

EP = Even Producer

EC = Even Consumer

OP = Odd Producer

OC = Odd Consumer

Testy dla rozpoczęcia z pustym buforem:

1. **Tylko proces EP:** Bufor wypełnił się do 10 elementów, a następnie doszło do blokady. Warunkiem wykonania procesu EP jest znajdowanie się w buforze mniej niż 10 liczb.

```
PS C:\Users\mikoł> & C:/Users/mikoł/AppData/Local/Programs/PowerShell/PowerShell.exe
EP: []
EP: [0]
EP: [0, 2]
EP: [0, 2, 4]
EP: [0, 2, 4, 6]
EP: [0, 2, 4, 6, 8]
EP: [0, 2, 4, 6, 8, 10]
EP: [0, 2, 4, 6, 8, 10, 12]
EP: [0, 2, 4, 6, 8, 10, 12, 14]
EP: [0, 2, 4, 6, 8, 10, 12, 14, 16]
EP: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

2. **Tylko proces OP:** Blokada następuje przed wykonaniem jakiejkolwiek operacji. Warunkiem wykonania procesu OP jest wystąpienie w buforze większej liczby wartości nieparzystych niż parzystych. W pustym buforze tych wartości mamy po 0, więc warunek nie jest spełniony
3. **Tylko proces EC i tylko proces OC:** Blokada następuje przed wykonaniem jakiejkolwiek operacji. Warunkiem wykonania każdego z tych procesów jest istnienie w buforze co najmniej 3/ co najmniej 7 wartości. W pustym buforze wartości jest 0, więc procesy nie mają jak wykonać jakiejkolwiek operacji
5. **Procesy EP i OP:** Do bufora są dodawane zarówno liczby dodatnie jak i ujemne aż do momentu, kiedy liczba wartości wyniesie 10 i następuje blokada. Wtedy liczba elementów nie jest mniejsza od 10 (warunek wykonania procesu EP) i wyrównuje się liczba parzystych i ujemnych (nie jest spełniony warunek procesu OP). Jest możliwe wystąpienie w buforze większej ilości elementów niż 10 w przypadku, gdyby w pierwszej kolejności bufor był wypełniony liczbami parzystymi. Natomiast w przypadku, gdy procesy wykonują się naprzemiennie – do bufora dodają się kolejne liczby od 0 do 9.

```

PS C:\Users\mikoł> & C:/Users/mikoł/AppData/Local/Programs/Python/Python39-64/Python.exe C:/Users/mikoł/AppData/Local/Programs/Python/Python39-64/Python.exe
EP: []
OP: [0]
EP: [0, 1]
OP: [0, 1, 2]
EP: [0, 1, 2, 3]
OP: [0, 1, 2, 3, 4]
EP: [0, 1, 2, 3, 4, 5]
OP: [0, 1, 2, 3, 4, 5, 6]
EP: [0, 1, 2, 3, 4, 5, 6, 7]
OP: [0, 1, 2, 3, 4, 5, 6, 7, 8]
EP: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
OP: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```

6. **Procesy EC i OC:** Blokada następuje przed wykonaniem jakiejkolwiek operacji. Nie da się spełnić warunku wykonania ani jednego ani drugiego procesu. Warunkiem wykonania procesu EC jest minimum 3 wartości w buforze, a wykonania procesu OC są minimum 7 wartości w buforze. 0 jest mniejsze i od 3 i od 7
7. **Procesy EP i EC:** Blokada nie występuje. Liczba elementów w buforze oscyluje pomiędzy 3 a 4. W momencie, kiedy dochodzi do 4 elementów, EC usuwa element. EP natomiast cały czas dodaje element, ponieważ liczba elementów w buforze jest stale poniżej 10.
8. **Procesy OP i OC:** Blokada występuje przed wykonaniem jakiejkolwiek operacji. Procesu OP nie da się wykonać, ponieważ nie jest spełniony warunek większej ilości liczb parzystych od nieparzystych. Jest ich równo, po 0. Natomiast warunek wykonania OC również nie jest spełniony, ponieważ liczb w buforze jest mniej niż 7.
9. **Wszystkie procesy:** Blokada nie następuje. Liczba elementów w buforze dąży do 10 i jest stale regulowana przez równowagę procesów producentów i konsumentów.

```

EC: [8, 7, 10, 9, 11, 12, 13, 14, 15]
EP: [8, 7, 10, 9, 11, 12, 13, 14, 15]
OC: [8, 7, 10, 9, 11, 12, 13, 14, 15]
EC: [10, 9, 11, 12, 13, 14, 15, 16]
OP: [10, 9, 11, 12, 13, 14, 15, 16]
OC: [9, 11, 12, 13, 14, 15, 16]
EP: [9, 11, 12, 13, 14, 15, 16]
EC: [11, 12, 13, 14, 15, 16, 18, 17]
OC: [11, 12, 13, 14, 15, 16, 18, 17]
EP: [12, 13, 14, 15, 16, 18, 17]
OP: [12, 13, 14, 15, 16, 18, 17]
EC: [13, 14, 15, 16, 18, 17, 20, 19]
OC: [14, 15, 16, 18, 17, 20, 19]
EP: [14, 15, 16, 18, 17, 20, 19]
OP: [15, 16, 18, 17, 20, 19]
EC: [15, 16, 18, 17, 20, 19, 22, 21]
OC: [15, 16, 18, 17, 20, 19, 22, 21]
OP: [16, 18, 17, 20, 19, 22, 21]
EP: [16, 18, 17, 20, 19, 22, 21]
OC: [18, 17, 20, 19, 22, 21, 23, 24]
OP: [17, 20, 19, 22, 21, 23, 24]
EC: [17, 20, 19, 22, 21, 23, 24, 25]
EP: [17, 20, 19, 22, 21, 23, 24, 25]

