



Ad a) Należy uważać na flagi począwszy od O2, ponieważ zawierają one domyślnie optymalizowanie nieużywanych zmiennych. Bez wypisywania wyniku obliczeń (używając zmiennej "sum") czas dla O2/O3 był bliski 0.00 sekundy, ponieważ kompilator pomijał pętlę w swoich obliczeniach.

Ad b) Czas liczony jest również podczas przygotowywania danych do pliku. Mogło to być dobre miejsce do optymalizacji dla kompilatora.

Ad c) Wygenerowany kod asemlera jest praktycznie taki sam, ponieważ używamy bardzo prostej funkcji "malloc" oraz pętli for, która jest napisany w sposób uniemożliwiający wektoryzację kodu. Z tego powodu, czasy wykonania dla różnych flag optymalizacji są bardzo zbliżone.

Ad d) Wyraźny skok wydajności widać od O1. Wynosi ok. x4 i nie zwiększa się znacznie dla kolejnych flag optymalizacji. Może być to związane z wektoryzacją pętli for.

Dla przedstawionych rezultatów widać, że optymalizacja z użyciem -O3 zapewnia najlepszy czas wykonywania lub porównywalny do -O2. Może się to jednak wiązać z dłuższym czasem generowania kodu lub plikami wykonywalnymi zajmującymi większą ilość pamięci.