

Programmieren II für Studierende der Mathematik

Blatt 4

Aufgabe 4 Sei $N \in \mathbb{N}$, $N \geq 2$. Vereinbaren Sie $N \leq 2^{32}$ als Konstante für das gesamte Programm am besten durch Zuweisung eines Literals in hexadezimaler Schreibweise. Kennt man die Primzahlen kleiner als N , so lässt sich die Primfaktorzerlegung einer Zahl $n \in \mathbb{N}$ mit $n < N^2$ durch fortgesetztes Abdividieren der Primteiler bestimmen.

Erstellen Sie zunächst eine Klasse `Primzahlen` mit einer internen statischen Datenkomponente vom Typ `bitset<N>`, die für jede Zahl $k \in \mathbb{N}$ mit $k < N$ speichert ob sie eine Primzahl ist, oder nicht. Implementieren Sie eine statische Komponentenfunktion die das Sieb des Eratosthenes verwendet um die statische Datenkomponente korrekt zu befüllen. Rufen Sie diese statische Komponentenfunktion zu Beginn des Hauptprogramms einmal auf.

Hinweis (Sieb des Eratosthenes). Um alle Primzahlen kleiner als N zu bestimmen können wir ausgehend von $k = 2$ alle Vielfachen von Primzahlen $i \cdot k$ streichen mit $i \geq k$ und $i \cdot k < N$. Die erste noch nicht gestrichene Zahl ist dann die nächste Primzahl k . Iterieren Sie dieses Verfahren solange $k^2 < N$.

Fügen Sie zur Klasse `Primzahlen` eine statische Komponentenfunktion `ist_primzahl` hinzu die für ein als Parameter gegebenes n mit $0 \leq n < N$ vom Typ `unsigned long` als `bool` zurück gibt ob n eine Primzahl ist.

Erstellen Sie eine Klasse `Primfaktor`. Objekte der Klasse `Primfaktor` sollen zur Modellierung einer Primzahlpotenz der Form p^k für p eine Primzahl und sowohl p wie auch k vom Typ `int` dienen.

Erstellen Sie eine Klasse `Primfaktorzerlegung`. Objekte der Klasse `Primfaktorzerlegung` modellieren eine natürliche Zahl indem intern eine Liste von Primfaktoren vom Datentyp `list<Primfaktor>` gespeichert wird. Als Invariante der Klasse soll sichergestellt werden, dass die Liste von Primfaktoren stets aufsteigend nach Primzahl sortiert vorliegt und für alle gespeicherten Primfaktoren p^k gilt, dass $k \geq 1$.

Erstellen Sie einen Konstruktor für die Klasse `Primfaktorzerlegung` mit einem Argument $0 < n < N^2$ vom Typ `unsigned long`. Für $n = 1$ soll die interne Liste leer angelegt werden. Verwenden Sie zur Berechnung der Primfaktorzerlegung die Funktion `ist_primzahl` aus der Klasse `Primzahlen`.

Überladen Sie den Ausgabeoperator für die Klasse `Primfaktorzerlegung` sodass die Ausgabe in der Form $p_1^{k_1} \cdot \dots \cdot p_n^{k_n}$ erfolgt. Für die leere Liste soll 1 ausgegeben werden.

Fügen Sie einen Typumwandlungsoperator hinzu um Objekte der Klasse `Primfaktorzerlegung` zu Werten vom Typ `unsigned long` implizit konvertieren zu können.

Erstellen Sie eine mit der Klasse `Primfaktorzerlegung` befreundete Funktion `ist_primzahl` die für ein Argument vom Typ `Primfaktorzerlegung` unter direkter Verwendung der internen Darstellung als Liste von Primfaktoren bestimmt und als `bool` zurückgibt ob das Argument eine Primzahl ist.

Erstellen Sie ein Hauptprogramm, das für Zahlen der Form $2^j - 1$ mit $j = 1, \dots, 64$ und $2^j - 1 \leq N^2 - 1$ jeweils die Primfaktorzerlegung ausgibt. Sofern es sich um eine Primzahl handelt soll jeweils zusätzlich der Text `Mersenne-Primzahl` ausgegeben werden. Überprüfen Sie zudem ob das Ergebnis der Typumwandlung von der Primfaktorzerlegung zurück zu `unsigned int` identisch ist mit der zu faktorisierenden Zahl und geben Sie eine geeignete Warnung aus, sollte dem nicht so sein.