```

10. **Wszystkie procesy x2:** Blokada nie następuje. Rezultat i uzasadnienie są podobne jak w punkcie 9. Podstawową różnicą jest fakt, że liczby w buforze są zdublowane.

```

EC: [2, 2, 3, 4, 3, 5, 4]
EC: [4, 3, 5, 4, 6, 7, 6, 5]
EP: [4, 3, 5, 4, 6, 7, 6, 5]
OP: [3, 5, 4, 6, 7, 6, 5]
OP: [3, 5, 4, 6, 7, 6, 5]
OC: [5, 4, 6, 7, 6, 5, 8]
EP: [3, 5, 4, 6, 7, 6, 5]
EC: [5, 4, 6, 7, 6, 5, 8, 9]
EC: [6, 7, 6, 5, 8, 9, 7, 8]
EP: [7, 6, 5, 8, 9, 7, 8]
OP: [7, 6, 5, 8, 9, 7, 8]
OC: [7, 6, 5, 8, 9, 7, 8]
OP: [7, 6, 5, 8, 9, 7, 8, 10]
OC: [7, 6, 5, 8, 9, 7, 8, 10, 11]
EP: [6, 5, 8, 9, 7, 8, 10, 11]
EC: [6, 5, 8, 9, 7, 8, 10, 11]
EC: [8, 9, 7, 8, 10, 11, 9, 10]
OP: [9, 7, 8, 10, 11, 9, 10]
EP: [9, 7, 8, 10, 11, 9, 10]
OP: [9, 7, 8, 10, 11, 9, 10, 12]
OC: [9, 7, 8, 10, 11, 9, 10, 12]
EP: [7, 8, 10, 11, 9, 10, 12]
EC: [7, 8, 10, 11, 9, 10, 12, 13]
OC: [7, 8, 10, 11, 9, 10, 12, 13]

```

Testy dla rozpoczęcia z 5 elementami w buforze:

```

myMonitor = MyMonitor()

myMonitor.buffer.add(np.random.randint(0,50))
myMonitor.buffer.add(np.random.randint(0,50))
myMonitor.buffer.add(np.random.randint(0,50))
myMonitor.buffer.add(np.random.randint(0,50))
myMonitor.buffer.add(np.random.randint(0,50))

```

1. **Tylko proces EP:** Blokada nastąpiła do uzupełnienia bufora do 10. Warunkiem wykonania procesu EP jest znajdowanie się w buforze mniej niż 10 liczb.

```

PS C:\Users\mikal> & C:/Users/mikal/AppData
EP: [0, 49, 15, 28, 28]
EP: [0, 49, 15, 28, 28, 0]
EP: [0, 49, 15, 28, 28, 0, 2]
EP: [0, 49, 15, 28, 28, 0, 2, 4]
EP: [0, 49, 15, 28, 28, 0, 2, 4, 6]
EP: [0, 49, 15, 28, 28, 0, 2, 4, 6, 8]

```

2. **Tylko proces OP:** W tym przypadku wynik w dużej mierze zależy od wylosowanych wartości inicjalizujących bufor. Jeśli bufor zostanie zainicjalizowany większą liczbą wartości parzystych od nieparzystych – proces uruchomi się i blokada nastąpi dopiero w momencie wyrównania ilości liczb parzystych i nieparzystych. W innym przypadku blokada wystąpi już na samym początku.
3. **Tylko proces EC:** W tym przypadku również wynik zależy od losowych wartości, którymi inicjalizujemy bufor. W przypadku jeśli liczba na wyjściu bufora będzie parzysta – proces uruchomi się i blokada wystąpi albo w momencie kiedy na wyjściu będzie liczba nieparzysta albo kiedy liczba wartości spadnie do 3. W innym przypadku blokada nastąpi przed uruchomieniem procesu.

