

Opgave 1

Kun je een 8 bij 8 schaakbord, waarbij een willekeurig wit veld veld en een willekeurig willekeurig zwart veld zijn weggelaten, volleggen met dominostenen? Zo ja, geef aan hoe. Zo nee, toon dit aan.

Opgave 1

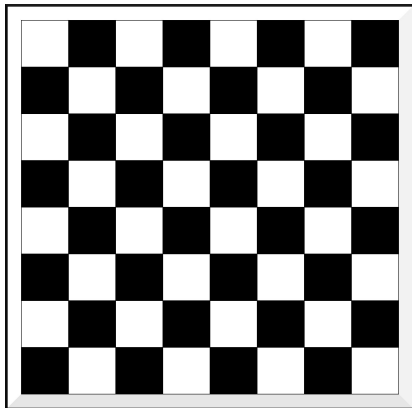


Figure: Schaakbord

Opgave 2 (Levitin 1.1.12)

Locker deuren. Er zijn n lockers in een gang, op volgorde genummerd van 1 t/m n . In het begin zijn ze allemaal dicht. Je loopt er n langs, iedere keer beginnend bij deur 1. Op de i^{de} keer dat je voorbij loopt, *switch* je de deur van iedere i^{de} locker. (Switchen wil zeggen: openen wanneer hij dicht zit, dicht doen wanneer hij open is.) Bijvoorbeeld, de tweede keer dat je voorbij loopt switch je deuren 2, 4, 6, 8 etc. De derde keer switch je deuren 3, 6, 9 etc Nadat je voor het laatst bent langs gelopen, welke deuren zijn er nu open?

Opgave 3 (Levitin 2.1.4a)

Handschoenen in droger. Er zitten 22 handschoenen in een droger: 5 paar roden, 4 paar gelen en 2 paar groenen. Het is donker, en je moet er een paar handschoenen uit selecteren, dat wil zeggen een linker en een recht van dezelfde kleur. Je pakt er een tegelijk. Hoeveel keer zul je in het beste geval een handschoen uit de droger moeten pakken? En in het slechtste geval?

Opgave 3 (Levitin 2.1.4b)

Verdwenen sokken. Na het wassen van 5 verschillende paren sokken, mis je twee sokken. In het beste geval heb je nu 4 complete paren sokken, in het slechtste geval 3. Neem aan dat de kans voor iedere sok om te verdwijnen even groot is, wat is de kans dat het best-case scenario van toepassing is? En wat is de kans dat het worst-case scenario van toepassing is? En the average case?

Antwoord opgave 7

```
1  buur* inhoud[n];
2  totaal = 0;
3  for( int knoop = 0; knoop < n; knoop++ ) {
4      teller = 0;
5      hulp = inhoud[knoop];
6      while( hulp != NULL ) {
7          teller ++;
8          hulp = hulp->volgende;
9      }
10     if( teller \% 2 == 1 ) {
11         totaal ++;
12     }
13 } // for
14 if( totaal <= 2 ) {
15     cout << "hooguit twee";
16 } else {
17     cout << "meer dan twee";
18 }
```

Antwoord opgave 8A

```
1  int takken( buur* G[n] ) {
2      int count = 0;
3      for( int i = 0; i < n; i++ ) {
4          buur* hulp = G[i];
5          while( hulp->volgende != NULL ) {
6              count++;
7              hulp = hulp->volgende;
8          }
9      }
10     // alles "per ongeluk" dubbel geteld -> corrigeren.
11     return count / 2;
12 }
```

Antwoord opgave 8B

```
1  int takken( int G[n][n] ) {
2      int count = 0;
3      for ( int i = 0; i < n; i++ ) {
4          for ( int j = 0; j < n; j++ ) {
5              if( G[i][j] > 0 ) {
6                  count++;
7              }
8          }
9      }
10     // alles "per ongeluk" dubbel geteld -> corrigeren.
11     return count / 2;
12 }
```


Antwoord opgave 9A

```
1 // er staat een tak van i naar j, dus i G[i][j] = 1 en
   G[j][i] = 0; Moet worden omgeraaid.
2 void draaiom( int G[n][n], int i, int j ) {
3     G[i][j] = 0;
4     G[j][i] = 1;
5 }
```

Antwoord opgave 9B

```
1 void draaiom2( buur* G[n], int i, int j ) {
2     // j uit buurlijst van i halen, i in buurlijst van j
      zetten; hoeft niet gesorteerd.
3     buur* hulp = G[i];
4     vorige = null;
5     while( hulp->knoopnummer != i ) {
6         vorige = hulp;
7         hulp = hulp->volgende;
8     }
9     if( vorige == null ) {
10        G[i] = hulp->volgende;
11        delete hulp;
12    } else {
13        vorige->volgende = hulp->volgende;
14        delete hulp;
15    }
16
17    hulp = new buur;
18    hulp->knoopnummer = i;
19    hulp->volgende = G[j];
20    G[j] = hulp;
21 }
```