 0
49             7 0
50             8 0
51             9 0
52
53         System.out.println("modifica Item in 3 con 23 in Array e visualizza
54         -----");
55         Elenco.modificaItemArray(vett,3,23);
56         Elenco.attraversaArray(vett);
57         modifica Item in 3 con 23 in Array e visualizza -----
58             0 10
59             1 20
60             2 30
61             3 23
62             4 40
63             5 50
64             6 0
65             7 0
66             8 0
67             9 0
68
69
70
71
72

```

```
73 System.out.println("ricerca LINEARE di Item 23 in Array e visualizza ");
74 Elenco.ricercaItemArrayL(vett,23);
75 Elenco.attraversaArray(vett);
76     ricerca LINEARE di Item 23 in Array e visualizza
77     Elemento trovato in posizione: 3
78     0 10
79     1 20
80     2 30
81     3 23
82     4 40
83     5 50
84     6 0
85     7 0
86     8 0
87     9 0
88
89 System.out.println("ricerca BINARIA di Item 23 in Array e visualizza ");
90 Elenco.ricercaItemArrayB(vett,23);
91 Elenco.attraversaArray(vett);
92     ricerca BINARIA di Item 23 in Array e visualizza
93     NON TROVATO
94     0 10
95     1 20
96     2 30
97     3 23
98     4 40
99     5 50
100    6 0
101    7 0
102    8 0
103    9 0
104
105 System.out.println("elimina Item in 3 in Array e visualizza -----");
106 Elenco.eliminaItemArray(vett,3);
107 Elenco.attraversaArray(vett);
108     elimina Item in 3 in Array e visualizza -----
109     0 10
110     1 20
111     2 30
112     3 40
113     4 50
114     5 0
115     6 0
116     7 0
117     8 0
118     9 0
119
120 System.out.println("ricerca BINARIA di Item 30 in Array e visualizza");
121 Elenco.ricercaItemArrayB(vett,30);
122 Elenco.attraversaArray(vett);
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
```

```
146
147     System.out.println("ricerca BINARIA di Item 30 in Array e restituzione del
148     risultato");
149     Trovato = Elenco.ricercaItemArrayBr(vett,30);
150     if (Trovato > 0) {
151         //TROVATO
152         System.out.println("TROVATO ALLA LOCAZIONE " + Trovato);
153     }
154     else {
155         //Ricerca infruttuosa
156         System.out.println("NON TROVATO");
157     }
158     System.out.println("");
159     ricerca BINARIA di Item 30 in Array e visualizza
160     trovato in posizione 2
161     0 10
162     1 20
163     2 30
164     3 40
165     4 50
166     5 0
167     6 0
168     7 0
169     8 0
170     9 0
171 }
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
```

```

218 class Array {
219     public void popolaArray (int[] vett, int NumItem, int base) {
220         for(int i=1; i<=NumItem; i++)
221             vett[i-1] = i*base;
222     }
223
224     public void attraversaArray (int[] vett) {
225         for(int i=0; i<vett.length; i++)
226             System.out.println(i + " " + vett[i]);
227     }
228
229     public void inserisceItemArray (int[] vett,int k,int item) {
230         int Libero = 0;
231         int n = vett.length;
232         for (int i = 0; i <= n-1; i++) {
233             if (vett[i] == 0) {
234                 System.out.println("Prima dell'inserimento il primo elemento LIBERO
235                 si trovava in posizione: " + i);
236                 Libero = i;
237                 break;
238             }
239         }
240         if (Libero>0) {
241             if (k>Libero) k=Libero;
242             int i = n-2;
243             while(i >= k) {
244                 vett[i+1] = vett[i];
245                 i--;
246             }
247             vett[k] = item;
248         }
249         else
250             System.out.println("Non ci sono elementi liberi");
251     }
252
253     public void modificaItemArray (int[] vett,int k,int item) {
254         vett[k] = item;
255     }
256
257     public void eliminaItemArray (int[] vett,int k) {
258         int i;
259         for(i=k; i < vett.length-1; i++)
260             vett[i] = vett[i+1];
261     }
262
263     public void ricercaItemArrayL (int[] vett,int item) {
264         int loc;
265         int Trovato = -1;
266         for(loc=0; loc<vett.length; loc++) {
267             if(vett[loc] == item) {
268                 System.out.println("Elemento trovato in posizione: " + loc);
269                 Trovato=1;
270                 break;
271             }
272         }
273         if (Trovato == -1) {
274             System.out.println("Elemento non trovato.");
275         }
276     }
277
278     int ricercaItemArrayLr (int[] vett,int item) {
279         int loc;
280         int Trovato = -1;
281         for(loc=0; loc<vett.length; loc++) {
282             if(vett[loc] == item) {
283                 Trovato=loc;
284                 break;
285             }
286         }
287         return Trovato;
288     }
289 }

```

