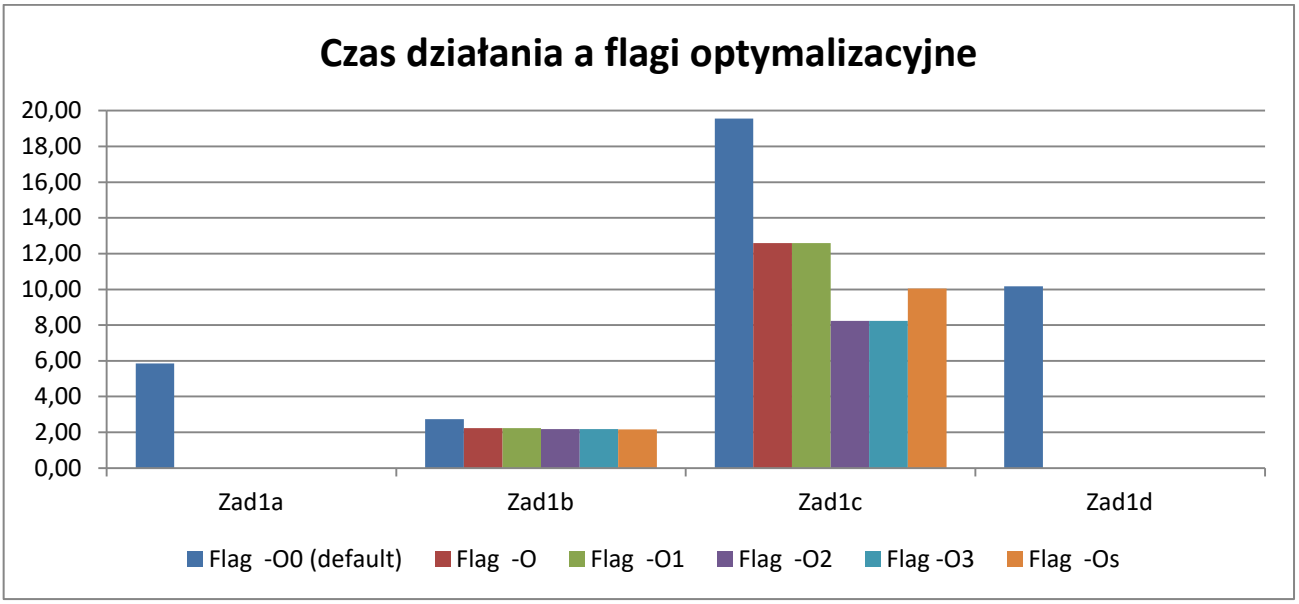
Raport

Zadanie 1. Kompilator, optymalizacja, pomiar czasu (50%)



Adnotacja do zad1a:

Program tworzy zmienna sum, a następnie wykonuje operacje na niej dodając lub odejmując kolejne liczby. Podczas kompilowania kompilator optymalizuje kod poprzez wyliczenie wartości stałych oraz redukcję zbędnych wyrażeń, co powoduje drastyczne zmniejszenie się czasu wykonania programu.

Adnotacja do zad1b:

Program tworzy 2 tablice char o rozmiarze SIZE i wpisuje do każdej komórki losowy znak od a-z. Następnie tworzy 3 tablice i scala dwie tablice w jeden napis. Czasy wykonania są niemal identyczne. Zastosowanie funkcji malloc oraz rand() nie pozwala wyraźną optymalizację kodu.

Adnotacja do zad1c:

Program tworzy tablice o rozmiarze SIZE i wpisuje do niej kolejne liczby ciagu Fibonacciego, uzywajac funkcji rekurencyjnej (zabieg celowy). Przy flagach O2 i O3 czas wykonania zmniejszył się prawie o połowę w porównaniu do domyślnej flagi. Efekt ten osiągnięto przez zastąpienie rekursji ogonowej przez iterację.

Adnotacja do zad1d:

Program wypisuje liczby doskonale w zakresie NUMBER lub w podanym przez użytkownika. Liczba doskonała to taka liczba, której suma dzielników, oprócz niej samej, jest równa rozpatrywanej liczbie. Podobnie jak w zad1a optymalizacja przebiega poprzez wyliczenie wartości stałych oraz redukcję zbędnych wyrażeń.

Wnioski:

Analizując powyższe wyniki, najlepszy czas wykonania się programu można osiągnąć poprzez zastosowanie flagi –O2 lub flagi –O3. Jednakże, z dwóch powyższych flag zalecaną jest flaga –O2. Jest to podyktowane tym, że flaga O3, która ma najwyższy poziom optymalizacji może spowodować zmniejszenie precyzji obliczeń oraz znaczny wzrost objętości kodu maszynowego.