



Übungsblatt 13

Anmerkung: Alle Punkte dieses Blatts werden als **Bonuspunkte** gewertet.

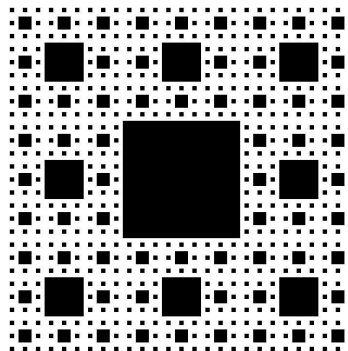
Zusatzaufgabe V Sierpinski-Teppich

[☒ 5 Punkte]

Sie sehen auf der Abbildung einen sogenannten Sierpinski-Teppich vierter Stufe. Ein Sierpinski-Teppich n -ter Stufe wird folgendermaßen erzeugt:

- Starte mit einem weißen Quadrat.
- Unterteile das Quadrat in 9 gleichgroße Unterquadrate.
- Färbe das mittlere Quadrat schwarz.
- Fülle die anderen 8 Unterquadrate mit Sierpinski-Teppichen $(n - 1)$ -ter Stufe.

Dabei ist ein Sierpinski-Teppich 0-ter Stufe ein weißes Quadrat.



Schreiben Sie ein Programm, welches einen Sierpinski-Teppich n -ter Stufe erzeugt. Ihr Programm soll dabei die Größe des Ausgangsquadrats und n als Kommandozeilenparameter akzeptieren.

Um Ihnen die Visualisierung zu erleichtern, steht auf der Vorlesungshomepage das Programm `sierpinski-graphical.cc` zur Verfügung. Dieses liest die Datei `sierpinski.dat` ein und erstellt ein Portable Bitmap-Bild, das sie mit Bildbetrachtern wie IrfanView anschauen können.

Dabei wird erwartet, dass die Ausgabe folgende Struktur hat:

- in der ersten Zeile steht die Größe des Quadrats/der Matrix
- in jeder weiteren Zeile steht genau eine Koordinate eines schwarz zu färbenden Eintrags, wobei x - und y -Koordinate durch ein Leerzeichen getrennt werden.

Sie können diese Ausgabe auf der Konsole ausgeben und mittels `>` in eine Datei umleiten, oder aber sich die Funktionsweise von `std::fstream` aneignen, um `sierpinski.dat` zu erzeugen.

Nicht bei jeder Wahl von Größe und Stufe entsteht eine komplett regelmäßige Matrix (z. B. nicht bei 500 und 6). Überlegen Sie sich warum dies so ist und bei welcher Wahl der Größe dies trotzdem immer gewährleistet ist.

Zusatzaufgabe VI Piping

[☒ 10 Punkte]

Bei der Verarbeitung von Daten in mehreren Schritten ist es häufig praktisch, die einzelnen Teile in Objekte zu packen, die jeweils von einem Vorgänger Daten empfangen und die bearbeiteten Daten an ein weiteres Objekt weiterzuschicken. Diesen Prozess bezeichnet man als *piping*¹. Verwendet wird diese Technik z. B. bei der Verkettung verschiedener XML-Transformationen.

Wir wollen ein Piping über ein gemeinsames Interface vieler Filterklassen erreichen, die jeweils einzelne Zeilen an das nächste Objekt weiterreichen. Ein Hauptprogramm, das eine Datei mit dem Namen `testdatei` einliest, alle Zeichen in Kleinbuchstaben verwandelt, mit `rot13` verschlüsselt², alle Zeilen umdreht und schliesslich auf dem Bildschirm ausgibt, könnte wie folgt aussehen:

```
int main()
{
    ConsoleWriter writer;
    ReverseFilter revert( writer );
    Rot13Filter rot( revert );
    LowercaseFilter lower( rot );
    FileReader reader( "testdatei", lower );

    // Der Erste in der Pipe fängt an, reicht die Daten an
    // den jeweils nächsten weiter
    reader.go();
};
```

Sie können aber auch ein anders geartetes Interface implementieren, die Hauptsache ist, dass zwischen Eingabe- und Ausgabeobjekt die Filter in beliebiger Reihenfolge eingefügt werden können.

