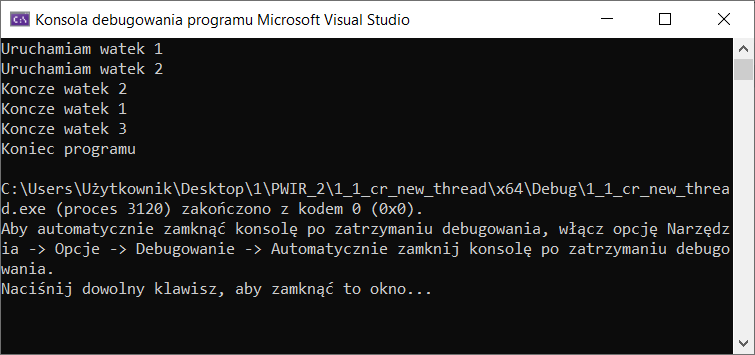
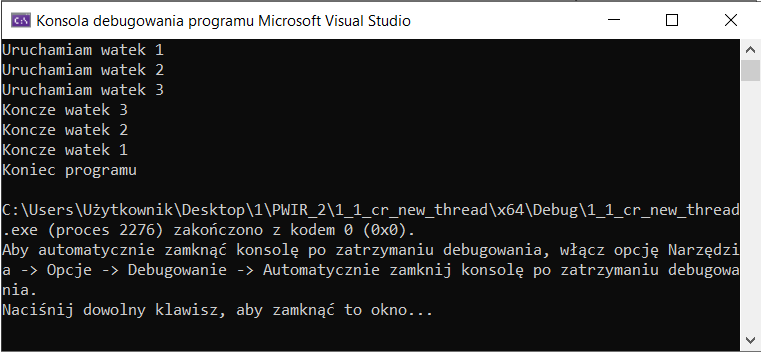
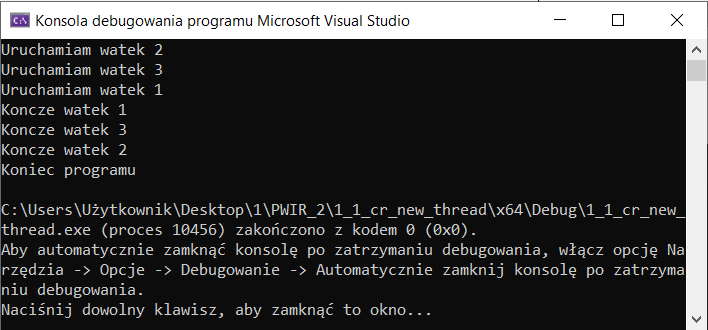
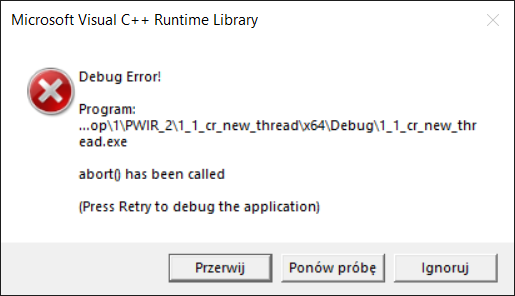
|  |  |
| --- | --- |
| Akademia Nauk Stosowanych w Nowym Sączu  Wydział Nauk Inżynieryjnych | |
| Imię i nazwisko: | Michał Bernardy |
| Grupa: | P1 |
| Ocena: |  |

