

Programmieren 1 Übung: Arrays: Sieb des Eratosthenes (Liste aller Primzahlen)

Klaus Kusche

Schon die alten Griechen kannten folgende Methode, um ohne Multiplizieren oder Dividieren zu einer Liste aller Primzahlen bis zu einer gewissen Grenze **n** zu gelangen:

- Du brauchst **n** Markierungen (ein Array): Die **i**-te Markierung gehört zur Zahl **i**. Am Ende des Programms sollen genau jene Zahlen markiert sein, die prim sind.
- Markiere als Erstes einmal vorbeugend alle Zahlen bis zur Grenze **n** als Primzahl.
- Gehe dann die Zahlen (Markierungen) von 2 aufwärts der Reihe nach bis **n** durch. Wenn du auf eine markierte Zahl triffst, so ist sie eine Primzahl:
 - Gib sie aus.
 - Streiche die Markierung von allen Vielfachen der Zahl weg.

Hinweise:

- Dein Programm soll mit einer Zahl auf der Befehlszeile aufgerufen werden und alle Primzahlen bis zu dieser Grenze ausgeben.
- Dein Array wird vom Typ **bool** sein:
Ist ein Element **true**, so ist es markiert und die entsprechende Zahl eine Primzahl, ist es **false**, so ist die Markierung gelöscht und die Zahl nicht prim.
Für **bool**, **true** und **false** brauchst du **stdbool.h**.
- Verwende eine fix definierte Konstante für die Größe deines Arrays und prüfe, ob die eingegebene Grenze kleinergleich dieser Arraygröße und größer als 2 ist (wenn nein: Fehlermeldung und Programmende!).

Hinweis:

Sowohl in Windows als auch in Linux ist das Speicherlimit für alle Arrays innerhalb von Funktionen ziemlich klein (bei beiden zwischen 1 MB und 8 MB).

Wenn du dein Array groß machen möchtest, musst du es daher statt innerhalb der **{ }** von **main** vor der Kopfzeile von main deklarieren!

Innerhalb von **main** kannst du Primzahlen bis rund 1 Mio. berechnen, wenn das Array außerhalb von **main** liegt, kann es rund 1 Mrd. Elemente haben.

- Zum Streichen aller Vielfachen einer Zahl wirst du eine Schleife brauchen.

Mit welchem Wert beginnt der Schleifenzähler

(was ist das kleinste "echte" Vielfache von **i**)?

Mit welcher Schrittweite muss der Schleifenzähler weitergezählt werden,

wenn er von einem Vielfachen von **i** zum nächsten Vielfachen springen soll?