```

290
291 public void ricercaItemArrayB (int[] vett,int item) {
292     int start = 0;
293     int end = vett.length-1;
294     int half = (start+end)/2;
295     int loc1;
296
297     while(start <= end && vett[half] != item) {
298         if(item < vett[half])
299             end = half-1;
300         else
301             start = half+1;
302         half = (start+end)/2;
303     }
304     if(vett[half] == item) {
305         loc1 = half;
306         System.out.println("trovato in posizione " + loc1);
307     }
308     else {
309         loc1 = 0;
310         System.out.println("NON TROVATO");
311     }
312 }
313
314 int ricercaItemArrayBr (int[] vett,int item) {
315     int start = 0;
316     int end = vett.length-1;
317     int half = (start+end)/2;
318     int loc1;
319
320     while(start <= end && vett[half] != item) {
321         if(item < vett[half])
322             end = half-1;
323         else
324             start = half+1;
325         half = (start+end)/2;
326     }
327     if(vett[half] == item) {
328         loc1 = half;
329     }
330     else {
331         loc1 = 0;
332     }
333     return loc1;
334 }
335 }
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362

```

```
363
364 OUTPUT:
365 attraversa Array VUOTO -----
366 0 0
367 1 0
368 2 0
369 3 0
370 4 0
371 5 0
372 6 0
373 7 0
374 8 0
375 9 0
376 popola Array e visualizza -----
377 0 10
378 1 20
379 2 30
380 3 40
381 4 50
382 5 0
383 6 0
384 7 0
385 8 0
386 9 0
387 inserisce Item 22 in 3 in Array e visualizza-----
388 Prima dell'inserimento il primo elemento LIBERO si trovava in posizione: 5
389 0 10
390 1 20
391 2 30
392 3 22
393 4 40
394 5 50
395 6 0
396 7 0
397 8 0
398 9 0
399 modifica Item in 3 con 23 in Array e visualizza -----
400 0 10
401 1 20
402 2 30
403 3 23
404 4 40
405 5 50
406 6 0
407 7 0
408 8 0
409 9 0
410 ricerca LINEARE di Item 23 in Array e visualizza
411 Elemento trovato in posizione: 3
412 0 10
413 1 20
414 2 30
415 3 23
416 4 40
417 5 50
418 6 0
419 7 0
420 8 0
421 9 0
422 ricerca BINARIA di Item 23 in Array e visualizza
423 NON TROVATO
424 0 10
425 1 20
426 2 30
427 3 23
428 4 40
429 5 50
430 6 0
431 7 0
432 8 0
433 9 0
434
435
```

```
436
437  elimina Item in 3 in Array e visualizza -----
438  0 10
439  1 20
440  2 30
441  3 40
442  4 50
443  5 0
444  6 0
445  7 0
446  8 0
447  9 0
448  ricerca BINARIA di Item 30 in Array e visualizza
449  trovato in posizione 2
450  0 10
451  1 20
452  2 30
453  3 40
454  4 50
455  5 0
456  6 0
457  7 0
458  8 0
459  9 0
460  ricerca BINARIA di Item 30 in Array e restituzione del risultato
461  TROVATO ALLA LOCAZIONE 2
462
463
```