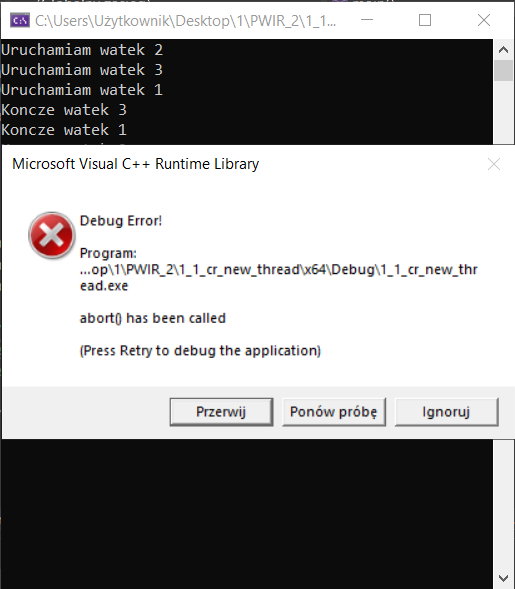
KOD1

1.Kompilując kod z pliku 1\_1\_cr\_new\_thread kilka razy możemy zauważyć że nie otrzymujemy tych samych wyników co widać na poniższych screenach z kompilacji. Kolejność zakończenia wątków jest losowa, ponieważ wątki są wykonywane równolegle, a system operacyjny decyduje o przydzieleniu czasu procesora na podstawie wielu czynników, takich jak obciążenie CPU.

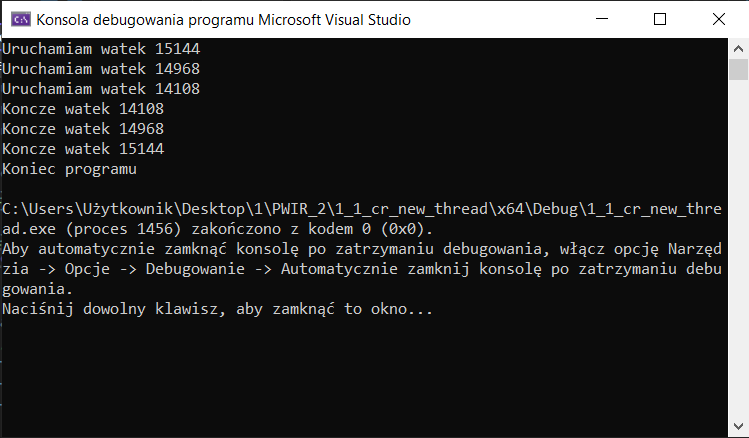


2. Usunięcie joinów powoduje, że wątek główny kończy działanie, zanim wątki poboczne się zakończą. Gdy kompilujemy kod bez joinów otrzymujemy takie wyjście: 

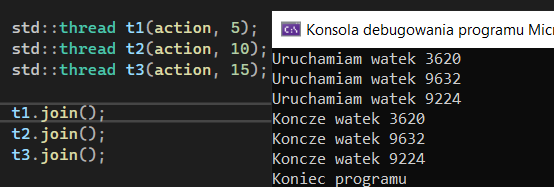
3. W tym przypadku dostajemy różne wyniki, możliwy ten ze screena z punktu 2 lub wykonanie jednego lub dwóch wątków zanim otrzymamy ten błąd.



4. Modyfikujemy kod tak, aby zamiast przekazywanego id funkcja action() (i usuwamy z niej pobieranie zmiennej int id ) używała std::this\_thread::get\_id() do wyświetlania identyfikatora wątku.

