

**Wichtig:** Lesen Sie auch den Teil **“Hinweise zur Aufgabe”** auf diesem Blatt; Spezifikationen in diesem Teil sind ebenfalls einzuhalten!

## Aufgabe 0: sieve (0.0 Punkte)

Implementieren Sie eine einfache Version des Siebs des Eratosthenes, einem Algorithmus zur Berechnung aller Primzahlen bis zu einem vorgegebenen Maximum. Nutzen Sie eine konstante Variable (`const int MAX = 100;`), die die größte betrachtete Zahl anzeigt. Achten Sie darauf, dass Ihre Implementierung auch mit MAX ungleich 100 zurecht kommt! Ihre Implementierung des Siebs des Eratosthenes soll dabei die folgenden Schritte umfassen:

1. **Initialisierung der benötigten Datenstrukturen:** Nutzen Sie ein Array `prime` des Datentyps `bool`, in dem vermerkt wird ob die Zahl  $i$  an Index  $i$  des Arrays `prim` oder nicht `prim` (von der Liste der Primzahlen „gestrichen“) ist. Beispielsweise gibt `prime[42]` an, ob die Zahl 42 `prim` ist. Achten Sie auf eine ausreichende Dimensionierung des Arrays und beachten Sie dabei, dass die Indizierung von Arrays in C bei 0 beginnt.
2. **Ausführung des Siebs des Eratosthenes:** Die Kernidee des Siebs des Eratosthenes ist es, aus einem Feld mit  $n$  Einträgen sukzessive nicht-prime Zahlen zu streichen, indem, beginnend bei 2, aufsteigend durch das Array gelaufen wird und alle Vielfachen von `primen`, d.h. nicht gestrichenen, Zahlen gestrichen werden.

Am Beispiel mit  $n = 26$ :

- (1) Die Zahlen 0 und 1 sind Sonderfälle und nicht `prim` - daher werden 0 und 1 gestrichen:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- (2) Der Eintrag an Index 2 ist noch nicht gestrichen, d.h. `prime[2]` ist weiter `true`. Daher wird die Zahl 2 als `prim` erkannt und alle Vielfachen von 2 gestrichen:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- (3) Der Eintrag an Index 3 ist noch nicht gestrichen, d.h. `prime[3]` ist weiter `true`. Daher wird die Zahl 3 als `prim` erkannt und alle Vielfachen von 3 gestrichen:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- (4) Der Eintrag an Index 4 ist durchgestrichen, d.h. `prime[4]` wurde in einem vorangegangenen Schritt auf `false` gesetzt. Einträge, die als nicht-`prim` erkannt wurden, werden übersprungen.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- (5) Der Eintrag an Index 5 ist noch nicht gestrichen, d.h. `prime[5]` ist weiter `true`. Daher wird die Zahl 5 als `prim` erkannt und alle Vielfachen von 5 gestrichen:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Nach diesem Schritt ist die Bearbeitung des Siebs des Eratosthenes abgeschlossen, da es keine nicht-primen Zahlen  $\leq 25$  mehr geben kann, in deren Primfaktorzerlegung nicht mindestens schon eine erkannte Primzahl vorkommt. Es ist also ausreichend die Berechnung abubrechen, sobald die Bedingung  $i * i \leq \text{MAX}$  nicht mehr erfüllt ist.

3. **Ausgabe der Zahlen**, die in `prime` nach Ausführung des Siebs des Eratosthenes weiterhin als `prim` markiert sind.

### Hinweise zur Aufgabe:

- Das C-Programm ist in der Datei `sieve.c` abzugeben. Es muss dem ANSI-C11-Standard entsprechen und mit dem GNU-C-Compiler in der vorgegebenen BS-VM kompilieren. Dazu ist der gcc mit folgenden Parametern aufzurufen:  
`gcc -std=c11 -pedantic -D_XOPEN_SOURCE=700 -Wall -Werror -o sieve sieve.c`
- Zur Nutzung des Datentyps `bool` muss der Header `stdbool.h` eingebunden werden: `#include <stdbool.h>`
- Ausgabefunktionen, wie `printf`, müssen in dieser Aufgabe nicht fehlerbehandelt werden.
- Die Ausgabe einer Ganzzahl in Variable `i` erfolgt in C mittels `printf("%d\n", i);`
- Achten Sie außerdem auf saubere Gliederung des Quellcodes und kommentieren Sie nicht-offensichtliche Codestellen!

### Hinweise zur Abgabe:

Bearbeitung: Einzelabgabe  
Abzugebende Dateien: `sieve.c`  
Abgabezeitpunkt: 17.04.2024, 17:00