

# Zadanie 1

## Minimalny stopień wielomianu

Zaimplementuj w asemblerze funkcję `polynomial_degree` wołaną z języka C o sygnaturze:

```
int polynomial_degree(int const *y, size_t n);
```

Argumentami funkcji są wskaźnik `y` na tablicę liczb całkowitych  $y_0, y_1, y_2, \dots, y_{n-1}$  i `n` zawierający długość tej tablicy. Wynikiem funkcji jest najmniejszy stopień wielomianu  $w(x)$  jednej zmiennej o współczynnikach rzeczywistych, takiego że  $w(x + kr) = y_k$  dla pewnej liczby rzeczywistej  $x$ , pewnej niezerowej liczby rzeczywistej  $r$  oraz  $k = 0, 1, 2, \dots, n-1$ .

Przyjmujemy, że wielomian tożsamościowo równy zeru ma stopień  $-1$ . Wolno założyć, że wskaźnik `y` jest poprawny i wskazuje na tablicę zawierającą `n` elementów, a `n` ma dodatnią wartość.

Zauważmy, że jeżeli wielomian  $w(x)$  ma stopień  $d$  i  $d \geq 0$ , to dla  $r \neq 0$  wielomian  $w(x + r) - w(x)$  ma stopień  $d-1$ .

## Kompilowanie rozwiązania

Rozwiązanie będzie kompilowane poleceniem:

```
nasm -f elf64 -w+all -w+error -o polynomial_degree.o polynomial_degree.asm
```

## Przykład użycia

Przykład użycia znajduje się w pliku `polynomial_degree_example.c`. Można go skompilować i skonsolidować z rozwiązaniem poleceniami:

```
gcc -c -Wall -Wextra -std=c17 -O2 -o polynomial_degree_example.o polynomial_degree_example.c
gcc -o polynomial_degree_example polynomial_degree_example.o polynomial_degree.o
```

## Oddawanie rozwiązania

Jako rozwiązanie należy wstawić w Moodle plik o nazwie `polynomial_degree.asm`.

## Ocenianie

Oceniane będą poprawność i szybkość działania programu, zajętość pamięci (rozmiary poszczególnych sekcji), styl kodowania, komentarze. Wystawienie oceny może też być uzależnione od osobistego wyjaśnienia szczegółów działania programu prowadzącemu zajęcia.

Tradycyjny styl programowania w asemblerze polega na rozpoczynaniu etykiet od pierwszej kolumny, mnemoników od dziewiątej kolumny, a listy argumentów od siedemnastej kolumny. Inny akceptowalny styl prezentowany jest w przykładach pokazywanych na zajęciach. Kod powinien być dobrze skomentowany, co oznacza między innymi, że każda procedura powinna być opatrzona informacją, co robi, jak przekazywane są do niej parametry, jak przekazywany jest jej wynik, jakie rejestry modyfikuje. To samo dotyczy makr. Komentarza wymagają także wszystkie kluczowe lub nietrywialne linie wewnątrz procedur lub makr. W przypadku

asemblera nie jest przesadą komentowanie prawie każdej linii kodu, ale należy jak ognia unikać komentarzy typu „zwiększenie wartości rejestru rax o 1”.