Implementieren Sie bitte: eine Einleseklasse (Datei oder Bildschirm), eine Ausgabeklasse (Datei oder Bildschirm) sowie mindestens drei der folgenden Filter:

¹In Unix und den daraus entstandenen Systemen kann man mit dem `|`-Zeichen die Ausgabe eines Programms an das nächste weiterschicken um diesen Effekt zu erreichen.

Siehe https://de.wikipedia.org/wiki/Pipe_%28Informatik%29#Unix

²<https://de.wikipedia.org/wiki/ROT13>

- alle Zeichen der Eingabe in Kleinbuchstaben verwandeln
- die Eingabe mit rot13 verschlüsseln bzw. entschlüsseln
- jede Zeile umdrehen (also das letzte Zeichen zuerst ausgeben)
- mehrfach vorkommende Zeilen nur einmal ausgeben
- Kompression und Dekompression einer zeilenweisen Wortliste nach dem folgenden Schema: überprüfe, wie viele Buchstaben am Wortanfang mit dem Anfang des letzten Worts übereinstimmen. Wenn diese Zahl n grösser 9 ist, setze sie auf 9. Gebe diese Zahl als Ziffer aus, danach den Rest des aktuellen Worts. Als Beispiel: die Wortliste

Aachen		0Aachen
Aachens		6s
Aal		2l
Aale		3e
Aalen		4n
Aales	wird zu	4s
abänderlich		0abänderlich
abänderliche		9che
abänderlichem		9chem
abänderlichen		9chen
abänderlicher		9cher

Diese Art der Kompression stammt aus dem Passwortcracker Crack und dient dazu, große Wörterbücher sehr platzsparend abzuspeichern.

- Für die ganz mutigen: ein Filter, der zeilenweise sortiert. Dieser macht natürlich vor der obigen Kompression viel Sinn.

Zusatzaufgabe VII Näherungswert für Pi

[☒ 5 Punkte]

Es sei

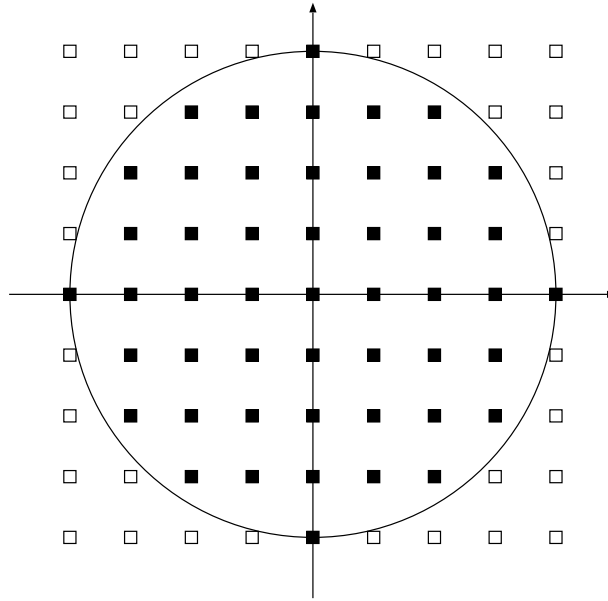
$$A_n = \{(x, y) \in \mathbb{Z} \times \mathbb{Z} \mid x^2 + y^2 \leq n^2\}$$

Schreiben Sie eine Funktion

```
int imkreis( int n )
```

die die Anzahl der Elemente der Menge A_n berechnet, also wieviele Punkte eines $(2n+1) \times (2n+1)$ -Gitters innerhalb oder auf der Kreislinie eines Kreises mit Radius n liegen.

Wie kann man über diese Anzahl eine Näherung von π berechnen?



**Abgabe: 28. Januar 2016, 14:15 Uhr,
in die Zettelkästen in der Ecke zwischen HS 2 und Seifertraum (INF 288)
sowie per E-Mail (✉) an Ihren Tutor**