

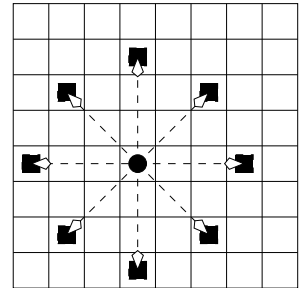
ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ (2006-07)

Άσκηση 4

Γράψτε ένα πρόγραμμα C (έστω ότι το πηγαίο αρχείο του ονομάζεται “squares.c”) το οποίο όταν καλείται σαν “squares n ” (με n κάποιο ακέραιο μεγαλύτερο του 4), να δημιουργεί και να εκτυπώνει στην έξοδο έναν πίνακα $n \times n$, στον οποίο να έχουν τοποθετηθεί όλοι οι ακέραιοι αριθμοί από το 1 έως το n^2 , σύμφωνα με τους εξής κανόνες:

1. Ο αριθμός 1 τοποθετείται στη θέση που βρίσκεται στην πρώτη γραμμή και στην πρώτη στήλη.

2. Για κάθε k από 1 έως $n^2 - 1$, οι αριθμοί k και $k + 1$ θα πρέπει να έχουν τοποθετηθεί στην ίδια γραμμή ή στην ίδια στήλη, έχοντας αφήσει δύο κενές θέσεις μεταξύ τους, ή στην ίδια διαγώνιο, έχοντας αφήσει μία κενή θέση μεταξύ τους. Για παράδειγμα, στο διπλανό σχήμα, αν ο αριθμός k έχει τοποθετηθεί στη θέση που βρίσκεται ο μαύρος κύκλος, οι δυνατές θέσεις για τον αριθμό $k + 1$ είναι αυτές με τα μαύρα τετράγωνα. Εννοείται, ότι δυνατές είναι μόνο οι θέσεις που βρίσκονται εντός του πλαισίου $n \times n$ που γεμίζουμε.



Σημειώνεται ότι το πρόβλημα δεν έχει λύση όταν $n \leq 4$ (εκτός από την περίπτωση της τετριμμένης λύσης για $n = 1$). Για κάθε $n > 4$ υπάρχουν πολλές λύσεις. Το πρόγραμμα που θα γράψετε αρκεί να βρίσκει μία απ' αυτές. Κάποιες πιθανές λύσεις του προβλήματος για $n = 5$ και $n = 6$ είναι οι εξής:

| | | | | |
|----|----|----|----|----|
| 1 | 19 | 16 | 4 | 20 |
| 14 | 24 | 8 | 11 | 23 |
| 17 | 5 | 21 | 18 | 6 |
| 2 | 10 | 15 | 3 | 9 |
| 13 | 25 | 7 | 12 | 22 |

| | | | | | |
|----|----|----|----|----|----|
| 1 | 28 | 7 | 2 | 25 | 8 |
| 13 | 18 | 34 | 10 | 15 | 35 |
| 32 | 3 | 24 | 29 | 6 | 23 |
| 20 | 27 | 14 | 19 | 26 | 9 |
| 12 | 17 | 33 | 11 | 16 | 36 |
| 31 | 4 | 21 | 30 | 5 | 22 |

Κάποια παραδείγματα εκτέλεσης του προγράμματος φαίνονται στη συνέχεια:

```
% ./squares 5
```

```
1 19 16 4 20
14 24 8 11 23
17 5 21 18 6
2 10 15 3 9
13 25 7 12 22
```

```
% ./squares 6
```

```
1 28 7 2 25 8
13 18 34 10 15 35
32 3 24 29 6 23
20 27 14 19 26 9
12 17 33 11 16 36
31 4 21 30 5 22
```

% ./squares 8

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 1 | 50 | 27 | 2 | 51 | 26 | 3 | 52 |
| 31 | 12 | 9 | 46 | 13 | 10 | 45 | 14 |
| 28 | 40 | 35 | 62 | 41 | 36 | 63 | 25 |
| 8 | 49 | 30 | 11 | 54 | 23 | 4 | 53 |
| 32 | 19 | 58 | 47 | 20 | 57 | 44 | 15 |
| 29 | 39 | 34 | 61 | 42 | 37 | 64 | 24 |
| 7 | 48 | 21 | 6 | 55 | 22 | 5 | 56 |
| 33 | 18 | 59 | 38 | 17 | 60 | 43 | 16 |