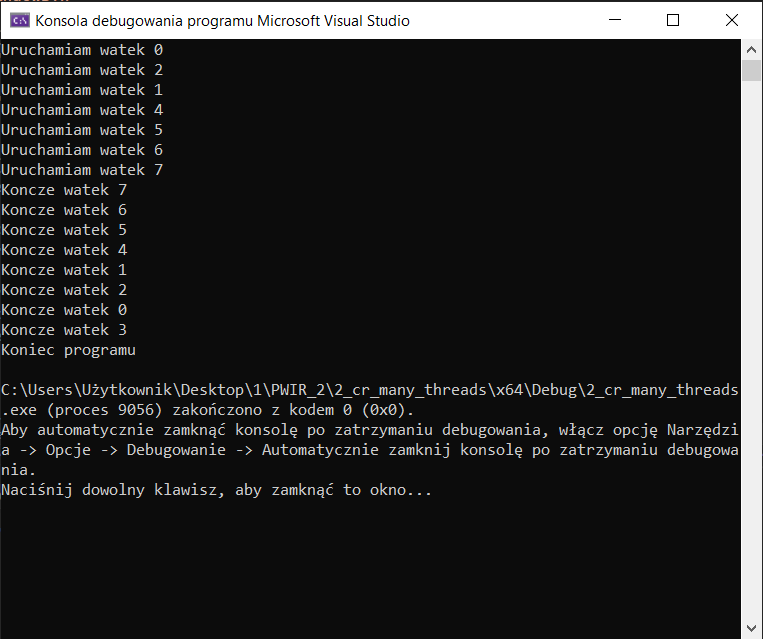
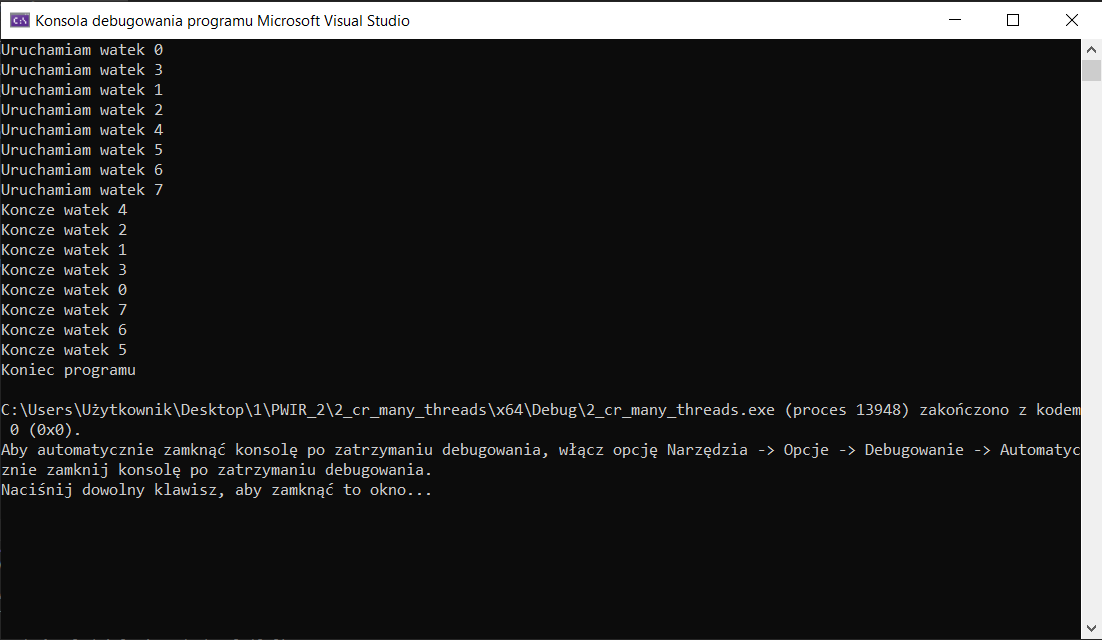
