

## Zadanie 1

### Diakrytyzator

Zaimplementuj w asemblerze x86\_64 program, który czyta ze standardowego wejścia tekst, modyfikuje go w niżej opisany sposób, a wynik wypisuje na standardowe wyjście. Do kodowania tekstu używamy UTF-8, patrz <https://pl.wikipedia.org/wiki/UTF-8>. Program nie zmienia znaków o wartościach unicode z przedziału od 0x00 do 0x7F. Natomiast każdy znak o wartości unicode większej od 0x7F przekształca na znak, którego wartość unicode wyznacza się za pomocą niżej opisanego wielomianu.

### Wielomian diakrytyzujący

Wielomian diakrytyzujący definiuje się przez parametry wywołania diakrytyzatora:

```
./diakrytyzator a0 a1 a2 ... an
```

jako:

$$w(x) = a_n * x^n + \dots + a_2 * x^2 + a_1 * x + a_0$$

Współczynniki wielomianu są nieujemnymi liczbami całkowitymi podawanymi przy podstawie dziesięć. Musi wystąpić przynajmniej parametr `a0`.

Obliczanie wartości wielomianu wykonuje się modulo 0x10FF80. W tekście znak o wartości unicode `x` zastępuje się znakiem o wartości unicode  $w(x - 0x80) + 0x80$ .

### Zakończenie programu i obsługa błędów

Program kwituje poprawne zakończenia działania, zwracając kod 0. Po wykryciu błędu program kończy się, zwracając kod 1.

Program powinien sprawdzać poprawność parametrów wywołania i danych wejściowych. Przyjmujemy, że poprawne są znaki UTF-8 o wartościach unicode od 0 do 0x10FFFF, kodowane na co najwyżej 4 bajtach i poprawny jest wyłącznie najkrótszy możliwy sposób zapisu.

### Przykłady użycia

Polecenie

```
echo "Zażółć gęślą jaźń..." | ./diakrytyzator 0 1; echo $?
```

wypisuje

```
Zażółć gęślą jaźń...
0
```

Polecenie

```
echo "Zażółć gęślą jaźń..." | ./diakrytyzator 133; echo $?
```

wypisuje

```
Zaaąąą gąąłą jaąąą
0
```

Polecenie

```
echo "ŁÓŚ" | ./diakrytyzator 1075041 623420 1; echo $?
```

wypisuje

```
„0”  
0
```

## Polecenie

```
echo -e "abc\n\x80" | ./diakrytnizator 7; echo $?
```

## wypisuje

```
abc  
1
```

## Oddawanie rozwiązania

Jako rozwiązanie należy wstawić w Moodle plik o nazwie **diakrytnizator.asm**. Rozwiązanie będzie kompilowane poleceniami:

```
nasm -f elf64 -w+all -w+error -o diakrytnizator.o diakrytnizator.asm  
ld --fatal-warnings -o diakrytnizator diakrytnizator.o
```

## 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”.

## Pytania

Pytania dotyczące zadania można zadawać w przeznaczonym do tego wątku na Moodle.