% ./squares 15

| | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 23 | 65 | 54 | 24 | 64 | 152 | 92 | 63 | 179 | 87 | 34 | 180 | 88 | 33 |
| 27 | 56 | 3 | 28 | 57 | 4 | 29 | 58 | 5 | 30 | 59 | 6 | 31 | 60 | 7 |
| 66 | 53 | 25 | 117 | 153 | 93 | 116 | 221 | 177 | 115 | 218 | 178 | 86 | 35 | 181 |
| 2 | 22 | 126 | 55 | 139 | 125 | 151 | 91 | 62 | 188 | 90 | 61 | 183 | 89 | 32 |
| 26 | 118 | 154 | 94 | 128 | 220 | 166 | 129 | 219 | 222 | 172 | 112 | 225 | 173 | 8 |
| 67 | 52 | 138 | 124 | 150 | 137 | 140 | 214 | 176 | 114 | 217 | 187 | 85 | 36 | 182 |
| 157 | 21 | 127 | 193 | 165 | 130 | 190 | 223 | 171 | 189 | 224 | 170 | 184 | 111 | 83 |
| 96 | 119 | 155 | 95 | 141 | 213 | 167 | 136 | 212 | 215 | 175 | 113 | 216 | 174 | 9 |
| 68 | 51 | 158 | 123 | 149 | 192 | 164 | 205 | 191 | 163 | 208 | 186 | 84 | 37 | 185 |
| 156 | 20 | 103 | 194 | 160 | 131 | 195 | 202 | 168 | 135 | 211 | 169 | 134 | 110 | 82 |
| 97 | 120 | 69 | 100 | 142 | 206 | 148 | 145 | 207 | 204 | 146 | 45 | 209 | 78 | 10 |
| 16 | 50 | 159 | 122 | 104 | 201 | 161 | 132 | 200 | 162 | 133 | 109 | 81 | 38 | 108 |
| 70 | 19 | 102 | 197 | 72 | 101 | 196 | 203 | 147 | 46 | 210 | 77 | 41 | 44 | 76 |
| 98 | 121 | 15 | 99 | 143 | 14 | 105 | 144 | 13 | 106 | 80 | 12 | 107 | 79 | 11 |
| 17 | 49 | 71 | 18 | 48 | 198 | 73 | 47 | 199 | 74 | 40 | 43 | 75 | 39 | 42 |

% ./squares 20

| | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 21 | 93 | 2 | 22 | 96 | 66 | 23 | 103 | 65 | 24 | 106 | 64 | 25 | 107 | 61 | 26 | 108 | 60 | 27 |
| 265 | 37 | 34 | 266 | 38 | 33 | 275 | 39 | 32 | 276 | 40 | 31 | 230 | 41 | 30 | 123 | 42 | 29 | 112 | 43 |
| 92 | 3 | 320 | 97 | 94 | 321 | 102 | 95 | 308 | 119 | 104 | 309 | 118 | 105 | 63 | 117 | 110 | 62 | 56 | 109 |
| 35 | 20 | 264 | 36 | 334 | 267 | 67 | 335 | 274 | 68 | 233 | 277 | 69 | 124 | 229 | 70 | 113 | 126 | 59 | 28 |
| 317 | 98 | 91 | 316 | 319 | 100 | 330 | 322 | 101 | 329 | 307 | 120 | 231 | 306 | 115 | 122 | 57 | 116 | 111 | 44 |
| 189 | 4 | 333 | 268 | 263 | 336 | 346 | 272 | 311 | 351 | 273 | 310 | 234 | 278 | 217 | 125 | 228 | 71 | 55 | 127 |
| 90 | 19 | 318 | 99 | 331 | 369 | 324 | 338 | 374 | 323 | 232 | 375 | 241 | 121 | 224 | 242 | 114 | 131 | 58 | 73 |
| 259 | 269 | 190 | 315 | 347 | 271 | 312 | 352 | 345 | 328 | 304 | 350 | 280 | 305 | 235 | 279 | 216 | 128 | 227 | 45 |
| 188 | 5 | 332 | 370 | 262 | 337 | 379 | 368 | 325 | 378 | 383 | 326 | 296 | 384 | 218 | 132 | 225 | 72 | 54 | 130 |
| 89 | 18 | 260 | 270 | 195 | 372 | 348 | 339 | 373 | 349 | 344 | 376 | 240 | 343 | 223 | 243 | 236 | 134 | 215 | 74 |
| 258 | 357 | 191 | 314 | 356 | 365 | 313 | 353 | 394 | 327 | 303 | 395 | 281 | 249 | 297 | 282 | 219 | 129 | 226 | 46 |
| 187 | 6 | 196 | 371 | 261 | 340 | 380 | 367 | 341 | 377 | 382 | 342 | 295 | 385 | 239 | 133 | 214 | 75 | 53 | 135 |
| 88 | 17 | 257 | 360 | 194 | 301 | 399 | 364 | 302 | 400 | 393 | 250 | 298 | 396 | 222 | 244 | 237 | 141 | 212 | 77 |
| 197 | 358 | 192 | 287 | 355 | 366 | 286 | 354 | 381 | 285 | 292 | 386 | 284 | 248 | 294 | 283 | 220 | 136 | 207 | 47 |
| 186 | 7 | 156 | 185 | 256 | 361 | 176 | 300 | 398 | 175 | 299 | 397 | 168 | 251 | 238 | 163 | 213 | 76 | 52 | 140 |
| 87 | 16 | 198 | 359 | 193 | 288 | 388 | 363 | 291 | 387 | 392 | 247 | 293 | 391 | 221 | 245 | 206 | 142 | 211 | 78 |
| 11 | 152 | 85 | 14 | 157 | 180 | 255 | 158 | 179 | 254 | 167 | 172 | 253 | 164 | 171 | 252 | 145 | 137 | 208 | 48 |
| 183 | 8 | 155 | 184 | 199 | 362 | 177 | 289 | 389 | 174 | 290 | 390 | 169 | 246 | 205 | 162 | 210 | 204 | 51 | 139 |
| 86 | 15 | 12 | 151 | 84 | 13 | 150 | 83 | 166 | 149 | 82 | 165 | 148 | 81 | 144 | 147 | 80 | 143 | 146 | 79 |
| 10 | 153 | 182 | 9 | 154 | 181 | 200 | 159 | 178 | 201 | 160 | 173 | 202 | 161 | 170 | 203 | 50 | 138 | 209 | 49 |