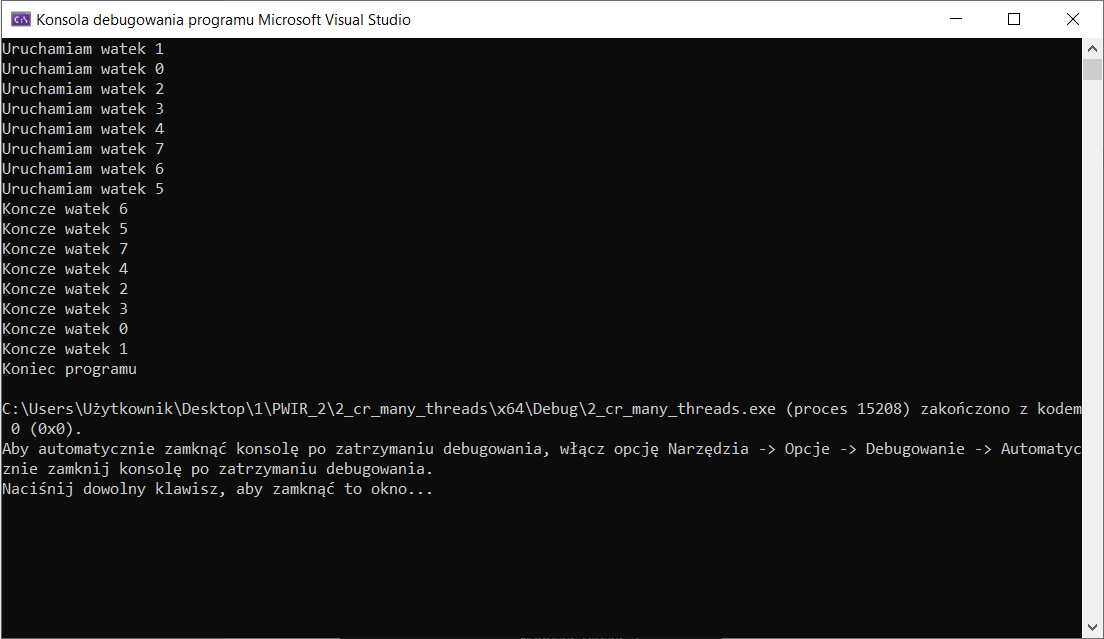
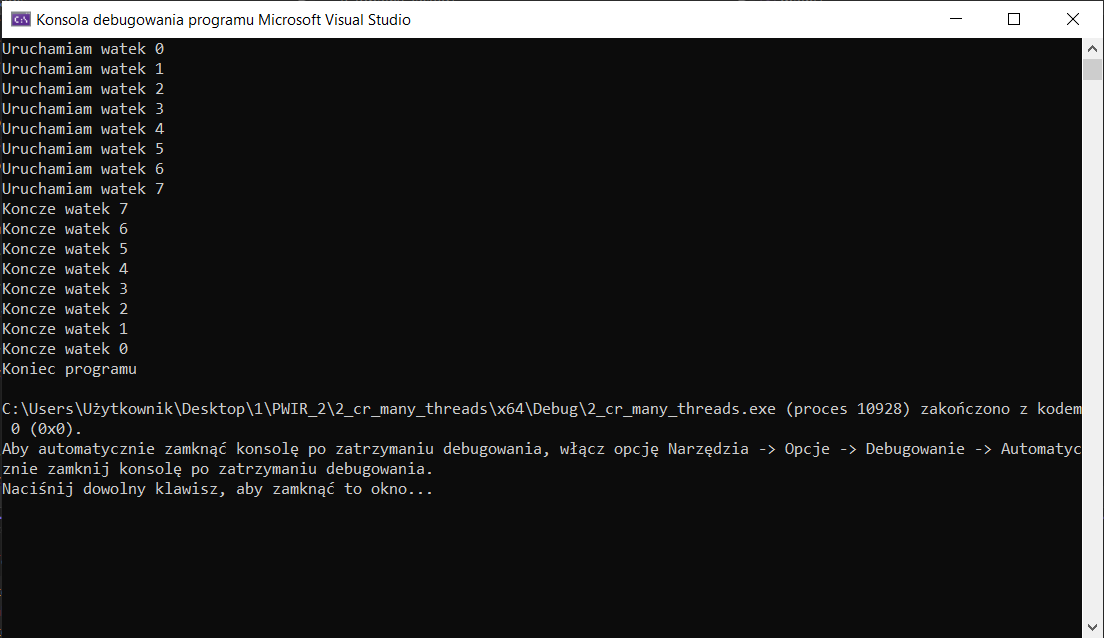


