

```

1  #include <iostream>
2  #include <limit.h>
3
4  using namespace std;
5
6  /*
7   Giovanni Ronchi 3BI
8
9   Gestire una squadra di calcio composta da n calciatori (nome,
cognome, ruolo, altezza >1,60 e anno di nascita >=1995). Realizzare un
programma che:
10  . Dati input nome e cognome restituisca ruolo del calciatore.
11  . Visualizzare i nomi dei calciatori con altezza max
12  . Visualizzare i calciatori con altezza compresa tra 1,80 e 1,90
13  . Visualizzare il calciatore con eta minore
14  . Visualizzare elenco alfabetico dei calciatori
15
16  descrizioni costanti
17  nmax, numero massimo di calciatori che il programma può inserire,
intero
18
19  descrizione variabili globali
20  nome, nome calciatore, array di nmax stringhe
21  cognome, cognome calciatore, array di nmax stringhe
22  ruolo, ruolo calciatore, array di nmax stringhe
23
24  pseudo
25
26  procedura Inserimento
27  descriz var locali
28  pos, variabile per ricerca di 1 elemento, intero
29
30  inizio
31  leggi cercaNome e controlla che sia diverso da stringa vuota
32  leggi cercaCognome e controlla che sia diverso da stringa vuota
33  pos=-1;
34  ricerca cercaNome e cercaCognome nei corrispettivi vettori nome e
cognome e tornami in pos la sua posizione oppure -1
35  se pos== -1
36  allora
37  scrivi calciatore non trovato
38  fse
39  fine
40
41  procedura RicMax
42  var locali:
43  cercaNome, nome da trovare, stringa
44  cercaCognome, cognome da trovare, stringa
45  temp, variabile per scambiare contenuto variabili nome e cognome,
stringa
46  altezza, altezza calciatore, array di nmax interi
47  anno, anno di nascita calciatore, array di nmax interi
48  scegli, variabile per scegliere che caso analizzare del menu, intero
49  maxAltezza, altezza massima calciatori, intero
50  minEta, anno più vecchio dei calciatori (quindi eta più giovane), intero
51
52  inizio
53  ricerca elemento massimo in altezza e metti in maxAltezza il suo
valore
54  i=0
55  mentre i<n
56  se altezza[i]==maxAltezza
57  allora
58  scrivi nome[i]
59  fse
60  i++

```

```

61     fciclo
62 fine
63
64 punto 3
65 inizio
66     i=0
67     mentre i<n
68         se altezza[i]>=180 && altezza[i]<=190
69             allora
70                 scrivi nome[i], cognome[i], ruolo[i]
71             fse
72             i++
73     fciclo
74 fine
75
76 punto 4
77 inizio
78     ricerca elemento minimo in anno e metti in minEta il suo valore
79     i=0
80     mentre i<n
81         se anno[i]==minEta
82             allora
83                 scrivi nome[i], cognome[i], ruolo[i]
84             fse
85             i++
86     fciclo
87 fine
88
89 punto 5
90 inizio
91     ordina il vettore nome e cognome in ordine alfabetico
92     stampa tutti i dati presenti in cognome e nome
93 fine
94 */
95 const int nmax=30;
96 string nome[nmax], cognome[nmax], ruolo[nmax];
97 int altezza[nmax], anno[nmax];
98
99 int n;
100 float leggiFloat( float vmin, ...){
101
102 }
103 string leggiString(string mess){
104     string aus;
105
106     do{
107
108     }while (aus!="");
109     return aus;
110 }
111 int leggiInt(int vmin,int vmax, string msg){
112     int num;
113     do{
114         cout<<msg<<endl;
115         cin>>num;
116         if(num>vmax || num<vmin)
117             cout<<"numero non valido"<<endl;
118     }while(num>vmax || num<vmin);
119     return num;
120 }
121
122
123
124
125
126 void Inserimento(){