4. **Tylko proces OC:** Blokada następuje przed uruchomieniem procesu. Liczba wartości w buforze jest równa 5, a do uruchomienia procesu potrzebujemy minimum 7.
5. **Procesy EP i OP:** Blokada następuje w momencie, kiedy liczba wartości przekroczy 10 (warunek procesu EP) i liczba parzystych nie będzie większa od nieparzystych (warunek procesu OP).
6. **Procesy EC i OC:** Proces OC nie wykonuje się ani razu, ponieważ nie jest spełniony warunek liczby wartości w buforze większej niż 7. Proces EC wykonuje się dopóki liczba nie spadnie do 3 lub na wyjściu nie otrzymamy liczby nieparzystej. Blokada następuje w każdym wypadku, od wartości losowych zależy jednak czy na samym początku czy po wykonaniu 1-2 iteracji.

```
PS C:\Users\mikor> & C:/Use
EC: [36, 10, 17, 35, 35]
OC: [10, 17, 35, 35]
EC: [10, 17, 35, 35]
EC: [17, 35, 35]
```

Na powyższym zrzucie ekranu występuje OC ze względu na niezbyt odpowiedni wybór miejsca printowania wartości bufora. Znajduje się ono przed sprawdzeniem warunków wykonania procesu.

7. **Procesy EP i EC:** Jeśli wylosowana liczba na wyjściu bufora będzie nieparzysta to blokada nastąpi po wypełnieniu bufora 10 liczbami. Proces EC będzie zablokowany od początku, a po umieszczeniu liczb do 10 zablokuje się proces EP. Jeśli natomiast wylosowane liczby będą parzyste, blokada nie nastąpi.
8. **Procesy OP i OC:** Blokada nastąpi na pewno. W przypadku jeśli wartości wylosowanych w buforze nieparzystych będzie więcej niż parzystych – nie uruchomi się proces OP i proces OC również się nie uruchomi, bo liczba wartości nie przekroczy 7. Jeśli natomiast liczba wartości parzystych będzie większa – proces OP będzie dodawał liczby aż do momentu wyrównania. Jeśli moment wyrównania nastąpi po przekroczeniu 7 wartości w buforze, a na wyjściu będzie liczba nieparzysta – uruchomi się proces OC. Blokada nastąpi dopiero, gdy na wyjściu bufora znajdzie się liczba parzysta. Zablokuje to działanie procesu OC, a w konsekwencji po wyrównaniu wartości parzystych i nieparzystych – także proces OP.
9. **Wszystkie procesy:** Blokada nie następuje. Liczba elementów w buforze dąży do 10 i jest stale regulowana przez równowagę producentów i konsumentów

```
EC: [3, 4, 6, 5, 7, 8, 9]
EP: [3, 4, 6, 5, 7, 8, 9]
OC: [3, 4, 6, 5, 7, 8, 9]
OP: [4, 6, 5, 7, 8, 9, 10]
EP: [4, 6, 5, 7, 8, 9, 10]
OC: [4, 6, 5, 7, 8, 9, 10, 11]
OP: [5, 7, 8, 9, 10, 11, 12]
EP: [5, 7, 8, 9, 10, 11, 12, 13]
EC: [5, 7, 8, 9, 10, 11, 12, 13]
OC: [5, 7, 8, 9, 10, 11, 12, 13, 14]
OP: [8, 9, 10, 11, 12, 13, 14]
EP: [8, 9, 10, 11, 12, 13, 14]
OC: [8, 9, 10, 11, 12, 13, 14, 15, 16]
EC: [8, 9, 10, 11, 12, 13, 14, 15, 16]
OP: [10, 11, 12, 13, 14, 15, 16]
EP: [10, 11, 12, 13, 14, 15, 16, 17]
OC: [10, 11, 12, 13, 14, 15, 16, 17]
EC: [11, 12, 13, 14, 15, 16, 17, 18]
```

10. **Wszystkie procesy x2:** Blokada nie następuje. Wynik i uzasadnienie są podobne jak w punkcie 9. Podstawową różnicą jest fakt, że liczby w buforze są zdublowane.

```
OC: [29, 45, 34, 1, 11, 0]
EP: [29, 45, 34, 1, 11, 0]
OP: [29, 45, 34, 1, 11, 0, 0]
EC: [29, 45, 34, 1, 11, 0, 0]
OC: [29, 45, 34, 1, 11, 0, 0]
EP: [29, 45, 34, 1, 11, 0, 0]
EP: [45, 34, 1, 11, 0, 0, 2, 1]
EP: [11, 0, 0, 2, 1, 1, 2]
OC: [11, 0, 0, 2, 1, 1, 2]
EP: [11, 0, 0, 2, 1, 1, 2, 4]
OP: [0, 0, 2, 1, 1, 2, 4]
OC: [0, 0, 2, 1, 1, 2, 4]
OP: [0, 0, 2, 1, 1, 2, 4, 4, 3]
EC: [0, 2, 1, 1, 2, 4, 4, 3]
EC: [0, 2, 1, 1, 2, 4, 4, 3, 3]
EP: [1, 1, 2, 4, 4, 3, 3]
OC: [1, 1, 2, 4, 4, 3, 3, 6]
OP: [1, 2, 4, 4, 3, 3, 6]
OP: [1, 2, 4, 4, 3, 3, 6]
EP: [1, 2, 4, 4, 3, 3, 6]
OC: [1, 2, 4, 4, 3, 3, 6, 5, 5]
EC: [1, 2, 4, 4, 3, 3, 6, 5, 5, 6]
EC: [1, 2, 4, 4, 3, 3, 6, 5, 5, 6]
```