

## Aufgabenblatt 4, Abgabetermin 8.6.2020

Lösen Sie Aufgaben im Umfang von 15 Punkten.

### Aufgabe 21: Quadratische Arrays durchlaufen

**5 Punkte**

Schreiben Sie ein Programm, das quadratische Arrays ausgibt, deren Felder unterschiedlich durchnummeriert sind. Das Programm liest `int`-Werte  $n$  und  $v$  ein.  $n$  ist die Seitenlänge des quadratischen Arrays, und  $v$  ist die Variante der Durchnummerierung.

$v = 1$ : die Einträge sind von links nach rechts und von oben nach unten durchnummeriert

$v = 2$ : die Einträge sind von oben nach unten und von links nach rechts durchnummeriert

$v = 3$ : die Einträge sind von links nach rechts und von unten nach oben durchnummeriert

$v = 4$ : die Einträge sind von links nach rechts in absteigenden Diagonalen durchnummeriert  
 Unten sehen Sie ein paar Beispiele (die Ausgabe Ihres Programms muss nicht so „schön“ sein).

```
[ [ 0, 1, 2, 3],
  [ 4, 5, 6, 7],
  [ 8, 9, 10, 11],
  [12, 13, 14, 15] ]
```

$n = 4, v = 1$

```
[ [0, 5, 10, 15, 20],
  [1, 6, 11, 16, 21],
  [2, 7, 12, 17, 22],
  [3, 8, 13, 18, 23],
  [4, 9, 14, 19, 24] ]
```

$n = 5, v = 2$

```
[ [6, 7, 8],
  [3, 4, 5],
  [0, 1, 2] ]
```

$n = 3, v = 3$

```
[ [ 0, 4, 8, 12],
  [13, 1, 5, 9],
  [10, 14, 2, 6],
  [ 7, 11, 15, 3] ]
```

$n = 4, v = 4$

### Aufgabe 22: Maulwürfe

**5 Punkte**

Maulwürfe untergraben eine Wiese. Stellen Sie dar, wo die Maulwürfe sind und wie gefährdet die Stellen sind, unter denen sie noch nicht sind.

Schreiben Sie dazu ein Programm, das zwei positive ganze Zahlen  $z$  (*Zeilen*) und  $s$  (*Spalten*) sowie eine `float`-Zahl  $p$  (*Probability*) aus dem Intervall  $[0, 1]$  einliest. Ihr Programm soll dann ein Array aus  $z$  Zeilen und  $s$  Spalten mit Wahrheitswerten füllen. Die Wahrscheinlichkeit für `True` soll  $p$  sein. An den Stellen mit `True` ist ein Maulwurfshügel, und an den Stellen mit `False` nicht. Geben Sie zuerst das Array aus, indem Sie `*` für `True` und `.` für `False` schreiben. Geben Sie danach das Array erneut aus, indem Sie diesmal die Anzahl der benachbarten Maulwurfshügel bei jedem Feld ohne Maulwurfshügel schreiben. Zum Beispiel:

```
* * . * . . . *
* . * . * . * .
* * . . . * . .
* . . . . * . .
* . . . . * . .
```

```
* * 3 * 2 2 2 *
* 6 * 3 * 3 * 2
* * 2 2 3 * 3 1
* 4 1 0 3 * 3 0
* 2 0 0 2 * 2 0
```

Tipp: Sie können in Ihrem Programm auch ein  $(z + 2)$  mal  $(s + 2)$  Array verwenden, um sich einige Spezialfälle zu sparen.

### Aufgabe 23: Magische Quadrate erkennen

**5 Punkte**

Ein *magisches Quadrat* ist eine quadratische Anordnung der  $n \cdot n$  Zahlen  $1, 2, \dots, n^2$ , so dass alle Zeilen, alle Spalten und die beiden Diagonalen die gleichen Summen haben. Unten stehen

magische Quadrate mit den Seitenlängen 3, 4 und 5. Im ersten Quadrat haben alle Zeilen, Spalten und Diagonalen die Summe 15, im zweiten ist die Summe 34, und im dritten ist sie 55.

8	1	6
3	5	7
4	9	2

16	6	9	3
1	11	8	14
7	13	2	12
10	4	15	5

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

Schreiben Sie ein Programm, das von einem gegebenen quadratischen 2d-Array überprüft, ob es ein magisches Quadrat ist.

**Aufgabe 24: Magische Quadrate erzeugen**

**5 Punkte**

Bei Wikipedia ist beschrieben, wie man magische Quadrate mit ungerader Seitenlänge  $n$  konstruieren kann.

Man fängt oben in der Mitte mit 1 an und füllt dann die anderen Zahlen der Reihe nach gemäß der folgenden Regel in die anderen Felder ein:

Wenn die zuletzt geschriebene Zahl kein Vielfaches von  $n$  ist, dann trage die nächste Zahl in das Feld oben rechts vom zuletzt ausgefüllten Feld. Ist die zuletzt geschriebene Zahl ein Vielfaches von  $n$ , dann trage die nächste Zahl in das Feld unter der zuletzt geschriebenen Zahl. (...)