```

```

1 27
2 78
3 79
4 80
5 81
6 82
7 83
8 84
9 85
10 86
11 87
12 88
13 89
14 90
15 91
16 92
17 93
18 94
19 95
20 96
21 97
22 98
23 99
24 100
25 101
26 102
27 103
28 104
29 105
30 106
31 107
32 108
33 109
34 110
35 111
36 112
37 113
38 114
39 115
40 116
41 117
42 118
43 119
44 120
45 121
46 122
47 123
48 124
49 125
50 126
51 127
52 128
53 129
54 130
55 131
56 132
57 133
58 134
59 135
60 136
61 137
62 138
63 139
64 140
65 141
66 142
67 143
68 144
69 145
70 146
71 147
72 148
73 149
74 150
75 151
76 152
77 153
78 154
79 155
80 156
81 157
82 158
83 159
84 160
85 161
86 162
87 163
88 164
89 165
90 166
91 167
92 168
93 169
94 170
95 171
96 172
97 173
98 174
99 175
100 176
101 177
102 178
103 179
104 180
105 181
106 182
107 183
108 184
109 185
110 186
111 187
112 188
113 189
114 190
115 191
116 192
117 193
118 194
119 195
120 196
121 197
122 198
123 199
124 200
125 201
126 202
127 203
128 204
129 205
130 206
131 207
132 208
133 209
134 210
135 211
136 212
137 213
138 214
139 215
140 216
141 217
142 218
143 219
144 220
145 221
146 222
147 223
148 224
149 225
150 226
151 227
152 228
153 229
154 230
155 231
156 232
157 233
158 234
159 235
160 236
161 237
162 238
163 239
164 240
165 241
166 242
167 243
168 244
169 245
170 246
171 247
172 248
173 249
174 250
175 251
176 252
177 253
178 254
179 255
180 256
181 257
182 258
183 259
184 260
185 261
186 262
187 263
188 264
189 265
190 266
191 267
192 268
193 269
194 270
195 271
196 272
197 273
198 274
199 275
200 276
201 277
202 278
203 279
204 280
205 281
206 282
207 283
208 284
209 285
210 286
211 287
212 288
213 289
214 290
215 291
216 292
217 293
218 294
219 295
220 296
221 297
222 298
223 299
224 300
225 301
226 302
227 303
228 304
229 305
230 306
231 307
232 308
233 309
234 310
235 311
236 312
237 313
238 314
239 315
240 316
241 317
242 318
243 319
244 320
245 321
246 322
247 323
248 324
249 325
250 326
251 327
252 328
253 329
254 330
255 331
256 332
257 333
258 334
259 335
260 336
261 337
262 338
263 339
264 340
265 341
266 342
267 343
268 344
269 345
270 346
271 347
272 348
273 349
274 350
275 351
276 352
277 353
278 354
279 355
280 356
281 357
282 358
283 359
284 360
285 361
286 362
287 363
288 364
289 365
290 366
291 367
292 368
293 369
294 370
295 371
296 372
297 373
298 374
299 375
300 376
301 377
302 378
303 379
304 380
305 381
306 382
307 383
308 384
309 385
310 386
311 387
312 388
313 389
314 390
315 391
316 392
317 393
318 394
319 395
320 396
321 397
322 398
323 399
324 400
325 401
326 402
327 403
328 404
329 405
330 406
331 407
332 408
333 409
334 410
335 411
336 412
337 413
338 414
339 415
340 416
341 417
342 418
343 419
344 420
345 421
346 422
347 423
348 424
349 425
350 426
351 427
352 428
353 429
354 430
355 431
356 432
357 433
358 434
359 435
360 436
361 437
362 438
363 439
364 440
365 441
366 442
367 443
368 444
369 445
370 446
371 447
372 448
373 449
374 450
375 451
376 452
377 453
378 454
379 455
380 456
381 457
382 458
383 459
384 460
385 461
386 462
387 463
388 464
389 465
390 466
391 467
392 468
393 469
394 470
395 471
396 472
397 473
398 474
399 475
400 476
401 477
402 478
403 479
404 480
405 481
406 482
407 483
408 484
409 485
410 486
411 487
412 488
413 489
414 490
415 491
416 492
417 493
418 494
419 495
420 496
421 497
422 498
423 499
424 500
425 501
426 502
427 503
428 504
429 505
430 506
431 507
432 508
433 509
434 510
435 511
436 512
437 513
438 514
439 515
440 516
441 517
442 518
443 519
444 520
445 521
446 522
447 523
448 524
449 525
450 526
451 527
452 528
453 529
454 530
455 531
456 532
457 533
458 534
459 535
460 536
461 537
462 538
463 539
464 540
465 541
466 542
467 543
468 544
469 545
470 546
471 547
472 548
473 549
474 550
475 551
476 552
477 553
478 554
479 555
480 556
481 557
482 558
483 559
484 560
485 561
486 562
487 563
488 564
489 565
490 566
491 567
492 568
493 569
494 570
495 571
496 572
497 573
498 574
499 575
500 576
501 577
502 578
503 579
504 580
505 581
506 582
507 583
508 584
509 585
510 586
511 587
512 588
513 589
514 590
515 591
516 592
517 593
518 594
519 595
520 596
521 597
522 598
523 599
524 600
525 601
526 602
527 603
528 604