%

Προσθέστε στο πρόγραμμα που θα γράψετε και τη δυνατότητα όταν καλείται σαν “squares -check n”, να διαβάζει από την είσοδο έναν $n \times n$ πίνακα αριθμών και να ελέγχει αν είναι της μορφής που παρουσιάστηκε στο πρόβλημα αυτό. Έτσι, θα μπορείτε να ελέγχετε αν το προγράμμα σας γεννά σωστούς πίνακες. Παραδείγματα:¹

```
% ./squares -check 5
 1 19 16  4 20
14 24  8 11 23
17  5 21 18  6
 2 10 15  3  9
13 25  7 12 22
^D
Everything is OK
% ./squares -check 5
 1 18 16  4 20
14 24  8 11 23
17  5 21 19  6
 2 10 15  3  9
13 25  7 12 22
^D
Problem in placing 18
% ./squares 20 | ./squares -check 20
Everything is OK
% ./squares 100 | ./squares -check 100
Everything is OK
%
```

Με την επιλογή `-check`, το πρόγραμμά σας αρκεί να εκτυπώνει μόνο το ελάχιστο k για το οποίο υπάρχει λάθος στην τοποθέτησή του.

Η παράδοση της άσκησης αυτής συνίσταται στην υποβολή του πηγαίου αρχείου `squares.c` με διαδικασία που θα ανακοινωθεί σύντομα.

Υπόδειξη: Το πρόβλημα που παρουσιάζεται στην άσκηση αυτή είναι ένα πρόβλημα αναζήτησης. Βρισκόμαστε, δηλαδή, σε μία αρχική κατάσταση, για παράδειγμα αυτή που έχουμε τοποθετήσει μόνο τον αριθμό 1 στο πλαίσιο, στην επάνω αριστερά θέση. Στη συνέχεια, για την τοποθέτηση του 2 έχουμε κάποιες (συγκεκριμένα, τρεις) επιλογές. Διαλέγουμε κάποια απ’ αυτές και προχωρούμε στην τοποθέτηση του 3, επιλέγοντας, πάλι, κάποια από τις διαθέσιμες επιλογές. Συνεχίζοντας με αυτόν τον τρόπο, φιλοδοξούμε κάποια στιγμή να τοποθετήσουμε και τον τελευταίο αριθμό με “νόμιμο” τρόπο στο τελευταίο τετραγωνίδιο. Είναι πολύ πιθανό όμως κατά τη διαδικασία τοποθέτησης των αριθμών, κάποια στιγμή να βρεθούμε σε αδιέξοδο. Δηλαδή, να μην υπάρχει ελεύθερο “νόμιμο” τετραγωνίδιο για την τοποθέτηση του τρέχοντος αριθμού. Τότε, πρέπει να *οπισθοδρομήσουμε* στην πιο πρόσφατη επιλογή που είχαμε κάνει, να την αναιρέσουμε και να ακολουθήσουμε κάποια άλλη. Για την υλοποίηση αυτής της οπισθοδρόμησης χρειαζόμαστε μία στοιβιά, στην οποία να κρατάμε την αλληλουχία των βημάτων που έχουμε κάνει, ή κάτι ισοδύναμο. Ας σημειωθεί, ότι για να μπορεί το πρόγραμμά σας να δουλεύει και για μεγάλα n , θα πρέπει όταν είναι να πάρει μία απόφαση, για το πού να τοποθετήσει τον επόμενο αριθμό, να το κάνει αυτό με κάποιο “έξυπνο” τρόπο. Απλώς, εσείς σκεφτείτε με τι είδους εξυπνάδα θα πρέπει να το εφοδιάσετε. Περισσότερη συζήτηση επί του θέματος θα γίνει σίγουρα στο φόρουμ.

¹ Στα δύο πρώτα παραδείγματα, οι γραμμές με τους αριθμούς που είναι τα στοιχεία του πίνακα, μέχρι και το Control-D (^D), πληκτρολογήθηκαν από τον χρήστη.