Hierbei wird das magische Quadrat als periodisch wiederholt angesehen, d. h., wenn man über den oberen Rand hinausgeht (das passiert schon beim ersten Schritt), kommt man von unten wieder hinein, und wenn man rechts hinausgeht, dann kommt man von links wieder hinein.

Die oben abgebildeten magischen Quadrate mit den Seitenlängen 3 und 5 sind nach diesem Vorgehen entstanden. Schreiben Sie ein Programm, das magische Quadrate mit ungeraden Seitenlängen nach diesem Vorgehen konstruiert.

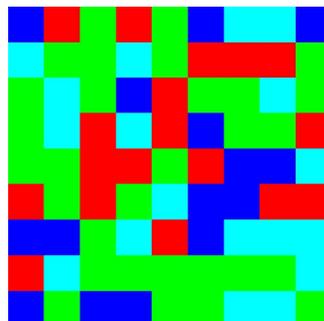
**Aufgabe 25: Häuser anmalen**

**5 Punkte**

In Quadratia streicht jeder Bürger an Pfingsten sein Haus neu an. Eine Woche zuvor wählt er zufällig aus seiner Nachbarschaft – das sind die 8 Nachbarhäuser und sein eigenes Haus – ein Haus aus und nimmt die Farbe dieses Hauses als Farbe, in der er Pfingsten sein Haus streichen wird. (Er kann sein Haus also auch in der alten Farbe neu anstreichen.)

Es gibt die Vermutung, dass nach gewisser Zeit alle Häuser des Landes die gleiche Farbe haben. Stimmt das?

Quadratia ist ein quadratisches Land aus lauter quadratischen Grundstücken. Die Abbildung



zeigt die aktuellen Farben der Häuser. Jeder Bürger sieht die Farben der 8 Nachbarhäuser und sein eigenes Haus. Die Bürger, die an der oberen Kante wohnen, haben die entsprechenden Nachbarn an der unteren Kante usw..

Schreiben Sie ein Programm, das  $n$  und  $c$  einliest. Es soll die  $n$ -mal- $n$  kleinen Quadrate in einem großen Quadrat mit Seitenlänge  $n$  zufällig mit  $c$  Farben anmalen und dann den jährlichen Neuanstrich simulieren, bis – vielleicht – alle Häuser die gleiche Farbe haben.

### Aufgabe 26: Zufallsgesteuerte Rasenmäher

5 Punkte

In Aufgabe 10 hatten wir einen Sucher durch ein Punktegitter gehen lassen. Ostern ist vorbei – jetzt wird wieder der Rasen gemäht. Nun sollen  $\ell$  Rasenmäher einen Rasen mähen, den wir uns als Punktegitter der Größe  $n$  mal  $n$  vorstellen. Ihre Schritte gehen – wie beim Sucher in Aufgabe 12 – zufällig in Richtung Norden, Osten, Süden oder Westen. Alle Rasenmäher starten im Punkt unten links – es können also auch viele Rasenmäher am selben Punkt arbeiten. Sie laufen solange, bis der ganze Rasen gemäht ist – d.h. bis alle Gitterpunkte mindestens einmal von irgendeinem Rasenmäher durchfahren wurden.

Schreiben Sie ein Programm, das  $\ell$  und  $n$  einliest und mittels Simulation die mittlere Anzahl der Schritte berechnet, die die Rasenmäher machen müssen, bis sie ihr gemeinsames Ziel erreicht haben.

2 Zusatzpunkte: stellen Sie graphisch auf der Konsole dar, wie die Rasenmäher den Rasen mähen.

### Aufgabe 27: Rechenmauern

5 Punkte

Eine Rechenmauer ist eine Anordnung von Zahlen, die man sich aufgetürmt wie Ziegelsteine vorstellen kann. Von Reihe zu Reihe liegt ein Stein weniger. Die Zahlen, die nicht in der untersten Reihe liegen, sind jeweils die Summe der beiden direkt unter ihnen liegenden „Steine“.

Eine beliebte Aufgabe in der Grundschule zum Üben von Addition und Subtraktion ist es, solche lückenhaften Rechenmauern ausfüllen zu lassen. Schreiben Sie ein Programm, das eine Zahl  $n$  einliest und eine Rechenmauer aus  $n$  Reihen (mit  $n$  Zahlen in der untersten Reihe) erzeugt. (Diese Mauer kann man dadurch erzeugen, dass man die Zahlen in der untersten Reihe zufällig wählt und die übrigen Zahlen ausrechnet.) Erzeugen Sie anschließend daraus eine lückenhafte Rechenmauer, indem Ihr Programm zufällig die Hälfte (oder auch etwas mehr) der Zahlen in der Mauer ersetzt.

Wenn Sie noch eine schöne Ausgabe (mindestens so schön wie unten) erzeugen, können Sie 2 Zusatzpunkte erreichen.

105	*	45	*
50 55	* 55	26 19	* *
23 27 28	23 27 *	14 12 7	14 * 7
11 12 15 13	* 12 15 13	5 9 3 4	5 9 * *
4 7 5 10 3	* * * * 3		

Eine vollständige Rechenmauer ( $n = 5$ )...

...und die daraus entstandene Rechenaufgabe.

Beim Durchlöchern dieser Rechenmauer ... ( $n = 4$ )

... ist eine Rechenaufgabe ohne eindeutige Lösung entstanden.