

## Ćwiczenie nr 5. Zliczanie słów

Twoim zadaniem jest zaimplementowanie programu zliczającego wystąpienia wszystkich słów w podanym ciągu znaków (wprowadzonym przez konsolę lub wczytanym z pliku). Np. jeśli wejściem jest:

```
lubie placki pyszne placki
```

program wypisać ma (w dowolnej kolejności, sortowanie nie jest konieczne):

```
placki 2
lubie 1
pyszne 1
```

Program **musi** być podzielony na funkcje oparte o konwencję ramek stosu. To znaczy:

- informacje przekazywane są jedynie poprzez rejestry `a0-a3` (argumenty) i `v0-v1` (wartości zwracane) – nie przez zmienne globalne (czy to rejestry, czy etykiety w `.data`),
- przekazywanie kontroli odbywa się przez instrukcje `jal` oraz `jr` (oczywiście z wyłączeniem pętli),
- wykorzystywany jest stos do zapamiętywania rejestrów funkcji wywołującej/wywoływanej.

Potrzebna Ci będzie możliwość przechowywania listy słów wraz z liczbą ich wystąpień. Można do tego podejść na co najmniej dwa sposoby, np.:

- zarezerwować jeden obszar pamięci (w sposób statyczny, w `.data`, lub dynamiczny, przez syscall `sbrk`) na przechowywanie łańcuchów (np. rozdzielonych znakami `'\0'`), drugi obszar na tablicę wskaźników do początków poszczególnych łańcuchów, oraz kolejny na tablicę ich licznosci,
- zarezerwować pewien obszar pamięci i posługiwać się nim jak tablicą lub listą jednokierunkową struktur zawierających ciąg znaków (czy to przez wskaźnik czy bezpośrednio w strukturze), liczbę wystąpień tego ciągu, adres następnej takiej struktury.

Nie wolno jednak stosować wielu osobnych etykiet na ściśle ograniczoną liczbę słów (np. `string1`, `string2`, `string3`, `string4`, ... - jako `.asciiz`).

Jeśli zdecydujesz się na implementację struktur, pamiętaj o zaplanowaniu ich układu: np. jeśli początek elementu jest pod adresem `ptr`, to np. `ptr+0` zawiera ilość wystąpień, `ptr+4` początek ciągu, itd. Pamiętaj o wyrównywaniu struktur do rozmiaru będącego wielokrotnością 4 bajtów!

Rozwiązanie powinno składać się z wyraźnie wydzielonych funkcji, szczególnie do obsługi struktur danych (dodaj element, znajdź element, wyświetl elementy) oraz operacji na łańcuchach ([strcmp](#), [strcpy](#) – jeśli potrzebne).

Przyjmij następujące założenia:

- wczytywany łańcuch jest długości ograniczonej do 1024 bajtów
- wczytywany łańcuch zawiera tylko litery (małe i wielkie – traktujemy tak samo!), spacje, i `'\0'`
- wczytywany łańcuch nie zawiera słów dłuższych niż 20 znaków

Pro-tip: przełącz widok segmentu danych MARSa w tryb ASCII, by lepiej widzieć treść w pamięci.

Podział na etapy (punktacja, jak zwykle, kumulacyjna):

- wczytanie łańcucha i pocięcie go na pojedyncze słowa (0.5 pkt)
- zapisywanie słów w wybranej strukturze danych (2.5 pkt za osobne tablice, 3.5 pkt za implementację struktur)
- przeszukiwanie struktury danych i wykrywanie powtórnych wystąpień (3.5/4.5 pkt, patrz punkt poprzedni)
- zabezpieczenia, np. małe/wielkie litery, wyczerpanie pamięci wewnętrznej, ... (4.0/5.0 pkt, j/w)