Prof. Franz Schweiggert
Christian Ehrhardt
Abteilung Angewandte Informationsverarbeitung
7. November 2002
Blatt 4

Allgemeine Informatik 3 (WS 2002/2003)

Abgabetermin 18.11.2002

Primzahlsieb (Sieb des Erathosthenes)

Aufgabe dieses Blattes ist es, zu ermitteln wieviele Primzahlen es gibt, die kleiner als ein bestimmtes Limit sind. Dazu müssen alle Primzahlen bis zu diesem Limit ermittelt werden. Ein Verfahren dazu ist ein sogenanntes Primzahlensieb, das in diesem Blatt implementiert werden soll.
Dieses Verfahren arbeitet auf einem Array, das genau einen Eintrag für jede Zahl zwischen 0 und dem Limit hat. Der Eintrag einer Zahl in diesem Array kann drei verschiedene Werte haben:

no Sicher keine Primzahl

yes Sicher eine Primzahl

maybe Noch unklar

Zu Beginn sind alle Zahlen mit Ausnahme von 0 und 1 als **maybe** markiert, 0 und 1 sind als **no** markiert. Anschließend werden die folgenden drei Schritte wiederholt durchgeführt:

- suche die kleinste Zahl, die als **maybe** markiert ist
- ändere ihre Markierung auf **yes**
- markiere alle Vielfachen dieser Zahl als **no**

Diese drei Schritte werden solange wiederholt, bis das Quadrat der Zahl, die im ersten Schritt gefunden wird, größer oder gleich dem Limit ist. Zum Schluß werden alle noch als **maybe** markierten Zahlen (die kleiner als das Limit sind) als **yes** markiert. Jetzt sind alle Primzahlen in dem Array als **yes** und alle Nichtprimzahlen als **no** markiert.

Beispiel: Limit = 10

Zu Beginn sieht das Array so aus:

Index	0	1	2	3	4	5	6	7	8	9
Wert	**no**	**no**	**maybe**	**maybe**	**maybe**	**maybe**	**maybe**	**maybe**	**maybe**	**maybe**

Die kleinste Zahl, die noch als **maybe** markiert ist, ist die 2. Diese wird also als **yes** markiert, ihre Vielfachen, also die Zahlen 4, 6 und 8, werden als **no** markiert:

Index	0	1	2	3	4	5	6	7	8	9
Wert	**no**	**no**	**yes**	**maybe**	**no**	**maybe**	**no**	**maybe**	**no**	**maybe**

Jetzt ist die kleinste Zahl, die noch als **maybe** markiert ist die 3. Also wird die 3 als **yes** und Ihre Vielfachen, also die 6 und die 9, als **no** markiert:

Index	0	1	2	3	4	5	6	7	8	9
Wert	**no**	**no**	**yes**	**yes**	**no**	**maybe**	**no**	**maybe**	**no**	**no**

Jetzt gibt es keine Zahl mehr, die mit **maybe** markiert ist und deren Quadrat kleiner als das Limit 10 ist. Also werden zum Schluß noch die restlichen mit **maybe** markierten Zahlen 5 und 7 als **yes** markiert. Es stellt sich also heraus, daß es 4 Primzahlen gibt, die kleiner als 10 sind, nämlich 2, 3, 5 und 7.

Hinweise und Ergänzungen

- Das Limit darf als Konstante im Programmtext auftauchen, es muß aber leicht (Änderung nur an einer Stelle im Programm) möglich sein, das Limit zu ändern.
- Sollte Bedarf bestehen, die Wurzel aus einer Zahl zu berechnen, dann kann dazu die Funktion `sqrt` aus `math.h` verwendet werden. **Beim Übersetzen muß dann dem Compiler zusätzlich die Option -lm übergeben werden.**
- Testet Euer Programm mit verschiedenen Werten für das Limit und versucht herauszufinden, wie hoch das Limit sein darf, damit die Laufzeit 20s nicht übersteigt. Die Laufzeit eines Programms kann einfach gemessen werden, indem man das Kommando time verwendet (time nice a.out, anschließend die CPU-Zeit in der Zeile User ablesen). Auf einer der neueren Suns in O27/211 sollte ein Limit von 20 Millionen in 20s ein realistisches Ziel sein.