



Übung 3

Aufgabe 1: Primfaktoren

Diese Aufgabe dient dem nochmaligen Üben von Kontrollstrukturen (Schleifen, if). Es soll ein Programm geschrieben werden, das eine Zahl einliest und die Primfaktorenzerlegung dieser Zahl berechnet. Dazu bedienen wir uns folgender Vorgehensweise:

1. Wir beginnen mit dem Faktorkandidaten 2.
2. Solange die eingegebene Zahl grösser als 1 ist:
3. Falls der Faktorkandidat ein tatsächlicher Faktor der eingegebenen Zahl ist, geben wir ihn aus und dividieren die eingegebene Zahl durch diesen Faktor. Das Ergebnis der Division betrachten wir als die eingegebene Zahl und fahren mit Schritt 2 fort.
4. Falls nicht, nehmen wir den nächst höheren Faktorkandidaten und fahren ebenfalls mit Schritt 2 fort.

Beispiel: Mit der eingegebenen Zahl 60 funktioniert das dann wie folgt:

1. 2 teilt 60 \Rightarrow Schritt 3: Die eingegebene Zahl wird $60/2 = 30$.
2. 2 teilt 30 \Rightarrow Schritt 3: Die eingegebene Zahl wird $30/2 = 15$.
3. 2 teilt nicht 15 \Rightarrow Schritt 4: Der nächste Faktorkandidat ist 3.
4. 3 teilt 15 \Rightarrow Schritt 3: Die eingegebene Zahl wird $15/3 = 5$.
5. 3 teilt nicht 5 \Rightarrow Schritt 4: Der nächste Faktorkandidat ist 4.
6. 4 teilt nicht 5 \Rightarrow Schritt 4: Der nächste Faktorkandidat ist 5.
7. 5 teilt 5 \Rightarrow Schritt 3: Die eingegebene Zahl wird $5/5 = 1$.

Aufgabe:

- a) Schreiben Sie ein C++-Programm, das den vorgegebenen Algorithmus implementiert. (Beispielausgabe $60 = 2 * 2 * 3 * 5$)

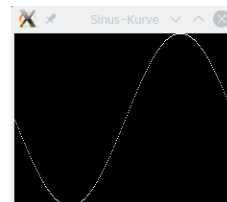
Beim vorgegebenen Algorithmus wird jeder Primteiler so oft ausgegeben, wie er die Zahl teilt. Für 10368 wird siebenmal die Zahl 2 ausgegeben und viermal die Zahl 3. Lieber hätte man eine Ausgabe der Form 2^7 und 3^4 .

- b) Erweitern Sie das Programm aus Unteraufgabe a) so, dass es jeden Primfaktor nur einmal mit der dazugehörigen Vielfachheit ausgibt. (Beispielausgabe $10368 = 2^7 * 3^4$)

Aufgabe 2: Graphik - Sinuskurve

Unten finden Sie die Beschreibung einer einfachen Graphikbibliothek.

1. Lesen Sie die Beschreibung durch.
2. Kopieren Sie die 3 Dateien `gfx.hh`, `gfx.cc` und `gfx_example.cc` von der Web Seite des Informatik Kurses in Ihr Directory.
3. Das Programm `gfx_simple.cc` kann dann mit der folgenden Anweisung kompiliert werden:
`g++ -lX11 -o gfx_example gfx.cc gfx_example.cc`



So sollte das Ergebnis der Aufgabe aussehen.

Aufgabe: Schreiben Sie ein Programm, das einen Graphen der Funktion $x \mapsto \sin(x)$ für den Bereich $x \in (0, 2\pi)$ auf den Bildschirm malt. Verwenden Sie dazu die Funktion `double sin(double)`; aus der Header-Datei `math.h`.

Beschreibung der Graphikbibliothek

Die Graphikbibliothek `gfx.hh` stellt elementare Graphikfunktionen zur Verfügung. Alle dabei auftretenden Koordinaten sind vom Typ `int` und werden 1:1 in Bildschirmpunkte übersetzt. Der Koordinatenursprung $(0,0)$ befindet sich in der linken oberen Ecke des Graphikfensters. Die x -Achse wächst von links nach rechts, die y -Achse von oben nach unten.

<code>void gfx_point(int x, int y)</code>	Punkt an Stelle (x, y) .
<code>void gfx_line(int x1, int y1, int x2, int y2);</code>	Linie von $(x1, y1)$ nach $(x2, y2)$.
<code>void gfx_rectangle(int x, int y, int w, int h);</code>	Rechteck mit Breite w , Höhe h und linker oberer Ecke (x, y) .
<code>void gfx_circle(int x, int y, int r);</code>	Kreis mit Zentrum (x, y) und Radius r .
<code>void gfx_ellipse(int x, int y, int w, int h);</code>	Ellipse mit Zentrum (x, y) und Achsen w und h .
<code>void gfx_open(int w, int h, const char *t);</code>	Öffnet Graphikfenster mit Höhe h , Breite w und Titel t .
<code>char gfx_wait();</code>	Wartet auf eine Zeicheneingabe.

Tabelle 1: Zeichenfunktionen der Graphikbibliothek `gfx.h`.

Bevor man zeichnen kann, muss das Graphikfenster auf dem Bildschirm erzeugt werden. Dies wird mit folgendem Aufruf gemacht: `gfx_open(w, h, name)`; Dabei spezifizieren `int w`; die Breite, `int h`; die Höhe und `char name[]`; die Überschrift des Fensters.