# Zadania

### Zadanie 0: Porównanie różnych poziomów optymalizacji

Zadanie pierwsze polega na wykorzystaniu linuksowej komendy time, w celu znalezienia najlepszej opcji optymalizacji dla przykładowego programu. Używając opcji optymaliacji, napisz skrypt bashowy do sprawdzania czasu wykonania programu z każdą opcją optymalizacji i znajdź dla której opcji:

1. czas wykonania jest najkrótszy
2. Plik wynikowy jest najmniejszy (najbardziej miarodajne będzie sprawdzenie kodu w assemblerze)

### Zadanie 1: Zmienne globalne oraz lokalne

1) -

2) zmienne globalne są zawsze trzymane osobno w sekcji .data, lokalne wrzucane są na stos

3) w przypadku optymalizacji lokalne zmienne mogą być ładowane od razu do rejestrów, jako stałe, brak użycia stosu do ich przechowywania

4) O3: wszystkie poza printf, kompilator może korzystać z takich Copy propagation CSE i constant folding

### Zadanie 2: Pętle

1) -

2) program zamienił sumowanie w pętli na mniej skomplikowany wzór, zmniejszajac ilość operacji oraz porównań (do 1).

gcc aktualnie jest w stanie doprowadzić tylko do sytuacji w której próbuje przewidzieć wynik i obliczyć jego wstępną wartość,

mimo wszystko potrzebuje dużo więcej kodu oraz występuje możliwość wystąpienia wielu porównań. Clang w przeciwnienstwie do gcc wykorzystuje

rozwijanie petli

3) -

4) -

### Zadanie 3: Instrukcje warunkowe

1) -

2) int conditiontest2(int test1, int test2)

{

int result = test1;

int p = test1 - test2;

if (p<=0)

{

result = 0;

if (p < 0)

result = test2;

}

return result;

}

3) z kodu nie zostala usunięta operacja else if

### Zadanie 4?: Eliminacja wspólnych podwyrażeń

1) -

### Zadanie 5: Optymalizacje O0

1) volatile const int x = 5;

\*const\_cast<int\*>(&x) = 2\*x;

f((const int)x);

### Zadanie 6: Ofast

1) Zmiana kolejnosci wykonywania dzialan

2) double x = 1.;

double y = -1.;

double z = 1e-20;

double w = 1.;

### Zadanie 7: Proste optymalizacje a optymalizacja kompilatora

1) -

2) -

3) -

### Zadanie 8: Link Time Optimization

Skompiluj podane pliki (przy użyciu O2 lub O3), dokonaj linkowania i sprawdź czas wykonania przy pomocy polecenia time. Następnie użyj opcji flto. Jak zmienił się czas wykonania i jak rozmiar pliku?

### 