5. Modyfikujemy kod tak, aby funkcja action() przyjmowała parametr sleep\_time, określający czas uśpienia wątku, zamiast używać stałej wartości. Usuwamy stałe Sleep(10 \* 1000) i zastępujemy je Sleep(sleep\_time \* 1000), co pozwala na dynamiczne określanie czasu uśpienia dla każdego wątku.

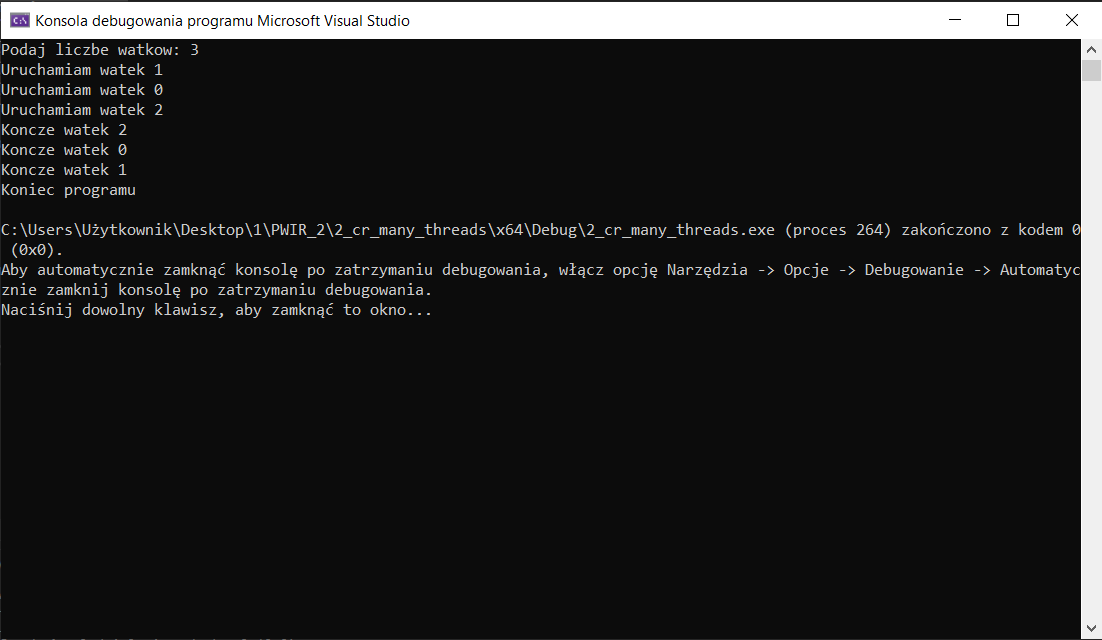
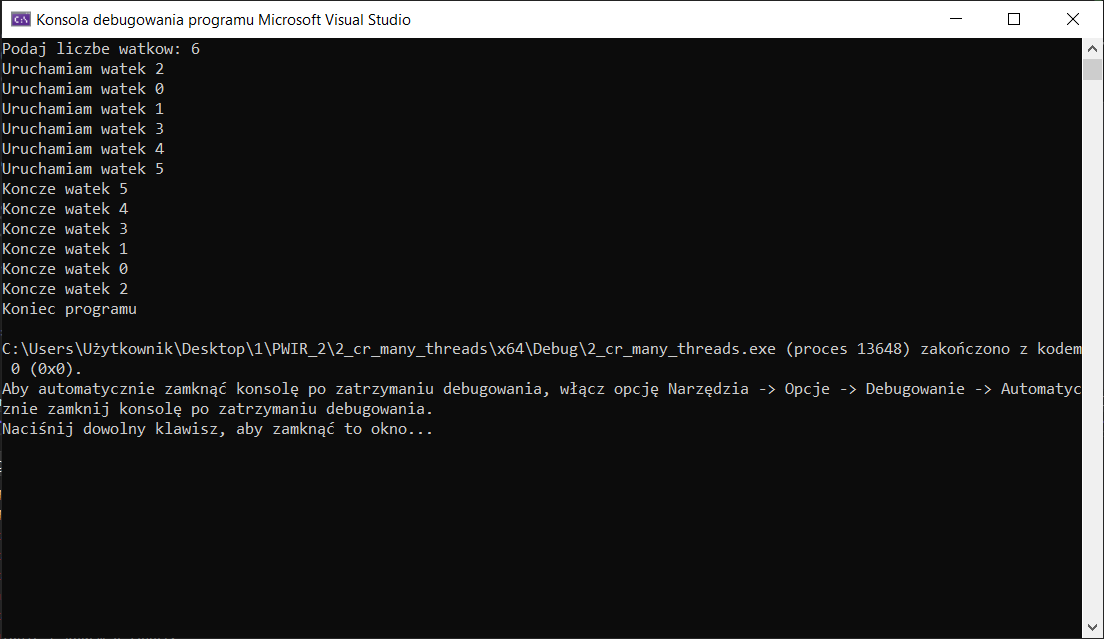
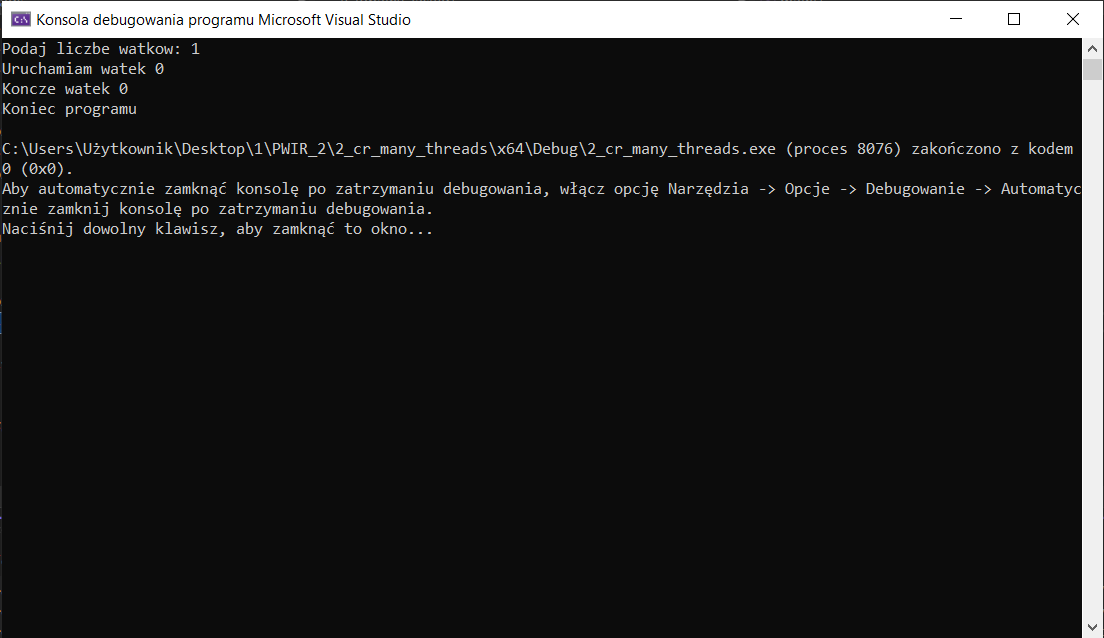


KOD2

1.W tym przypadku również kompilując kod z 2\_cr\_many\_threads kilka razy możemy zauważyć że nie otrzymujemy tych samych wyników co widać na poniższych screenach z kompilacji. Kolejność wykonywania działa również jak w punkcie pierwszym z kodu pierwszego.

2. Wprowadzono zmianę, aby liczba wątków była pobierana od użytkownika za pomocą funkcji scanf(). Na podstawie tej wartości program dynamicznie alokuje tablicę wskaźników na wątki i tworzy odpowiednią liczbę wątków do wykonania.

3. Zamiast używać statycznej tablicy wskaźników na wątki, program teraz korzysta z std::vector, który automatycznie zarządza rozmiarem i pamięcią. Wątki są dodawane do wektora za pomocą funkcji push\_back(), a pamięć jest zwalniana ręcznie za pomocą delete , podobnie jak w poprzedniej wersji programu.

