Zadanie 1.

1) Napisać program współbieżny symulujący działanie m producentów i n konsumentów komunikujących się przez ograniczony bufor cykliczny.
2) Działanie wątku producenta polega na wielokrotnym (zadana liczba powtórzeń) wykonywaniu po sobie ciągów instrukcji odpowiadających tzw. *produkcji danych* oraz synchronizowanego wstawienia ich do buforu. Produkcja danych polegać ma na wstrzymaniu wątku przez losowy czas w przedziale <1, 10> milisekund oraz wylosowaniu liczby całkowitej z przedziału <0, 99>. Dana jest typu string i ma następującą postać:

   Dana=[*id producenta*, *nr powt.*, *wartość*]

   `Dana=[P-1, 100, 88]`

3) Działanie wątku konsumenta polega na wielokrotnym (zadana liczba powtórzeń) wykonywaniu po sobie synchronizowanego pobrania danych z buforu oraz tzw. *konsumpcji danych*. Konsumpcja danych polegać ma na wstrzymaniu wątku przez losowy czas w przedziale <2, 12> milisekund oraz wypisania stosownego komunikatu.
4) Komunikat powinien mieć postać:

   [id konsumenta, nr powt.] >> Dana=[*id producenta*, *nr powt.*, *wartość*]

   `[K-2, 33] >> Dana=[P-1, 100, 88]`

5) W celu synchronizacji procesów wykorzystać obiekty klasy `Semaphore`.
6) Działanie programu zademonstrować dla m=4, n=5.

Zadanie 2.

1) Napisać program współbieżny symulujący działanie m czytelników i n pisarzy korzystających ze współdzielonej czytelni.
2) Działanie wątku czytelnika oraz pisarza polega na wielokrotnym (zadana liczba powtórzeń) wykonywaniu po sobie ciągów instrukcji odpowiadających tzw. *sprawom własnym* oraz synchronizowanemu korzystaniu z czytelni. Sprawy własne polegać maja na wstrzymaniu wątku przez losowy czas w przedziale <5, 15> milisekund.
3) Synchronizowane korzystanie z czytelni dla wątków czytelników i pisarzy objawia się wstrzymaniem wątku przez losowy czas w przedziale <1, 5> milisekund oraz wypisaniem stosownych komunikatów.
4) Komunikat przed wejściem i po wyjściu wypisujemy dwukrotnie na początku i na końcu sekcji krytycznej związanej z semaforem „*chroń*” i dostępem do danych współdzielonych. Komunikat powinien mieć postać (początkowe symbole oznaczają: „>>>” – przed, „<<<” – po):

   >>> [id wątku, nr powt.] :: [licz_czyt, licz_czyt_pocz, licz_pis, licz_pis_pocz]

   `>>> [C-2, 33] :: [2, 1, 0, 1]`

5) Dwa dodatkowe komunikaty wskazujące na faktyczne rozpoczęcie i zakończenie korzystania z czytelni w odpowiedni dla rodzaju procesu sposób. Komunikat powinien mieć postać (początkowe symbole oznaczają: „==>” – przed, „<==” – po):

   ==> [id wątku, nr powt.] :: [licz_czyt, licz_czyt_pocz, licz_pis, licz_pis_pocz]

   `==> [P-2, 33] :: [2, 1, 0, 1]`

6) W celu synchronizacji procesów wykorzystać obiekty klasy `Semaphore`.
7) Działanie programu zademonstrować dla m=4, n=2.

Zadanie 3.

1) Napisać program współbieżny symulujący działanie m czytelników i n pisarzy korzystających ze współdzielonej czytelni o pojemności K < m.
2) Działanie wątku czytelnika oraz pisarza polega na wielokrotnym (zadana liczba powtórzeń) wykonywaniu po sobie ciągów instrukcji odpowiadających tzw. *sprawom własnym* oraz synchronizowanemu korzystaniu z czytelni. Sprawy własne polegać maja na wstrzymaniu wątku przez losowy czas w przedziale <5, 15> milisekund.
3) Synchronizowane korzystanie z czytelni dla wątków czytelników i pisarzy objawia się wstrzymaniem wątku przez losowy czas w przedziale <1, 5> milisekund oraz wypisaniem stosownych komunikatów podczas rozpoczęcia i zakończenia korzystania z czytelni.
4) Komunikat powinien mieć postać:

   >>> [id wątku, nr powt.] :: [licz_czyt, licz_pis]

   `>>> [C-2, 33] :: [2, 0]`

5) W celu synchronizacji procesów wykorzystać obiekty klasy `Semaphore`.
6) Działanie programu zademonstrować dla m=5, n=2, K=3.

Zadanie 4.

1) Napisać program współbieżny symulujący działanie ucztujących filozofów.
2) Działanie wątku filozofa polega na wielokrotnym (zadana liczba powtórzeń) wykonywaniu po sobie ciągów instrukcji odpowiadających tzw. *sprawom własnym* oraz synchronizowanemu spożywaniu posiłków. Sprawy własne polegać maja na wstrzymaniu wątku przez losowy czas w przedziale <5, 15> milisekund.
3) Synchronizowane spożywanie posiłków z wykorzystaniem widelców objawia się wstrzymaniem wątku przez losowy czas w przedziale <1, 5> milisekund oraz wypisaniem stosownych komunikatów podczas rozpoczęcia (>>>) i zakończenia (<<<) spożywania.
4) Komunikat powinien mieć postać:

   >>> [id wątku, nr powt.] :: [licz, w0 zajęty przez, w1 zajęty przez, w2 zajęty przez, w3 zajęty przez, w4 zajęty przez]

   `>>> [F-1, 33] :: [1, 1, 1, W, W, W]`

5) W celu synchronizacji procesów wykorzystać obiekty klasy `Semaphore`.