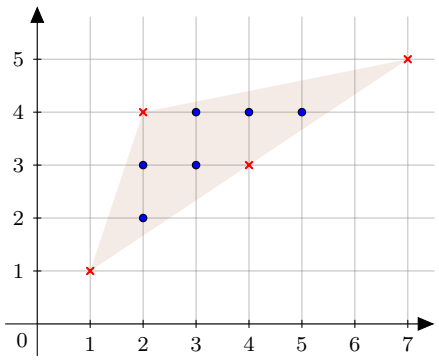
Aufgabe 5: patric (18.0 Punkte)

Schreiben Sie ein Programm `patric` (**Pa**rallel **tri**angle **c**ounter), welches eine einfache Aufgabenverteilung von Arbeitspaketen an Arbeiterthreads implementiert. Ziel der Aufgabe ist es, die Verteilung auf die Arbeiterthreads und die Synchronisierung der Ausgabe zu implementieren. Als Arbeitspakete werden dem Programm Eckpunkte auf der Standardeingabe zur Verfügung gestellt. Die Arbeiterthreads sollen die Anzahl der Punkte auf ganzzahligen Koordinaten zählen die innerhalb des Dreiecks (*interior points*, im Beispielbild ●) oder auf dessen Begrenzungslinien liegen (*boundary points*, ×).
Das Programm wird wie folgt aufgerufen: `patric <Maximale Anzahl der Arbeiterthreads>`

Einlesen der Arbeitspakete

Auf der Standardeingabe werden zeilenweise Dreiecke zur Verarbeitung angegeben. Eine Eingabezeile besteht aus drei, durch Komma getrennte Koordinatenpaaren. Ein Koordinatenpaar wird im Format *(x,y)* angegeben; bei x und y handelt es sich jeweils um Ganzzahlen. Zeilen, die nicht dem erwarteten Format entsprechen, sollen unter Ausgabe einer Warnung verworfen werden und die Ausführung des Programms fortgeführt werden.
Zur Vereinfachung dürfen Sie annehmen, dass valide Eingabezeilen keine Leerzeichen enthalten. Die Eingabezeile *(1,1),(2,4),(7,5)* symbolisiert das in nebenstehender Beispielgrafik dargestellte Dreieck und soll zu einer Erhöhung der gefundenen Punkte um 4 boundary points, bzw. 6 interior points führen.



Abarbeiten der Arbeitspakete

Für jede Eingabezeile soll ein Arbeiterthread erzeugt werden, der das entsprechende Dreieck verarbeitet. Die maximale Anzahl der gleichzeitig existierenden Arbeiterthreads soll auf die im Parameter angegebene Zahl begrenzt werden. Bei einer darüber hinausgehenden Anzahl an Arbeitspaketen soll **passiv** gewartet werden, bis diese Grenze wieder unterschritten ist, bevor neue Arbeiterthreads erzeugt werden. Die Anzahlen der gefundenen Punkte sollen über alle Dreiecke hinweg aufsummiert werden. Nutzen Sie die Funktion `countPoints` **aus dem vorgegebenen Modul** `triangle` (`triangle.c`, `triangle.h`) zum Zählen der Punkte. Die Funktionsweise des Moduls ist in der zugehörigen Headerdatei beschrieben.

Statusausgabe

Während der Ausführung soll das Programm folgende Informationen in einer Zeile ausgeben:

- Anzahl der insgesamt gefundenen Punkte, die auf Dreiecken liegen (*boundary points*)
- Anzahl der insgesamt gefundenen Punkte, die innerhalb von Dreiecken liegen (*interior points*)
- Anzahl der momentan aktiven Arbeiterthreads (d. h. Threads, die gerade `countPoints` ausführen)
- Anzahl der Threads, die die Bearbeitung ihres Arbeitspaketes abgeschlossen haben

Die Ausgabe soll in einem getrennten Kontrollfluss (*Ausgabethread*) stattfinden und dabei kontinuierlich aktualisiert werden. Zwischenzeitlich wartet der Ausgabethread **passiv**.

Makefile

Erstellen Sie ein zur Aufgabe passendes Makefile, welches die Targets `all`, `clean` und `patric` unterstützt. Greifen Sie dabei stets auf Zwischenprodukte zurück. Achten Sie darauf, dass das Makefile ohne eingebaute Regeln funktioniert (`make -Rr`). Nutzen Sie die im Moodle-Kurs genannten Compiler Flags.

Synchronisierung

Bei Zugriffen auf globale Datenstrukturen und Koordination zwischen Threads muss auf geeignete Synchronisierung geachtet werden. Nutzen Sie hierfür das vorgegebene Semaphor-Modul (`sem.o`, `sem.h`).

Hinweise zur Aufgabe:

- Hilfreiche *Manual-Pages*: **scanf(3)**, **fflush(3)**
- Der Cursor kann durch Ausgabe des Zeichens `\r` an den Anfang der Zeile zurückgesetzt werden.
- Zum Übersetzen des Programmes ist das zusätzliche Compiler Flag `-pthread` notwendig. Ihr Programm muss also mit den folgenden Flags kompilieren:
  `-std=c11 -pedantic -Wall -Werror -D_XOPEN_SOURCE=700 -pthread`
  Diese Flags werden zur Bewertung herangezogen.
- Eine Referenzimplementierung sowie einfache Testdaten finden Sie im zip-Archiv. Orientieren Sie sich am Ausgabeformat der Referenzimplementierung.
- Die Arbeiterthreads sollen im **detached**-state laufen (**pthread_detach(3p)**).
- Langsame Funktionen (z. B. **printf(3)**) dürfen nicht in kritischen Abschnitten ausgeführt werden.

Hinweise zur Abgabe:

Bearbeitung:	Zweiergruppen
Abzugebende Dateien:	`patric.c` (16 Punkte), `Makefile` (2 Punkte)
Abgabezeitpunkte nach Tafelübungsgruppe:	T01-T09: 13.07.2024, 17:30 Uhr