

Ćwiczenie 2

Organizacja instrukcji warunkowych i pętli z wykorzystaniem skoków warunkowych

Napisz program znajdowania wszystkich liczb pierwszych pośród N początkowych liczb naturalnych $\{1, 2, 3, \dots, N-2, N-1, N\}$. Użyj w tym celu algorytmu znanego jako sito Eratostenesa (szczegóły - można znaleźć w wielu miejscach w Internecie). Zarys tego algorytmu jest następujący:

a) Wypełnij tablicę liczbami od 1 do N

b) Powtarzaj dla $n=2, \dots, N/2$:

Wykreśl z tablicy wszystkie wielokrotności liczby n

c) Liczby pierwsze to te które nie zostały wykreślone.

Załącz, że zakres N to stała ustalona podczas pisania programu. Zarezerwuj tablicę liczb, z których liczby będą wykreślane w obszarze `.data`. Nadaj jej nazwę `numall`. "Wykreślanie" liczb z tablicy zaimplementuj jako wpisywanie wartości 0. Po wykreśleniu wszystkich liczb nie będących liczbami pierwszymi przepisz pozostałe liczby pierwsze do obszaru(tablicy) o nazwie `primes` w sekcji `.data`. Liczbę znalezionych liczb pierwszych zapisz w polu `nprimes` (o typie `.word`). Po wypełnieniu tablicy `primes`, wypisz jej zawartość (znalezione liczby pierwsze) na konsoli.

UWAGA: zadbaj o efektywność Twojej implementacji. Zastanów się nad prostymi i nie zmieniającymi idei algorytmu ulepszeniami algorytmu pozwalającymi uzyskać zwiększenie efektywności czasowej programu.

Przygotuj schemat blokowy zaimplementowanego algorytmu lub zapisz go w postaci pseudokodu w języku wysokiego poziomu (np. Java). Upewnij się, że Twój kod w assemblerze odpowiada schematowi/pseudokodowi.

Poeksperymentuj z dużymi wartościami N . Oszacuj szybkość działania Twojego kodu w symulatorze (liczba wykonanych instrukcji na sekundę). Zaimplementuj analogiczny algorytm w Javie i porównaj czasy wykonania (zbadaj tylko czas wykonania algorytmu do chwili wypełnienia tablicy `nprimes` - nie uwzględniaj czasu wypisywania liczba na konsolę).