Problemy komunikacji między wątkami

W programach współbieżnych zmienne globalne mogą być wykorzystane do komunikacji między procesami przy pomocy których procesy wymieniają między sobą dane. Powstałe w ten sposób problemy synchronizacyjne można rozwiązać przy użyciu semaforów.

Dla podanych poniżej programów współbieżnych zapisanych w pseudokodzie należy:

a) wykorzystując semafory zapisać odpowiednią synchronizację
b) implementować program w języku C

ćwiczenie 1:

Napisać synchronizację w programie współbieżnym dla dowolnych ustalonych ilości wątków p i q.

```
                    N ← ustalone
                    tablica x ← rozmiar ustalony
p0:i ← 0
p1:powtarzaj N razy          q1:powtarzaj N razy
p2:    i ← random            q1:    wypisz x
p3:    x ← i
```

Sprawdzić czy szybkość przetwarzania danych zależy od rozmiaru tablicy.

ćwiczenie 2:
Napisać synchronizację w programie współbieżnym w którym trzy wątki (p,q,r) przetwarzają dane potokowo wadług schematu p→q→r , w ten sposób że watek q wypisuje liczby parzyste, a wątek r liczby nieparzyste, które są generowane przez wątek p.

```
                    N ← ustalone
                    tablica x ← rozmiar ustalony

p0: i ← 0
p1: powtarzaj N razy    q1: powtarzaj N razy          r1:powtarzaj N razy
p2:    i ← random       q2:    wypisz_parzyste x      r2:    wypisz_nieparzyste x
p3:    x ← i
```

Sprawdzić czy szybkość przetwarzania danych zależy od rozmiaru tablicy.

Zadania dodatkowe

1) Wątki zliczają liczby pierwsze na podstawie danych zapisanych w tablicy. Pseudokod wątków obliczających wygląda następująco:

```
A[1..N] ← liczby losowe
ile ←0, ind ← 0

p1: powtarzaj dopóki ind < N
p2:     ind ← ind+1
p3:     if pierwsza( A[ind] ) = true
p4:             ile ←ile +1
p5:exit
```

Program działający dla jednego wątku jest zapisany w pliku *zadanie1.c*. Zapisać go w wersji wielowątkowej. Porównać czasy wykonania programu w zależności od ilości wątków (polecenie: *time ./a.out*)

2) Wątki zliczają liczby pierwsze na podstawie danych zapisanych w tablicy i przesyłają je do wątku głównego. Pseudokod wątków obliczających wygląda następująco:

```
A[0..N-1] ← liczby losowe
ile ←0, ind ← 0
BUFOR[SIZE]
iz ← 0
```

```
[wątek]                                  [wątek główny]
p0: j ← 0
p1: powtarzaj dopóki ind < N             m1: powtarzaj do końca danych
p2:     if pierwsza( A[ ind ] ) = true   m2:     odczyt (BUFOR,liczba)
p3:             ile ←ile +1              m3:     wypisz liczba
p4:             liczba ← A [ ind ]
p5:             zapis(BUFOR,liczba)
p6:     ind ← ind+1
p7: zapis(BUFOR,KONIEC)
p8:exit
```

Problem końca danych dla wątku głównego można rozwiązać w ten sposób, że wątki *p* zapisują wartość -1 do bufora gdy skończą przetwarzać dane. Wątek główny musi odczytać tyle liczb o wartości -1 ile jest wątków *p* po czym powinien zakończyć odczyty z bufora.
Porównać czasy wykonania programu w zależności od ilości wątków