```

3 per visualizzare i calciatori con altezza compresa tra 1,80 e 1,90, 4 per visualizzare il calciatore con età minore, 5 per visualizzare elenco alfabetico dei calciatori, 6 per uscire <<endl;

```

190 cin>>scegli;
191 switch(scegli){
192     case 1:
193         ruoloGiocatore();
194         break;
195     case 2:
196         for(int i=0; i<n; i++){
197             if(altezza[i]>maxAltezza)
198                 maxAltezza=altezza[i];
199         }
200         cout<<"i calciatori con altezza maggiore sono: "<<endl;
201         for(int i=0; i<n; i++){
202             if(altezza[i]==maxAltezza)
203                 cout<<"nome: "<<nome[i]<<endl;
204         }
205         break;
206     case 3:
207         cout<<"i calciatori con altezza compresa tra 1,80 e 1,90
208 sono: "<<endl;
209         for(int i=0; i<n; i++){
210             if(altezza[i]>=180&&altezza[i]<=190)
211                 cout<<"nome: "<<nome[i]<<" cognome: "<<cognome[i]
212 <<" ruolo: "<<ruolo[i]<<endl;
213         }
214         break;
215     case 4:
216         minEta=anno[0];
217         for(int i=0; i<n; i++){
218             if(anno[i]<minEta)
219                 minEta=anno[i];
220         }
221         cout<<"i calciatori con eta minore sono: "<<endl;
222         for(int i=0; i<n; i++){
223             if(anno[i]==minEta)
224                 cout<<"nome: "<<nome[i]<<" cognome: "<<cognome[i]
225 <<" ruolo: "<<ruolo[i]<<endl;
226         }
227         break;
228     case 5:
229         for(int k=0; k<n-1; k++){
230             for(int i=k+1; i<n; i++){
231                 if(nome[k]>nome[i] || (nome[k]==nome[i] && cognome[k]
232 >cognome[i])){
233                     temp=nome[k];
234                     nome[k]=nome[i];
235                     nome[i]=temp;
236                     temp=cognome[k];
237                     cognome[k]=cognome[i];
238                     cognome[i]=temp;
239                 }
240             }
241         }
242         cout<<"elenco alfabetico dei calciatori:"<<endl;
243         for(int i=0; i<n; i++){
244             cout<<nome[i]<<" "<<cognome[i]<<endl;
245         }
246         break;
247     }
248 }while(scegli!=6);
249 return 0;
250 }

```